2011 First International Conference on Data Compression, Communications and Processing

A review of DNA microarray image compression

Miguel Hernández-Cabronero*, Ian Blanes*†, Joan Serra-Sagristà*, Michael W. Marcellin*‡
Email: mhernandez@deic.uab.cat, {Ian.Blanes,Joan.Serra}@uab.cat, mwm@email.arizona.edu
*Department of Information and Communications Engineering
Universitat Autònoma de Barcelona. Barcelona, Spain
†Centre National d'Études Spatiales (CNES)
Toulouse, France
‡Department of Electrical and Computer Engineering
University of Arizona. Tucson, AZ, USA

***Abstract*—We review the state of the art in DNA microarray image compression. First, we describe the most relevant approaches published in the literature and classify them according to the stage of the typical image compression process where each approach makes its contribution. We then summarize the compression results reported for these microarray-specific image compression schemes. In a set of experiments conducted for this paper, we obtain results for several popular image coding techniques, including the most recent coding standards. Prediction-based schemes CALIC and JPEG-LS, and JPEG2000 using zero wavelet decomposition levels are the best performing standard compressors, but are all outperformed by the best microarray-specific technique, Battiato's CNN-based scheme.**

***Keywords*-microarray images, microarray image compression, image coding standards, JPEG2000.**

I. Introduction

DNA microarray technology allows the analysis of the expression of thousands of genes in a single experiment, and has become a very important tool in medicine and biology for the study of genetic function, regulation and interaction [1]. Genome-wide monitoring is possible with existing DNA microarrays, which are used in research against cancer [2] and HIV [3], among many other applications.

DNA microarrays consist in a solid surface on which thousands of known genetic sequences are bound. Each sequence is placed in one microscopic hole or spot and all spots are arranged conforming to a regular grid pattern. Two samples, for example from healthy and tumoral tissue, are labeled, respectively, with green and red fluorescent markers called Cy3 and Cy5. Then, equal amounts of the labeled samples are made to react on the microarray. If one sample had expressed a given sequence placed in the microarray, part of it is hybridized and fixed in the correspondent spot; else, it gets washed away and will not be present in the spot. Once the hybridization and washing have concluded, the microarray is exposed to ultraviolet light so that the emissions from the fluorescent Cy3 and Cy5 dyes can be scanned and registered. Spots whose corresponding sequences are more strongly expressed in the first sample will have more Cy3 dye present and thus will emit more intense green light. The same can be said for the second sample and the red Cy5 dye. Comparing the relative intensity of the green and red channels, it is possible to detect which genes have not been equally expressed in both samples. This can be used to hypothesize about the function of individual genes under many different conditions.

The first output of a microarray experiment is a pair of monochrome images, one for the green channel and another for the red channel. An example microarray image can be seen in Fig. 1. Due to the microscopic size of the spots, images have a high spatial resolution, and thus are of large dimensions. Images from 1000×1000 onwards are described in the literature, and sizes over 2200×2200 are common. Since gene expression can vary in a very wide range, image pixel intensities have a depth of 16 bpp (bits per pixel).

Microarray images are computer analyzed to obtain the contained genetic information. However, it is not desirable to only keep this genetic information and discard the microarray images. Analysis techniques are not fully mature or universally accepted, and are subject to changes. Furthermore, repeating an experiment is expensive and not always possible. Depending on the microarray size and the scanner spatial resolution, raw data for a single channel can require from a few to tens of Megabytes. With the increasing interest in DNA microarrays, and since many experiments are conducted under several different conditions, great amounts of data are generated in laboratories around the world. Because of the need of keeping and sharing microarray images, a necessity for efficient storage and transmission methods arises, and so compression emerges as a natural approach. The role of data compression in computational biology and the state of the art has been addressed by several authors [4], [5].

Both lossy and lossless techniques have been proposed in the literature. Lossy approaches exhibit good compression performance on microarray images, but there is still an open debate on whether information loss is acceptable or not, since it can alter genetic information extraction results. On the other hand, pure lossless methods guarantee immutable extraction results but offer poorer compression performance.

978-0-7695-4528-8/11 $26.00 © 2011 IEEE
DOI 10.1109/CCP.2011.21

IEEE computer society

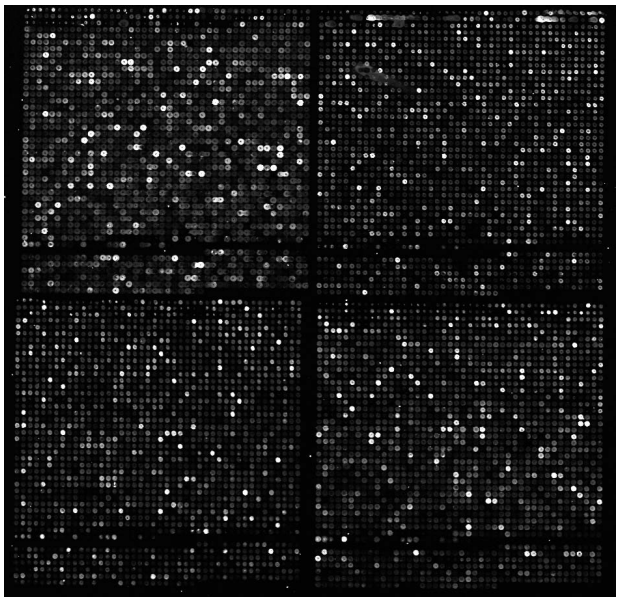

Figure 1. Modified portion (940 × 910) of red channel image y744n43_ch2.tif from the Yeast dataset. Gamma levels have been adjusted for better viewing.

This is partly due to the considerable amount of noise and the abundance of high frequencies present in this type of image.

This paper is structured as follows. In Section II we review compression schemes specific for microarray images, including the most recent ones. In Section III we summarize lossless compression results reported in the literature for these microarray-specific techniques, and compare them to results achieved by standard compression techniques in experiments that we have conducted. Section IV draws some conclusions.

II. Microarray-specific compression techniques

Image compression processes typically comprise up to 5 stages: preprocessing, transform, quantization, entropy coding and postprocessing. Microarray image compression can be modeled likewise. Relevant contributions for microarray image compression published in the literature are reviewed in Subsections II-A to II-F, depending on the stages where they make their contribution. This extends Luo and Lonardi's 2005 article by addressing newer techniques that have appeared since then, and providing a more complete and structured description [6].

A. Preprocessing

The preprocessing stage comprises any computation performed on an image to prepare it for the compression and analysis process. It is very important in DNA microarray images because many of the existing techniques rely heavily on the results of this stage to obtain competitive coding performance and to extract accurate genetic information. Denoising and segmentation are the main preprocessing techniques applied to this type of image, and are described next.

1) Denoising: Microarray images contain variable amounts of noise, which is sometimes considerable. This is due to irregularities in the conducting of the experiment and also in the image digitalization process. There has been a lot of research on denoising, but most of it is focused on the microarray analysis process and not on compression.

In 2006, Adjeroh et al. published their approach [7] based on the translation invariant (TI) transform. They argue that wavelet based denoising techniques that are not translation invariant suffer from an adverse pseudo-Gibbs phenomenon at discontinuities. This is specially important in microarray images since the edge of each spot can be considered a discontinuity. In their work, the original image is shifted horizontally, vertically and diagonally, and each of these shifted images plus the original one are denoised separately. The resulting images are then shifted back and combined to obtain the denoised image. Adjeroh et al. proposed three different approaches for combining the deshifted images: TI-hard, TI-median and TI-median2. The TI-hard method consists of outputting the average value of each pixel in the four deshifted images. The TI-median technique outputs the median instead of the average. The TI-median2 technique works by applying TI-median twice.

Many more authors have covered microarray image denoising for genetic information extraction, but only some of the most recent and relevant are referenced here. In 2005, Lukac proposed a method [8] based on fuzzy logic and local statistics for noise removal. Also in 2005, Smolka et al. discussed the peer group concept [9] as a means to remove impulsive noise. In 2007, Chen and Duan presented a simple method for denoising microarray images [10] based on comparing the edge features of the red and green channels. Recently, in 2010, Zifan et al. have designed multiwavelet transformations [11] to denoise images.

2) Segmentation: Segmentation, also know as spot finding or addressing, consists in determining which of the image pixels belong to spots (i.e., the foreground), as opposed to those that do not (i.e., the background). As it will be later discussed in Section II-D1, this can be very useful in later stages of the compression process. For example, it is possible to code separately foreground and background or to exploit differences in pixel intensity distributions between sets. We consider here only specific approaches oriented towards image compression, and omit techniques focused on general microarray image analysis.

In 2003, Faramarzpour et al. proposed a lossless coder whose segmentation stage consists of two steps. First, spot regions are located by studying the period of the signal obtained from summing the intensity by rows and by columns, and studying its minima. After that, spot centers are esti-

mated based on the region centroid, which is needed for their spiral scanning procedure, as explained in Section II-D1. Simpler versions of this spot region location idea had already been used in microarray image analysis and also in Jörnsten and Yu's work in 2002 [12], where they proposed a lossy-to-lossless compression scheme. In their technique, a seeded region growing algorithm is used to obtain a coarse mask for the spot pixels, which is then refined.

Later, in 2004, Lonardi and Luo presented their MicroZip lossy or lossless compression software [13]. They used a variation of Faramarzpour's spot region finding idea, but they considered the existence of subgrids which are located before spot regions. Four subgrids can be appreciated in Fig. 1, but other images may contain more.

In 2004, Hua et al. proposed a lossy or lossless scheme [14] with a segmentation technique based on the Mann-Whitney U test, which helps in deciding whether two independent sets of samples have equally large values. Once a region containing a spot has been located, a default mask is applied to obtain an initial set of pixels classified as spot pixels, and then the test is applied to add or remove pixels from this mask. This had already been used in microarray image segmentation [15], but the authors proposed a variation that speeds up the algorithm up to 50 times.

Also in 2006, Bierman et al. described a pure lossless compression scheme [16] with a simple threshold method for dividing microarray images in low and high intensities. It consists of determining the lowest of the threshold values from 2^8, 2^9, 2^{10} or 2^{11} such that approximately 90% of the pixels fall within it.

In 2007, Neekabadi et al. proposed another threshold-based technique for segmentation [17], this time in three subsets: background, edge and spot pixels. Their lossless proposal performs segmentation in two steps. First, they determine the optimal threshold value by minimizing the total standard deviation of pixels above and below it. Then they segment the image in the mentioned subsets. To do so, first they determine the spot pixels by eroding the mask formed with pixels above the selected threshold. Edge pixels are the ones surrounding the spot pixels, and background pixels are all the others.

Finally in 2009, Battiato and Rundo published an approach [18] based on Cellular Neural Networks (CNNs). They define two layers for their lossless system, each with as many cells as the image has pixels. The input and state of the first layer are the pixels of the original image, its output is the input and state of the second layer. By defining the cloning templates that drive the whole CNN dynamic, the second layer tends to its saturation equilibrium state and the resulting output tends to a "local binary image" where spot pixels tend to 1 and background pixels to 0.

B. Transform

The transform stage consists of changing the image domain from the spatial domain to another domain where it can be more efficiently processed or coded. Examples of this are applying the DCT to obtain a frequency representation, or using a wavelet transform to change to the spatial-frequency domain.

However, transform based compression is not typically as efficient for microarray images as it is for other types of images because of high frequencies due to the presence of abundant noise and thousands of spots. For this reason, transformations are not frequently researched in microarray image compression, although they are used in some works. Since these papers provide little or no original contribution on the transformation stage, to avoid redundancy they are not discussed in this section but only referenced.

In 2004, Hua et al. [14] published a modification of the EBCOT algorithm that included a tailored integer odd-symmetric transform for their lossy or lossless scheme (see Section II-D1). In 2004, Lonardi and Luo [13] made use of the Burrows-Wheeler transform [19] for lossy or lossless compression in their MicroZip software (see Section II-D1). In 2006, Adjeroh et al. used a variation of the TI transform [7] for denoising (see Section II-A1). In 2007, Peters et al. [20] applied a slightly modified version of the SVD in their lossy compression scheme (see Section II-E). In 2010, Zifan et al. [11] used multiwavelet transformations, also for denoising (see Section II-A1). In 2011, Avanaki et al. [21] tested the use of an existing wavelet transform before applying fractal lossy compression (see Section II-E).

C. Quantization

The stage of quantization consists of assigning a range of values to a single quantized value, effectively reducing the total number of symbols needed to be represented and thus increasing compressibility, at the expense of introducing information loss. In microarray image compression literature, there are almost no original contributions for the quantization stage, partly because information loss is not always acceptable. There are however two exceptions.

In 2000 and in 2003 Jörnsten et al. [22], [23] proposed both scalar and L1-norm vector adaptive quantizers (see Section II-D1), that can be used in lossy or lossless compression. In 2007, Peters et al. [20] used simple truncation in their lossy SVD-based technique (Section II-E).

D. Entropy coding

In this stage of the image compression process, data obtained from previous stages are expressed in an efficient manner to generate a more compact bitstream.

DNA microarray images show a strong spatial regularity, and this has been used in most techniques present in the literature. Many of them segment the image in foreground (spot) and background pixels and code them separately.

Others build contexts or try to predict the intensity of the next pixels based on the previous ones, sometimes after segmenting the image.

Ideas following each of these patterns are discussed in the next two subsections. Some of the works could be classified in both of these groups. Here they have been assigned to the one that, in our opinion, is more important for the algorithm.

1) Segmentation based coding: DNA microarray images are usually segmented in spot and background pixels as part of the preprocessing stage, and always when extracting its genetic information. Particular specific segmentation proposals have been discussed in Section II-A2. Several techniques that exploit this segmentation in their coding stage are presented next.

In 2002 and in 2003 Jörnsten et al. [12], [22] presented a lossy-to-lossless compression scheme called SLOCO, a version of the LOCO-I algorithm, the basis of the JPEG-LS standard. After griding and segmentation, the image is divided in multiple rectangular subblocks, one for each spot. In this way, each spot can be accessed and sent independently and with different quality. For each spot subblock, two subimages are created: one where the background has been set to the spot mean value (the spot subimage), and another where the foreground has been set to the estimation of the background value. Then each of the images is processed with SLOCO, which uses prediction in the spatial domain [24]. An important contribution of SLOCO is the use of an adaptive quantizer (UQ-adjust instead of UQ), that permits variable pixel-error δ so that spot pixels with higher intensities can be expressed with lower precision. Jörnsten et al. proposed both a scalar and a L1-norm vector quantizer (L1VQ), whose errors can be bitplane-encoded to obtain progressive lossy-to-lossless compression.

In 2003, Faramarzpour et al. presented a prediction-based lossless compression technique [25]. The image is gridded in rectangular subblocks, and spot centers are then estimated. For each of the subblocks, a spiral path is created to transform the 2D sequence into a 1D one. A linear prediction scheme that uses neighbor pixel intensities and their distances to the spot center is then applied on the 1D sequence. Differences between consecutive prediction errors form a sequence that is adaptive Huffman coded after being split on the index that minimizes the expected length of the coded sequences.

In 2004, Hua et al. presented microarray BASICA software [14] and proposed a progressive lossy-to-lossless compression scheme. In their work, they first grid and segment the image. Then they separate each of the subblocks into foreground and background and then code them with a modified version of the EBCOT algorithm, the basis of the JPEG2000 standard [26]. The main modification to EBCOT is an adaptation of the original context modeling, which allows a better handling of the irregular shapes of the foreground and background subimages. Bit shifts are also performed when coding the foreground so that the most relevant information is sent first in this progressive scheme.

Also in 2004, Lonardi and Luo presented their MicroZip software [13], which offers lossless or lossy compression. In their work, the image is first gridded and segmented in foreground and background. Then each of the 16-bit streams is divided into two 8-bit substreams comprising the most and least significant bytes. The four resulting substreams are losslessly coded except for the LSB of the background, which can be either lossless or lossy compressed. Lossless coding is done with the help of the Burrows-Wheeler Transform [19], originally designed for text compression, which reduces the total entropy by computing all permutations of a given channel and then sorting them lexicographically. Lossy coding is done with the SPIHT algorithm [27].

In 2006, Bierman et al. presented their lossless compression MACE (Micro Array Compression and Extraction) software [16]. In their work, high and low intensity pixels are separated using a simple threshold-based method to exploit the fact that intensity distribution in microarray images is very skewed. One image is generated for the low intensity pixels, and another for the high intensity ones. In the low intensity image, high intensity pixels are set to zero, and vice versa. The low intensity image is then losslessly coded using dictionary-based techniques such as Gzip or LZW, after being split in two subimages consisting of the most and least significant bytes of the original image, respectively. The high intensity image is processed with a sparse matrix algorithm, and then compressed. Later in 2007, Bierman et al. studied the performance impact of varying the dictionary size in their compression techniques [28]. They concluded that compression improved up to a certain dictionary size, where the performance stopped improving and began degrading.

In 2009, Battiato and Rundo published a lossless compression algorithm [18] based on image segmentation and color reindexing. As previously discussed, segmentation is made by means of a CNN-based system and produces two complementary subimages. The foreground is compressed with a generic lossless algorithm and stored separately. The background is first transformed into an indexed image. Then its color palette is reindexed with an algorithm that reduces the zero-order entropy of local differences, which are losslessly coded. The reindexing algorithm had been previously presented by the authors in 2007 [29].

2) Context-based coding: Contexts are used in image compression because they allow a more precise estimation of the occurrence probabilities of each symbol, which results in a reduction of the total entropy and thus of the compressed bitstream size.

In 2005 and in 2006 Zhang et al. [7], [30] proposed a context-based lossless approach which also employed segmentation. In their work, they define a mixture model for microarray images where they consider two structural components (foreground and background) and assign prob-

abilities based on the gamma distribution. Considering this model, they divide the image into two streams (foreground and background) and then each of those into two substreams, one for the most significant byte and the other for the least significant byte. MSB substreams are then processed by a simple predictive scheme, but LSB substreams are not. These four substreams are then coded with prediction by partial approximate matching (PPAM), a lossless compression technique also proposed by Zhang and Adjeroh [31]. In this paper, multicomponent compression is briefly addressed by compressing first one channel, I_r, and then the pixel by pixel difference, $I_D = I_r - I_g$, obtaining slightly better results than compressing I_r and I_g separately.

In 2006, Neves and Pinho [32] proposed another context-based lossless approach. It is a bitplane-based technique that uses 3D finite-context models to drive an arithmetic coder. The most significant bitplane is encoded first, with a causal context formed by four surrounding pixels, and bitplanes from the second to the eighth most significant bitplanes are encoded using bits from the one being encoded and from the ones previously coded. Finally, the 8 least significant bitplanes are coded using only bits from the previous bitplanes for the context model. The probabilities used to drive the arithmetic coder are based on the number of times that a given symbol has appeared in the image while in a given context. The average coding bitrate for each bitplane is monitored and whenever one shows an expansive behavior (more than 1 bpp), that bitplane and the following bitplanes are not arithmetically coded, but simply output raw. Neves and Pinho used a trial and error procedure to build these context templates, which are the same for every image. In 2009, they extended this procedure so that specific templates are built for each image using a greedy approach, obtaining better results [33].

E. General techniques applied to microarray images

Several authors have considered adapting generic image compression algorithms to DNA microarray images, as discussed previously. Others have opted to apply them directly, as described next.

In 2007, Peters et al. presented a lossy compression method [20] based on singular value decomposition (SVD). More recently, in 2011, Avanaki et al. [21] used fractal and wavelet-fractal lossy compression techniques on microarray images.

F. Postprocessing

After compression, general images are sometimes processed to enhance their visual quality or to provide new features. DNA microarray images are not usually post-processed with these goals, but instead they are analyzed in order to extract genetic information. Because of this, traditional quality measures such as MSE or PSNR may not be completely suitable when performing lossy compression on DNA microarray images. Inspired by the manner in which genetic information is extracted, some authors have proposed specific distortion measures for lossy microarray image compression, and have measured the performance of some of the discussed algorithms. The ideas on which these measures are sustained are described next.

1) Spot detection: Spot detection consists of labeling spots as valid or invalid depending on a measure of the reliability of the extracted information. When performing lossy compression, this labeling might be affected, and thus one can define a distortion measure based on the number of differences in the classification [14].

2) Spot identification: Spot identification consists of determining whether a particular gene is being expressed with higher, lower or the same intensity in two samples that correspond to the two channels of a typical DNA microarray image. This is usually done by comparing pixel intensity properties of these two channels for each valid spot, that is, with product intensity over a given threshold β [34]. Lossy compression can affect these pixel intensity properties, and thus one can define a distortion measure by counting the number of differences in the classification after compression [14], [22], or even a quantitative difference between intensity logarithms [14].

3) Spot classification: Once spots have been labeled in the identification step, it is also possible to apply one or more classification or clustering algorithms and determine the discrepancy between an original and a lossy compressed image to obtain new distortion measures. Hierarchical clustering and k-means are the most widely used algorithms for this purpose [7], but expression based classification has also been proposed [34].

4) Distortion results: There are not many published surveys that study measured distortion, and the existing ones consider mostly generic image compression techniques like SPIHT and JPEG2000. The only specific algorithm for DNA microarray images studied is SLOCO, Jörnsten's algorithm [22], [34], [35].

Even so, all authors that discuss distortion measures agree on the fact that lossy compression, even at really low bitrates (high compression), affect these measures in a very limited way [7], [22], [34]. Some claim that the variability induced by the lossy compression process is lower than that introduced when replicating an experiment [22], and even that lossy compression may improve the quality of the extracted information [7].

G. Technique summary

All techniques reviewed above are summarized in Table I according to the subsections where they have been discussed. They have been sorted chronologically and marked as lossless, lossy, or both.

Table I
MICROARRAY-SPECIFIC TECHNIQUES DISCUSSED IN THIS DOCUMENT. PURE LOSSY METHODS ARE MARKED WITH BROWN AND ×. PURE LOSSLESS WITH BLUE AND □. LOSSY AND LOSSLESS WITH GREEN AND ☒.

Preprocessing		**Transform**	**Quantization**	**Entropy coding**		**Generic**	**Postprocessing**
Denoising	Segmentation			Segmentation	Context		
× [8], 2005	☒ [12], 2002	☒ [14], 2004	× [23], 2000	☒ [12], 2002	□ [30], 2005	× [20], 2007	☒ [22], 2003
× [9], 2005	□ [25], 2003	☒ [13], 2004	☒ [22], 2003	☒ [22], 2003	□ [7], 2006	× [21], 2011	☒ [14], 2004
× [7], 2006	☒ [14], 2004	□ [7], 2006	× [20], 2007	□ [25], 2003	□ [32], 2006		× [7], 2006
× [10], 2007	☒ [13], 2004	× [20], 2007		☒ [14], 2004	□ [33], 2009		☒ [34], 2009
× [11], 2010	□ [16], 2006	× [11], 2010		☒ [13], 2004			
	□ [17], 2007	× [21], 2011		□ [16], 2006			
	□ [18], 2009			□ [28], 2007			
				□ [29], 2007			
				☒ [18], 2009			

III. RESULTS COMPARISON

In this section we show lossless compression results for generic and microarray-specific techniques. We analyze and compare those results to determine their relative compression performances for DNA microarray images.

We first present image sets that have been used for benchmarking in the literature in Subsection III-A. Reported compression results for microarray-specific techniques are shown on Subsection III-B. We have performed experiments with standard compression schemes as well. Our results are reported and compared to the ones provided by microarray-specific methods in Subsection III-C.

Lossy compression results are not discussed in this document because it is not yet clear what an admissible information loss is, and also because published papers do not generally provide data tables that allow homogeneous comparisons among the different approaches.

A. Image sets

Several different image datasets have been used for benchmarking in microarray image compression, but none is common across all of the publications. In some papers, images used are not specified. In others, only the source of the images is mentioned, but no other information about their size or characteristics is disclosed. Different datasets described and referenced in the literature are presented in Table II. Information about the number of images that they contain and their approximate size is also provided.

Table II
IMAGE SETS REFERENCED IN THE LITERATURE. ALL IMAGES ARE 16 BPP.

Dataset	**Images**	**Size (px)**
MicroZip [36]	3	$> 1800 \times 1900$
Yeast [37]	109	1024×1024
ApoA1 [38]	32	1044×1041
ISREC [39]	14	1000×1000

We also obtained 20 large images (over 2000×2000) from the Stanford Microarray Database public FTP at ftp://smd-ftp.stanford.edu/pub/smd/transfers/Jenny. We will refer to those images as the Stanford image set.

B. Microarray-specific techniques results

Results from lossless compression schemes described in Section II are presented in Table III. Techniques are listed chronologically, oldest first. Values for each algorithm and dataset are taken directly from the original papers. Dashes mean results were not provided for a given image set, and the *Unspecified* column is used when the image set is not revealed by the authors. Results are expressed in bits per pixel (bpp), so lower is better. Original images are all 16 bpp.

No single image set has been uniformly used for benchmarking in all techniques, so it is difficult to compare performance fairly. MicroZip, ApoA1 and ISREC are the sets that have been employed more frequently. Attending only to the results on these three corpora, Battiato's method based on CNNs performs best in all three with compression ratios of 8.619 bpp, 9.52 bpp and 9.49 bpp, respectively. Jörsten's SLOCO claims better results for the ApoA1 set, but cannot be consistently compared because of the lack of data on the other sets. A lower bound of 8 bpp is believed to exist for microarray images due to the presence of random noise in the least significant bitplanes. However, some authors have been able to obtain slightly better results using high order contexts [30].

C. Standard techniques results

It is important to compare microarray-specific techniques with generic compression techniques, especially standard techniques, in order to estimate the benefits of developing new microarray-specific techniques. In 2006, Pinho et al. [40] compared the performance of lossless JPEG2000, JBIG and JPEG-LS on the MicroZip, ApoA1 and ISREC image sets. We have conducted our own set of experiments and have been able to verify and extend their results. In addition, we have also used the Yeast set (with over 100

Table III
COMPARISON OF LOSSLESS MICROARRAY-SPECIFIC SCHEMES ON 16 BPP IMAGES. ALL RESULTS ARE EXPRESSED IN BITS PER PIXEL (BPP), SO LOWER IS BETTER. ALL RESULTS ARE TAKEN FROM THE REFERENCES SPECIFIED IN THE TABLE.

Algorithm	Year	MicroZip	Yeast	ApoA1	ISREC	Unspecified
SLOCO [12]	2002	—	—	8.556	—	—
Faramarzpour [25]	2003	—	—	—	—	9.091
Hua [14]	2004	—	—	—	—	6.985
MicroZip [13]	2004	9.843	—	—	—	—
PPAM [30]	2005	9.587	6.601	—	—	—
MACE [16]	2006	—	—	—	—	7.070
Neves [32]	2006	8.840	—	10.280	10.199	—
Neekabadi [17]	2007	8.856	—	10.250	10.202	—
Battiato [18]	2009	8.369	—	9.52	9.49	—
Neves [33]	2009	8.619	—	10.194	10.158	—

images), and the Stanford set (with 20 images) which had not been mentioned in Pinho's work. We have tested new algorithms and compression modes as well. Our results are presented and discussed next.

1) General compression schemes: Results for general compressors (not image compressors) are shown in Table IV. Results for each method and dataset are obtained averaging the resulting bitrates for compressing independently the image files in raw format of all images in a set. All compressors have been invoked in best compression mode. Best results in all datasets are obtained with Bzip2, which outperforms XZ (LZMA2) by 5.10% in the Yeast set, and less in the other sets.

2) Image compression standards: In our experiments we have tested the performance of standard image compression schemes as well. As in Pinho's work, we have evaluated lossless JPEG2000, JBIG and JPEG-LS. In addition, we have examined results for CALIC and different modes for lossless JPEG2000. While Pinho et al. tested only the jj2000 implementation using the default DWT transform with 5 levels of decomposition, we have also evaluated the performance with 0 to 5 decomposition levels with both the jj2000 and Kakadu implementations. Our results are provided in Table V. As for the general compression schemes, results are obtained by averaging the resulting bitrate for compressing the images independently.

We have obtained essentially identical results as in Pinho et al.'s work for the three standards on which they have reported. In addition, we have observed that the Kakadu implementation of lossless JPEG2000 yields worse results when compressing microarray images as the number of decomposition levels increases. When no transformation is applied with the Kakadu implementation, a bitrate reduction of 52.88% is observed for the Yeast set when compared to the 5 level Kakadu implementation, while reductions under 6.5% are obtained for the other data sets. The jj2000 implementation only improves its performance for 2 of the 5 sets when no transformation is applied, and shows a bitrate reducion of 32.31% for the Yeast set. For every image set and number of decomposition levels, the Kakadu implementation of JPEG2000 retrieves better results than the jj2000 implementation.

CALIC and JPEG-LS prediction-based schemes perform better than all the other non-specific schemes on all image sets except for the Yeast set, where Kakadu lossless JPEG2000 with 0 decomposition levels significantly outperforms CALIC (43.96%). The best image compressor for each set is between 2.86% and 5.25% better than Bzip2, the best generic compressor. Battiato's algorithm performs 12.81%, 10.45% and 11.85% better than the best non-microarray-specific technique for the MicroZip, ApoA1 and ISREC sets, respectively.

IV. CONCLUSION

DNA microarrays are state-of-the art tools widely used in biology and medicine. Analysis of microarray images is still being developed and repeating experiments is expensive or impossible, so storing them for re-analysis is necessary.

At least 10 compression schemes specific for microarray image compression have been proposed in the literature. We have classified them according to the stage or stages of the image compression process that they contribute most to, and have described their most relevant ideas.

Lossless compression results for both microarray-specific and standard image compression methods and the datasets used for benchmarking in the literature have been discussed. The best microarray-specific technique reported for a variety of data sets is Battiato's CNN-based proposal. According to our experiments, the best standard lossless image compressors are the prediction-based algorithms CALIC and JPEG-LS, except for the Yeast set, where lossless JPEG2000 with zero decomposition levels clearly outperform the other techniques. The best image compressor for each set is between 2.86% and 5.25% better than Bzip2 and Battiato's algorithm performs 12.81%, 10.45% and 11.85% better than the best non-microarray-specific technique for the three image sets for which data from most algorithms exist.

Table IV
RESULTS IN BPP FOR GENERAL ENTROPY CODERS AND INDIVIDUAL IMAGE COMPRESSION. ALL IMAGES ARE 16 BPP.

Algorithm	MicroZip	Yeast	ApoA1	ISREC	Stanford
Gzip	11.736	7.548	12.711	12.462	9.813
AC (word)	11.269	7.688	12.531	12.011	9.564
XZ (LZMA2)	10.140	6.385	11.321	11.015	8.163
Bzip2	9.841	6.075	11.067	10.921	7.867

Table V
RESULTS IN BPP FOR GENERIC IMAGE COMPRESSORS AND INDIVIDUAL IMAGE COMPRESSION. ALL IMAGES ARE 16 BPP.

Algorithm	MicroZip	Yeast	ApoA1	ISREC	Stanford
CALIC	9.582	8.502	10.515	10.615	7.592
JBIG	9.747	6.888	10.852	10.925	7.776
JPEG-LS	9.441	8.580	10.608	11.145	7.571
jj2000 (0 levels)	10.455	6.862	11.569	10.929	8.581
jj2000 (1 level)	10.016	9.123	11.142	11.521	8.160
jj2000 (2 levels)	9.976	9.092	11.068	11.376	8.039
jj2000 (3 levels)	9.971	9.081	11.062	11.363	8.004
jj2000 (4 levels)	9.968	9.079	11.063	11.365	7.992
jj2000 (5 levels)	9.967	9.079	11.063	11.366	7.990
Kakadu (0 levels)	9.587	5.906	10.784	10.627	7.666
Kakadu (1 level)	9.739	8.851	10.902	11.229	7.834
Kakadu (2 levels)	9.862	8.988	10.970	11.276	7.910
Kakadu (3 levels)	9.906	9.019	10.992	11.297	7.940
Kakadu (4 levels)	9.915	9.027	11.000	11.310	7.946
Kakadu (5 levels)	9.917	9.029	11.002	11.314	7.948

ACKNOWLEDGMENTS

MicroZip corpus was kindly provided by Neves and Pinho.

This work has been partially supported by the European Union, by the Spanish Government (MICINN), by FEDER, and by the Catalan Government, under Grants FP7-PEOPLE-2009-IIF FP7-250420, TIN2009-14426-C02-01, FPU AP2007-01555, and 2009-SGR-1224.

REFERENCES

[1] S. Moore, "Making chips to probe genes," *IEEE SPECTRUM*, vol. 38, no. 3, pp. 54–60, MAR 2001.

[2] M. S. Giri, M. Nebozhyn, L. Showe, and L. J. Montaner, "Microarray data on gene modulation by HIV-1 in immune cells: 2000-2006," *JOURNAL OF LEUKOCYTE BIOLOGY*, vol. 80, no. 5, pp. 1031–1043, NOV 2006.

[3] S. Satih, N. Chalabi, N. Rabiau, R. Bosviel, L. Fontana, Y.-J. Bignon, and D. J. Bernard-Gallon, "Gene Expression Profiling of Breast Cancer Cell Lines in Response to Soy Isoflavones Using a Pangenomic Microarray Approach," *OMICS-A JOURNAL OF INTEGRATIVE BIOLOGY*, vol. 14, no. 3, pp. 231–238, JUN 2010.

[4] O. U. Nalbantoglu, D. J. Russell, and K. Sayood, "Data compression concepts and algorithms and their applications to bioinformatics," *Entropy*, vol. 12, pp. 34–52, 2010.

[5] R. Giancarlo, D. Scaturro, and F. Utro, "Textual data compression in computational biology: a synopsis," *Bioinformatics*, vol. 25, no. 13, pp. 1575–1586, 2009.

[6] Y. Luo and S. Lonardi, "Storage and transmission of microarray images," *Drug Discovery Today*, vol. 10, no. 23-24, pp. 1689 – 1695, 2005.

[7] D. A. Adjeroh, Y. Zhang, and R. Parthe, "On denoising and compression of DNA microarray images," *Pattern Recognition*, vol. 39, no. 12, pp. 2478–2493, December 2006.

[8] R. Lukac, K. Plataniotis, B. Smolka, and A. Venetsanopoulos, "A data-adaptive approach to cDNA microarray image enhancement," in *Computational Science - ICCS 2005, PT 2*, ser. LECTURE NOTES IN COMPUTER SCIENCE, vol. 3515. SPRINGER-VERLAG BERLIN, 2005, Proceedings Paper, pp. 886–893.

[9] B. Smolka and K. Plataniotis, "Ultrafast technique of impulsive noise removal with application to microarray image denoising," in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds., vol. 3656. SPRINGER-VERLAG BERLIN, 2005, Proceedings Paper, pp. 990–997.

[10] X. Chen and H. Duan, "A vector-based filtering algorithm for microarray image," in *2007 IEEE/ICME International Conference on Complex Medical Engineering*, vol. 1-4. IEEE, 2007, Proceedings Paper, pp. 794–797.

[11] A. Zifan, M. H. Moradi, and S. Gharibzadeh, "Microarray image enhancement by denoising using decimated and undecimated multiwavelet transforms," *Signal Image and Video Processing*, vol. 4, no. 2, pp. 177–185, June 2010.

[12] R. Jornsten, Y. Vardi, and C. hui Zhang, "On the bitplane compression of microarray images," in *in Proc. of the 4th Int. L1-norm Conf*, 2002.

[13] S. Lonardi and Y. Luo, "Gridding and compression of microarray images," in *2004 IEEE Computational Systems Bioinformatics Conference, Proceedings*. IEEE, 2004, Proceedings Paper, pp. 122–130.

[14] J. Hua, Z. Liu, Z. Xiong, Q. Wu, and K. Castleman, "Microarray basica: Background adjustment, segmentation, image compression and analysis of microarray images," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 1, pp. 92–107, January 2004.

[15] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cdna microarray images," *Journal of Biomedical Optics*, vol. 2, pp. 364–374, October 1997.

[16] R. Bierman, N. Maniyar, C. Parsons, and R. Singh, "MACE: lossless compression and analysis of microarray images," in *Proceedings of the 2006 ACM symposium on Applied computing*, ser. SAC '06. ACM, 2006, pp. 167–172.

[17] A. Neekabadi, S. Samavi, S. A. Razavi, N. Karimi, and S. Shirani, "Lossless microarray image compression using region based predictors," in *2007 IEEE International Conference on Image Processing*, vol. 1-7. IEEE, 2007, Proceedings Paper, pp. 913–916.

[18] S. Battiato and F. Rundo, "A bio-inspired CNN with reindexing engine for lossless dna microarray compression and segmentation," in *2009 16th IEEE International Conference on Image Processing*, vol. 1-6, IEEE. IEEE, 2009, Proceedings Paper, pp. 1717–1720.

[19] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm." HP, Tech. Rep. 124, 1994.

[20] T. J. Peters, R. Smolikova-Wachowiak, and M. P. Wachowiak, "Microarray image compression using a variation of singular value decomposition," in *2007 Annual International Conference of the IEE Engineering in Medicine and Biology Society*, vol. 1-16. IEEE, 2007, Proceedings Paper, pp. 1176–1179.

[21] M. R. N. Avanaki, A. Aber, and R. Ebrahimpour, "Compression of cDNA microarray images based on pure-fractal and wavelet-fractal techniques," *ICGST International Journal on Graphics, Vision and Image Processing, GVIP*, vol. 11, pp. 43–52, March 2011.

[22] R. Jornsten, W. Wang, B. Yu, and K. Ramchandran, "Microarray image compression: Sloco and the effect of information loss," *Signal Processing*, vol. 83, no. 4, pp. 859–869, April 2003.

[23] R. Jornsten and B. Yu, ""comprestimation": Microarray images in abundance." in *2000 Conference on Information Sciences and Systems*, P. University, Ed., 2000.

[24] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The loco-i lossless image compression algorithm: Principles and standardization into jpeg-ls," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.

[25] N. Faramarzpour, S. Shirani, and J. Bondy, "Lossless DNA microarray image compression," in *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, vol. 2, November 2003, pp. 1501–1504.

[26] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Boston, 2002.

[27] A. Said, W. A. Pearlman, and S. Member, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, 1996.

[28] R. Bierman and R. Singh, "Influence of dictionary size on the lossless compression of microarray images," in *Twentieth IEEE International Symposium on Computer-Based Medical Systems, Proceedings*, P. Kokol, V. Podgorelec, D. Micetic-Turk, M. Zorman, and M. Verlic, Eds. IEEE, 2007, Proceedings Paper, pp. 237–242.

[29] S. Battiato, F. Rundo, and F. Stanco, "Self organizing motor maps for color-mapped image re-indexing," *Image Processing, IEEE Transactions on*, vol. 16, no. 12, pp. 2905 –2915, December 2007.

[30] Y. Zhang, R. Parthe, and D. Adjeroh, "Lossless compression of DNA microarray images," in *Computational Systems Bioinformatics Conference, 2005. Workshops and Poster Abstracts. IEEE*, August 2005, pp. 128 – 132.

[31] Y. Zhang and D. Adjeroh, "Prediction by partial approximate matching for lossless image compression," *IEEE Transactions on Image Processing*, vol. 17, no. 6, pp. 924 –935, June 2008.

[32] A. J. R. Neves and A. J. Pinho, "Lossless compression of microarray images," in *2006 IEEE International Conference on Image Processing, ICIP 2006, Proceedings*. IEEE, 2006, Proceedings Paper, pp. 2505–2508.

[33] A. J. R. Neves and A. J. Pinho, "Lossless compression of microarray images using image-dependent finite-context models," *IEEE Transactions on Medical Imaging*, vol. 28, no. 2, pp. 194–201, February 2009.

[34] Q. Xu, J. Hua, Z. Xiong, M. L. Bittner, and E. R. Dougherty, "The effect of microarray image compression on expression-based classification," *Signal Image and Video Processing*, vol. 3, no. 1, pp. 53–61, February 2009.

[35] Q. Xu, J. Hua, Z. Xiong, M. Bittner, and E. Dougherty, "Accuracy of differential expression detection with compressed microarray images," in *Genomic Signal Processing and Statistics, 2006. GENSIPS '06. IEEE International Workshop on*, May 2006, pp. 43 –44.

[36] "MicroZip test image set." [Online]. Available: http://www.cs.ucr.edu/~yuluo/MicroZip

[37] "Stanford Yeast Cell-Cycle Regulation Project." [Online]. Available: http://genome-www.stanford.edu/cellcycle/data/rawdata/individual.html

[38] "ApoA1 experiment data." [Online]. Available: http://www.stat.berkeley.edu/users/terry/zarray/Html/apodata.html

[39] "ISREC image set." [Online]. Available: http://www.isrec.isb-sib.ch/DEA/module8/P5_chip_image/images/

[40] A. Pinho, A. Paiva, and A. Neves, "On the use of standards for microarray lossless image compression," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 3, pp. 563–566, March 2006.