DRUG DISCOVERY TODAY BIOSILICO

Storage and transmission of microarray images

Yu Luo and Stefano Lonardi

With the recent explosion of interest in microarray technology, massive amounts of microarray images are currently being produced. The storage and transmission of these types of data are becoming increasingly challenging. This article reviews the latest technologies that allow for the compression and storage of microarray images in dedicated database systems.

The first attempts at global analysis of gene expression were undertaken in the mid-1970s with studies of the hybridization of an mRNA pool with radioactively labeled cDNA. Interest in gene expression increased steadily during the 1980s, and in the 1990s a new era of high throughput gene expression studies unfolded with the development of microarrays [1,2]. Although microarrays are relatively new, their penetration has been phenomenal. Biologists and physicians have enthusiastically embraced this technology, and are currently producing an unprecedented quantity of microarray data. As of March 2005, for example, the Stanford microarray database (SMD) [3] contained sets of images for ~54,000 microarrays, relating to the biology of 35 organisms. The database has been growing exponentially since its inception.

Microarrays allow studies of gene expression on a massively parallel scale; a single microarray experiment can provide information on the expression of thousands of genes. More specifically, a microarray experiment is designed to compare the transcriptional activity of a set of genes under two conditions (hereafter called 'reference' and 'experimental'). The outcome is a quantitative measure of the relative change of expression of each gene in an experimental condition compared with the reference condition.

Currently, a few variants of microarray technology exist. To simplify the discussion in this review, we will describe the so-called cDNA microarrays. Proprietary technologies developed by Affymetrix and NimbleGen Systems employ somewhat different strategies – although the general principles remain the same. cDNA microarrays are fabricated on a glass or a nylon substrate by specialized high-speed robots. The fabrication process creates thousands of microscopic spots containing DNA probes that are immobilized in the substrate. The DNA probes are chosen to hybridize to unique sequences in the genes being studied.

The mRNA from the two different conditions is obtained separately and reverse transcriptase is used to transcribe the mRNA into cDNA. The cDNA is labeled with a green or red dye, depending on which conditions it corresponds to. The microarray chip is then exposed to a mix of the two populations of cDNAs. A given strand of cDNA will hybridize with the DNA probe that was selected from the gene that produced that transcript. The chip is then washed to remove any unbound cDNAs (see [1,2] for more details on the protocols). The microarray chip is finally scanned using a confocal laser or a charge-coupled device (CCD) to generate two digital images,

Yu Luo*
Stefano Lonardi**
Department of Computer Science & Engineering, University of California, Riverside, CA 92521, USA
*e-mail: yuluo@cs.ucr.edu
**e-mail: stelo@cs.ucr.edu

1359-6446/05/$ – see front matter PII: S1359-6446(05)03647-0

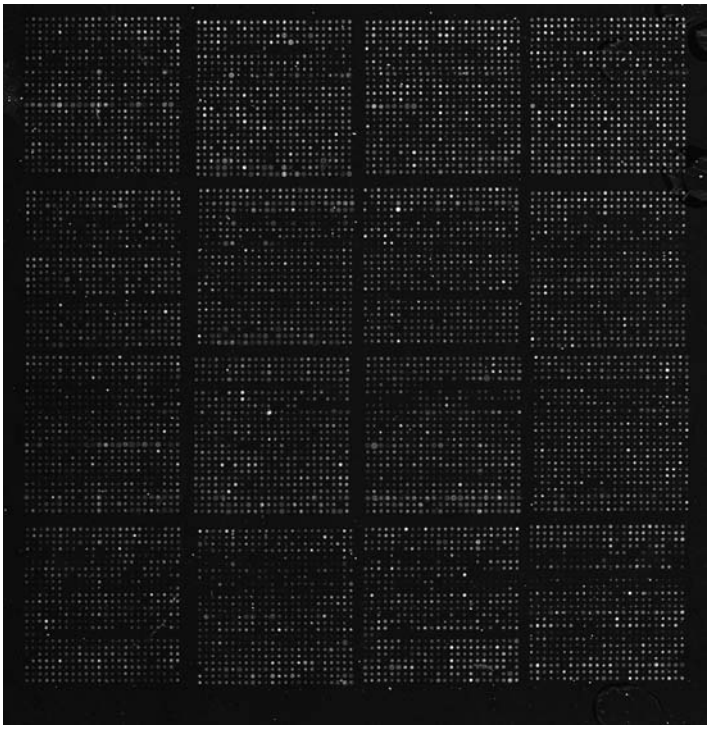

FIGURE 1

A typical microarray image, composed by 4x4 sub grids. Each sub grid is composed of a 24×24 matrix of spots. The original resolution of this image is 1872×1916 pixels, 16 bpp.

each corresponding to one of the colors [4,5]. The output of a single microarray experiment is a pair of 16 bits per pixel (bpp) digital images whose total size is typically in the tens of megabytes [6].

Microarray images are usually structured as a series of high intensity spots located on a regular grid. For example, Figure 1 shows a microarray with 16 subgrids for a total of ~9000 spots. A variety of methods and software tools are available to extract the gene expression information from microarray images (reviewed in [4]). However, there is still a debate in the scientific community about the accuracy, reliability and robustness of the analytical and statistical methods that need to be used. In practice, microarray images are challenging to process. For example, when the image contains high levels of noise, the detection of low-intensity spots is easily compromised. Because there is not yet a clear choice of a specific analytic method, it is believed crucial to keep the raw image data in some permanent storage medium [7]. Simply discarding the raw data and repeating the experiment is not an option because of the high costs associated with each experiment. Given the massive amount of data produced and the need of long-term storage and efficient transmission, *ad hoc* compression methods and dedicated database systems are currently an active area of research in computational biology.

Before turning the attention to the current research in microarray image compression, we report on the image compression formats employed by some of the microarray databases currently available (Table 1 is a summary of some commonly used image formats). The focus of this review is mainly on databases in the public domain because of the difficulties involved in finding technical specifications of proprietary platforms.

Storage of microarray images

The importance of organizing and storing the data of microarray experiments in relational databases cannot be overemphasized. Many microarray users are still struggling in the transition from spreadsheets to databases designed to handle the explosive growth of their microarray datasets. An international effort led by the Microarray Gene Expression Data (MGED) Society (www.mged.org) is under way to 'establish standards for microarray data annotation and exchange, facilitate the creation of microarray databases and related software implementing these standards, and promote the sharing of high quality, well annotated data within the life sciences community'. The MGED group has prepared standards on the 'minimum information about a microarray experiment' (MIAME) that requires all the information needed to interpret, share [8,9] and possibly replicate the results of a microarray experiments to be recorded in the database and made public [10–12]. A relational database system allows users to process and store the large quantities of data produced by microarray experiments and thereby accommodate the enforcements of standards like MIAME. Moreover, because microarray experiments typically depend on the work of several people in the laboratory, database systems can easily enforce common principles in data format and data entry [13]. To facilitate the exchange of information between databases, the MGED group has also designed a standard called microarray mark-up language (MAML).

A proliferation of microarray databases has occurred in response to the demands of the life science communities [14–16]. We have reviewed several databases both for local installation and public data repositories, and here we present our findings. Public data repositories allow multiple users to access the data remotely via a browser, whereas local installations allow access only through the machine in which the database is installed. Some databases are not designed to manage images but only gene expression data [e.g. ChipDB (http://chipdb.wi.mit.edu) or AMAD (www.microarrays.org)].

Local installation databases, such as GeneDirector (www.biodiscovery.com), mAdb (http://madb.niaid.nih.gov), maxdSQL (http://bioinf.man.ac.uk/microarray/maxd) and the SMD (http://genome-www5.stanford.edu) [3,17], store file images in TIFF format and allow users direct access to them. SMD allows images in GIF format to be obtained for viewing purposes or web posting and the mAdb database can export images in JPEG format for presentation. Among the databases for public data repositories, ArrayExpress (www.ebi.ac.uk/arrayexpress) [17–20], the RNA Abundance

TABLE 1

Common image formats

Image format	Extensions	Description
JPEG (joint photographic experts group)	.jpg, .jpeg, .jif and .jfif	JPEG is a lossy format. JPEG is most suitable for images with subtle color variations, such as a photograph. JPEG allows 24 bpp color or 8 bpp grayscale.
JPEG-LS	.jls	JPEG-LS is a lossless/near-lossless compression standard for JPEG. The standard is based on the LOCO algorithm. JPEG-LS allows up to 24bpp color or 16bpp grayscale.
JPEG 2000	.jp2, .jpx, .j2k and .j2c	New JPEG standard based on wavelets. It is mainly a lossy format and it provides a better compression ratio than JPEG. It also has a lossless mode. JPEG2000 allows up to 38bpp color or grayscale.
PNG (portable network graphics	.png	Official version support lossless compression only. Uses a two phase prediction and LZ-77. PNG allows up to 48 bpp true color or 16 bpp grayscale
TIFF (tagged image file format)	.tif and .tiff	TIFF supports storing multiple images in a single file. TIFF can be an uncompressed or compressed. If compressed, it can be compressed lossless with LZW or lossy with JPEG. TIFF allows up to 64 bpp color or 16 bpp grayscale.
GIF (graphics interchange format)	.gif	GIF is a lossless format internally compressed with LZW, but limited to 8 bpp color or grayscale.

Database (RAD; www.cbil.upenn.edu/rad) [21,22], SMD, and the National Center for Biotechnology Information (NCBI) Geo (www.ncbi.nlm.nih.gov/geo) [23,24] store links to TIFF images. NCBI Geo also allows users to link to JPEG images. (See Table 2 for summaries of the image formats supported by these databases.)

All the databases mentioned above claim to follow the MGED minimum information guidelines. AMAD is the only nonrelational database in the group. GeneDirector, maxdSQL and SMD use Oracle as the underlying database management system (DBMS), and mAdb uses Sybase DBMS. Some of these databases already allow the users to upload data in the MAML format (e.g. ArrayExpress). All the databases, with the exception of ChipDB, allow the storage of gene expression data obtained with cDNA microarrays. Only GeneDirector, maxdSQL, ArrayExpress, ChipDB, NCBI Geo and RAD also allow managing data generated with Affymetrix arrays.

The image format specifications of commercial platforms are harder to obtain. For example, the documentation of the Affymetrix GeneChip operating system (GCOS) mentions several intermediate data formats. Raw files generated by the Affymetrix proprietary chip scanning software are 16 bit TIFF files that are converted to 8 bit PNG or JPEG format for viewing. The feature extraction tool used by Agilent only supports uncompressed TIFFs (Agilent TIFF and standard TIFF) as an input file format. Agilent TIFF contains both channel images and also includes the grid and the protocol information. The output format supports JPEG for viewing and for downstream analysis packages.

As in many domains of application in computer science, the technologies currently used are several years behind what is proposed in the literature.

Compression of microarray images

Data compression is a crucial step in several technologies that rely on the storage or transmission of large quantities of data. Data compression methods can be divided into two disjoint classes depending on the quality of the decompressed file. If the decompressed file is identical to the original, the compression is termed lossless; if any loss of quality occurs, the compression is called lossy. For lossless files, the typical strategy to reduce the size of the file is to use an encoding scheme that exposes the regularities of the data (for example, repetitions or periodicities). For lossy files, the general idea is to use fewer bits to encode less important features of the image. The specification of what constitutes an 'important feature' (sometimes called a region of interest) depends on the domain of application. For example, to compress real-world photographs one could design an encoder that uses more bits to encode the edges of the objects and fewer bits to encode weak variations of colors.

In microarray images, the regions of interest are the subset of pixels that correspond to the spots (the foreground). The shape of the spots is approximately circular, although significant variations from this shape are possible because of experimental deviations in the spotting procedure. Perturbations on the spot position, irregularities in the spot shape and size, holes in spots, unequal distribution of the DNA probe within spots, variable background and global problems that affect multiple spots (like scratches, contamination, dust, etc.) are common. In general, the shape and the size of the spots can fluctuate significantly across the array. The background is usually much less crucial for the downstream image-processing step, although it is used as a reference. In cDNA microarrays, some of the labeled cDNA will attach to the substrate even where there are no DNA probes. The intensity of a spot must therefore be corrected by the intensity of the background in a nearby area [25].

It is well known that microarray images cannot be easily compressed with standard approaches. As a consequence of the microscopic size of the spots and the digitization process, microarray images contain a high level of noise. In addition, the region of the image carrying the useful information is composed of thousands of tiny spots that are placed close to each other. Transform-based

TABLE 2

A sample of microarray databases grouped by the format of the microarray images supported

Database type	Supported image format	Database name	URL
Local installation	**TIFF**	GeneDirector	www.biodiscovery.com
		GeneX	http://genex.sourceforge.net
		mAdb	http://madb.niaid.nih.gov
		maxdSQL	http://bioinf.man.ac.uk/microarray/maxd
		SMD	http://genome-www5.stanford.edu
	None	AMAD	www.microarrays.org
Online public data repositories	**TIFF**	ArrayExpress	www.ebi.ac.uk/arrayexpress
		GeneX	http://genex.sourceforge.net
		RAD	www.cbil.upenn.edu/RAD
		SMD	http://genome-www5.stanford.edu
	TIFF+JPEG	NCBI Geo	www.ncbi.nlm.nih.gov/geo
	None	ChipDB	http://chipdb.wi.mit.edu

image-compression methods (e.g. JPEG and JPEG 2000) have been designed for real-world images and they are known to perform poorly on images containing large numbers of sharp edges. In fact, microarray images are perhaps the worst possible type of image for these techniques. When transform-based methods are applied to microarray images, the distortions in the decompressed images are very noticeable and likely to introduce too much variability in downstream stages.

In general, the first decision to make when choosing an image compression method is whether one absolutely needs lossless compression or can afford to have some loss of data in the reconstruction. If lossy data is acceptable, one must also decide the amount of acceptable loss. The measure used to quantify the loss must be tailored to the domain of application. For photographic images, the standard measure of quality is mean square error or peak signal to noise ratio (PSNR), although measures that better capture the perceptual loss have been proposed (for example, see Ref. [26]). In the microarray image domain, Jornsten *et al.* [27,28] allow a distortion in the reconstructed images that would not affect the extraction of the useful information in downstream stages [4]. According to the authors, an acceptable loss is a loss of data in the reconstructed microarray image that is smaller than the variability between different images obtained by scanning the same microarray multiple times [27].

Although the scientific literature on image compression is vast (a search on the collection of computer science bibliographies with the keywords 'image compression' returned almost 6000 results), the body of published knowledge on compression of microarray image is not very extensive. Nevertheless, some general principles are emerging. For example, it is clear that to achieve good compression, one needs to exploit the unique geometric structure of microarray images. Because most of the information is in the foreground, the accurate segmentation of the background is a crucial step. If the encoder does a poor job in segmenting the image, some of the foreground will be misclassified and the compression will suffer. Another distinctive feature of microarray images is the fact that they come in pairs and that the two images are strongly correlated. Because both images have the spots exactly in the same location, an encoder could exploit the data in one image to compress the other.

The compression of microarray images without loss of data turns out to be particularly challenging [29–32]. Several papers report that the eight least significant bits of each pixel of microarray images statistically resemble random data (for example, see Ref. [30]), which, if true, would imply that they would be impossible to compress. The consequence of this is a theoretical upper bound of 2:1 of lossless compression, which in fact is rarely achieved in practice. Zhang *et al.* [33] were the first to report a slight reduction in the size of the eight least significant bits of microarray images.

As mentioned, the first step in most approaches to microarray image compression is the separation of the foreground from the background. The rationale is that the data in the foreground have different statistical features than the data in the background and must be treated differently. In fact, in the majority of the papers reviewed, background and foreground are compressed separately [29–31]. It is therefore necessary for the encoder to have the capability to determine which pixels belong to the foreground and which ones belong to the background (the segmentation problem).

Microarray images are segmented using different strategies, the most adaptive of which are capable of handling rotations and distortions of the microarray grid. There is now strong evidence that the segmentation method can significantly affect the accuracy of the data extracted from microarray images [34]. According to Yang *et al.* [4], existing methods can be grouped in four categories: fixed circle segmentation; adaptive circle segmentation; adaptive shape segmentation; and histogram segmentation. Fixed

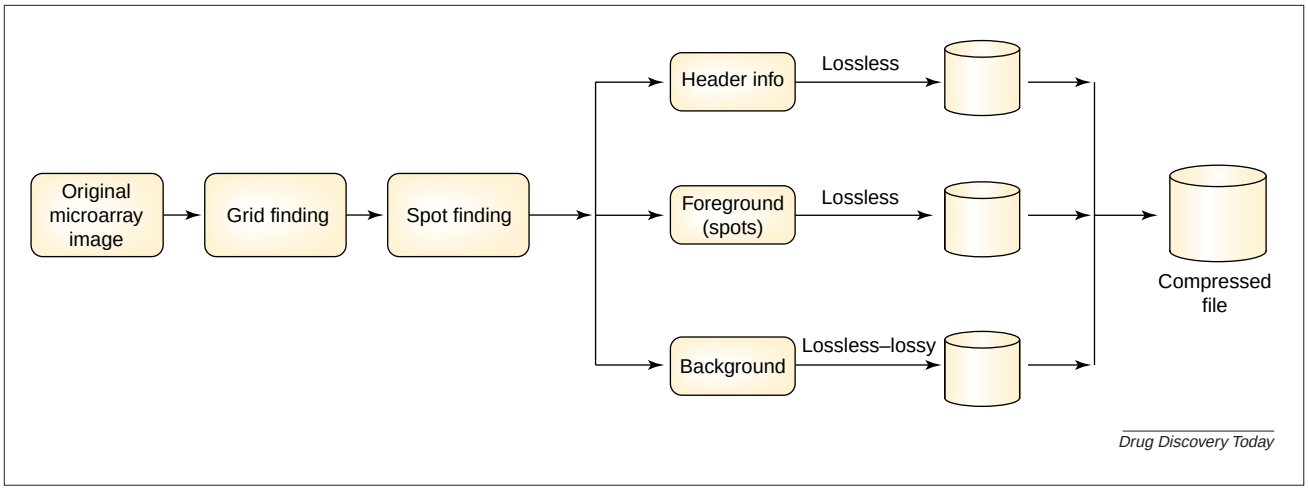

FIGURE 2

An outline of the typical workflow in lossy and lossless methods for microarray image compression. In the grid-finding step, the geometry of the image is recovered. In the spot-finding step, the spots are located and encoded. The content of image is divided in three channels, namely the foreground, the background and the header. The foreground stores the spots, the background the rest of the image and the header contains information about the geometry of the image. Each channel can be compressed lossy or lossless depending on the application.

circle segmentation assumes that the spots have a circular shape and fits a circle with fixed radius to all spots. Adaptive circle segmentation allows each spot to have its own radius. In adaptive shape segmentation, the shape of the spots is arbitrary, for example, Yang *et al.* [4] show that the foreground and the background can be grown from two initial seeds to become arbitrary shapes. Histogram segmentation methods are purely intensity-based. Pixels are classified as foreground or background using thresholds on the histogram of pixel values. For example, we have used an iterative refinement algorithm based on the horizontal and vertical histograms of the microarray images [31]. Some recent techniques classify the pixels based on statistical tests [29,33,34]; the tests are based on the assumption that foreground and background have different probability distributions.

When the image is segmented, the compressor must encode the foreground and the background. In some works, spots are encoded by first transforming them into 1D time series by tracing spiral paths [31,32]. Hua *et al.* [29,35] showed that an integer wavelet transforms for arbitrarily shaped objects can be used to encode both the foreground and the background. Zhang *et al.* [33] have compressed the data in the foreground and background with a technique called prediction by partial approximate matching (PPAM), which extends the idea proposed in the prediction by partial matching (PPM) text compression algorithm [36]. In PPAM, the encoder has an adaptive predictor that is capable of predicting the intensity of a pixel based on the brightness of neighboring pixels.

Comparing experimental results across different approaches is difficult because each paper uses different microarray images. However, when custom-designed lossless encoders are compared to JPEG-LS and JPEG 2000 lossless standard, the latter are still very competitive [29,31]. For example, our method [31] is on average just about 0.75 bpp better than JPEG-LS. When both the red and the green image are compressed jointly, the results are much more promising; Zhang *et al.* [33] report 6.3 bpp for compressing both images.

A significant improvement in compression can be achieved if one is willing to lose some data in the reconstruction (for example, see Refs [27,28,30,31,37]). Again, the typical strategy is to encode separately foreground and background, compress the spots without loss and compress the background lossy and more aggressively (see Figure 2 for an outline of the typical compression pipeline).

We [31] have used a combination of two powerful encoders, namely the Burrows–Wheeler transform [38] for the foreground and the set partitioning in hierarchical trees method [39] for the background, to achieve 4–6 bpp with extremely high quality (PSNR of 60 dB or better). Jornsten *et al.* [27,28,30] proposed keeping higher precision in low intensity spots and coarse image reconstruction near high intensity spots. More specifically, they compressed the foreground with a variant of the low complexity lossless compression (LOCO) algorithm, which is the JPEG-LS lossless and near-lossless standard [40]. Faramarzpour *et al.* [37] proposed to use a circle-to-square transform to map the foreground data to a spatial domain where a regular transform (i.e. discrete cosine transform) can be directly applied. Several authors have attempted to separate a greater amount of background data, typically based on the bit-planes of the image [30,31]. Jornsten *et al.* [30] have compressed the background using a variant of vector quantization, a method frequently used to compress real-world images. In several papers (for example, see Ref. [30]) it is reported that compression rates of 4 bpp (lossy compression of 4:1) minimally affects the information extracted from the images. In the realm of lossy–lossless hybrid schemes, these custom techniques for microarray

images have a significant advantage over, say, the JPEG standards. In fact, it is clear that none of the general-purpose compression methods can achieve these compression ratios without introducing significant distortions. An orthogonal approach to the compression and transmission of microarray images is the use of specialized hardware, as proposed by Samavi *et al.* [41].

Conclusions

Our brief review of microarray databases indicates that currently the large majority of them rely on the TIFF format for the long-term storage of raw microarray data. Frequently, JPEG (or GIF) is available for viewing or exporting the images. While research in microarray image compression is still in its infancy, it is reasonable to assume that it will take several years before a new standard dedicated to these images will emerge. A standard is necessary because it allows interoperability among different software tools, platforms and users, and guarantees some longevity to the data. Still, the administrators of large repositories of microarray images are struggling to keep their datasets within reasonable dimensions.

While research in microarray image compression is ongoing, it would be certainly beneficial if developers of microarray databases would include in their supported formats other high-performance lossless compression standards, like JPEG-LS or JPEG 2000. According to the literature, JPEG-LS and JPEG 2000 are likely to deliver compression ratios between 1.5:1 and 2:1 on microarray images. Currently, none of the microarray databases surveyed in this review uses the JPEG-LS and JPEG 2000 formats. If TIFF is the only lossless format allowed, we advise practitioners to consider saving their images in TIFF-LZW (i.e. TIFF compressed internally with Lempel–Ziv–Welch) to save storage space.

In the domain of lossy compression (or lossy–lossless hybrid schemes) the choice of the best standard is not very clear. General-purpose compression methods based on transforms (JPEG, JPEG 2000, etc.) introduce significant distortions in the regions around the spots, especially at low bit rates. These artifacts could significantly affect the extraction of information from the spot, and it is therefore risky to use these methods as a permanent storage format. The reality is that there is no commercial-quality lossy encoder that is capable of compressing microarray images to the bit rates and the quality described in the research papers surveyed in this review.

From the research viewpoint, we believe that the key to significant improvements in compression of microarray images is held in the invention of novel segmentation algorithms to separate background and foreground more effectively. Several automatic techniques have been proposed to handle microarray images in the presence of distortions in the shape of the spots, rotations of the grid, missing spots, deviations in the spot positions and so on. Most of the methods are Bayesian image processing techniques called Markov random fields (for reviews see [42–54]). Markov random fields and their variants have been shown to be very promising in capturing the unique geometry of microarray but they have not been used yet in microarray image compression.

We want to conclude on a practical note. This review of microarray image compression has revealed the urgent need of a standard set of microarray images to benchmark microarray image compression algorithms, like the Calgary corpus for text compression algorithms. A benchmark dataset would be of great help for comparing the quality of different algorithms (i.e. their compression efficiency and their computational cost) and for helping researchers in designing and fine-tuning new methods.

References

1 Schena, M. *et al.* (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467–470
2 DeRisi, J.L. *et al.* (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–686
3 Ball, C.A. *et al.* (2005) The Stanford Microarray Database accommodates additional microarray platforms and data formats. *Nucleic Acids Res.* 33 (Database issue), D580–582
4 Yang, Y.H. *et al.* (2002) Comparison of methods for image analysis on cDNA microarray data. *J. Comput. Graph. Statist.* 11, 108–136
5 Kuklin, A. (2000) Laboratory automation in microarray image processing. *Am. Lab.* 32, 64–67
6 Jain, A.N. *et al.* (2002) Fully automatic quantification of microarray image data. *Genome Res.* 12, 325–332
7 Miles, M.F. (2001) Microarrays: lost in a storm of data? *Nat. Rev. Neurosci.* 2, 441–443
8 Becker, K.G. (2001) The Sharing of cDNA Microarray Data. *Nat. Rev. Neurosci.* 2, 438–440
9 Geschwind, D.H. (2001) Sharing Gene Expression Data: an array of options. *Nat. Rev. Neurosci.* 2, 435–438
10 Brazma, A. *et al.* (2001) Minimum information about a microarray experiment (MIAME) – toward standards for microarray data. *Nat. Genet.* 29, 365–371
11 Brazma, A. *et al.* (2002) Microarray Data Representation, Annotation and Storage. *Adv Biochem Eng Biotechnol* 77, 113–139.
12 Stoeckert, J.S., Jr *et al.* (2002) Microarray Databases: Standards and Ontologies. *Nature genetics supplement*, Vol 32 pp. 469–473.
13 Goncalves, J. and Marks, L.W. (2002) Roles and Requirements for a Research Microarray Database. *IEEE Eng. Med. Biol. Mag.* 21, 154–157
14 Gardiner-Garden, M. and Littlejohn, G.T. (2001) A Comparison of Microarray Databases. *Brief. Bioinform.* 2, 143–158
15 Penkett, J.C. *et al.* (2004) Getting the Most from Public Microarray Data. *European Pharmaceutical Review* 1, 8–17
16 Anderle, P. *et al.* (2003) Gene Expression Databases and Data Mining. *Biotechniques* 34 (Suppl.), S36–S44
17 Gollub, J. *et al.* (2003) The Stanford Microarray Database: data access and quality assessment tools. *Nucleic Acids Res.* 31, 94–96
18 Brazma, A. *et al.* (2003) ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* 31, 68–71
19 Sarkans, U. *et al.* (2005) The ArrayExpress Gene Expression Database: a Software Engineering and Implementation Perspective. *Bioinformatics* 21, 1495–1501
20 Parkinson, H. *et al.* (2005) ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* 33, D553–D555
21 Manduchi, E. *et al.* (2001) RAD (RNA Abundance Database): an infrastructure for array data analysis. *Proceedings of SPIE* 4266, 68–78
22 Manduchi, E. *et al.* (2004) RAD and the RAD Study-Annotator: an approach to collection, organization, and exchange of all relevant information for high throughput gene expression studies. *Bioinformatics* 20, 452–459
23 Edgar, R. *et al.* (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.* 30, 207–210
24 Barrett, T. *et al.* (2005) NCBI GEO: mining millions of expression profiles—database and tools. *Nucleic Acids Res.* 33, D562–D566

25 Kooperberg, C. *et al.* (2002) Improved Background Correction for Spotted DNA Microarrays. *J. Comput. Biol.* 9, 55–66

26 Wang, Z. *et al.* (2004) Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 600–612

27 Jornsten, R. *et al.* (2002) Microarray image compression: SLOCO and the effects of information loss, *Signal Processing Journal (Special Issue on Genomic Signal Processing)*.

28 Jornsten, R. and Yu, B. (2000) Comprestimation: Microarray images in abundance. In *Proceedings of Conference on Information Science and Systems,* Princeton.

29 Hua, J. *et al.* (2002) Fast Segmentation and Lossy-To-Lossless Compression of DNA Microarray Images. In *Proceedings of Workshop on Genomic Signal Processing and Statistics* (GENSIPS), Raleigh, NC, USA

30 Jornsten, R. *et al.* (2002) On the bitplane compression of microarray images. In *Statistical data analysis based on the L_1 norm and related methods*. Springer-Verlag

31 Lonardi, S. and Luo, Y. (2004) Gridding and Compression of Microarray Images. 2004 *IEEE Computational Systems Bioinformatics Conference*, pp.122–130, Stanford, CA, USA

32 Faramarzpour, N. *et al.* (2003) Lossless DNA Microarray Image Compression. *IEEE Asilomar Conference on Signals, Systems and Computers*, pp.1501–1504 Vol.2, Pacific Grove, CA, USA.

33 *Zhang*, Y. *et al.* (2005) Lossless Compression of DNA Microarray Images. *Poster abstract of IEEE Computational Systems Bioinformatics Conference*, pp.128–132, Stanford, CA, USA

34 Ahmed, A.A. *et al.* (2004) Microarray segmentation methods significantly influence data precision. *Nucleic Acids Res.* 32, e50

35 Hua, J. *et al.* (2003) Microarray BASICA: Background adjustment, segmentation, image compression and analysis of Microarray Images. In *Proceedings IEEE International Conference on Image Processing,* pp. I585-588 (vol.1) Barcelona, Spain

36 Cleary, J. and Teahan, W. (1997) Unbounded length contexts for PPM. *The Computer Journal* 40, 67–75

37 Faramarzpour, N. *et al.* (2004) Lossy Compression of DNA Microarray Images. *Proceedings of CCECE / CCGEI 2004,* Ontario, Canada

38 Burrows, M. and David, J. Wheeler. (1994) A block-sorting lossless data compression algorithm. Technical Report, Hewlett Packard Laboratories, Number SRC-RR-124

39 Said, A. and Pearlman, W.A. (1996) A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology* 6, 243–250

40 Weinberger, S.M. and Seroussi, G. (2000) The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. Image Process.* 9, 1309–1324

41 Samavi, S. *et al.* (2003) DNA Microarray Image Compression by Pipeline Architecture. *IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 2176–2179 Vol.2, Pacific Grove, CA, USA

42 Carstensen, J.M. (1996) An active lattice model in a Bayesian framework. *Comput. Vis. Image Underst.* 63, 380–387

43 Katzer, M. *et al.* (2003) A Markov random field model of microarray gridding. In *Proceedings of the ACM Symposium on Applied Computing,* pp.72-77, Melbourne, FL

44 Ho, J. *et al.* Gridding Spot Centers of Smoothly Distorted Microarray Images. *IEEE Trans. Image Process.* (in press)

45 Li, Q. *et al.* (2005) Donuts, scratches and blanks: robust model-based segmentation of microarray images. *Bioinformatics* 21, 2875–2882

46 Wang, X.H. *et al.* (2003) Application of wavelet modulus maxima in microarray spots recognition. *IEEE Trans Nanobioscience* 2, 190–192

47 Katzer, M. *et al.* (2003) Methods for automatic microarray image segmentation. *IEEE Trans Nanobioscience* 2, 202–214

48 Demirkaya, O. *et al.* (2005) Segmentation of Microarray cDNA Spots Using Markov Random Field Modeling. *Bioinformatics* 21, 2994–3000

49 Bergemann, T.L. (2004) A statistically driven approach for image segmentation and signal extraction in cDNA microarrays. *J. Comput. Biol.* 11, 695–713

50 Brandle, N. *et al.* (2003) Robust DNA microarray image analysis. *Mach. Vis. Appl.* 15, 11–28

51 Glasbey, C.A. and Ghazal, P. (2003) Combinatorial image analysis of DNA microarray features. *Bioinformatics* 19, 194–203

52 Liew, A.W.C. *et al.* (2003) Robust adaptive spot segmentation of DNA microarray images. *Pattern Recognit.* 36, 1251–1254

53 Steinfath, M. *et al.* (2001) Automated image analysis for array hybridization experiments. *Bioinformatics* 17, 634–641

54 Galinsky, V.L. (2003) Automatic registration of microarray images. I. Rectangular grid. *Bioinformatics* 19, 1824–1831