An Overview of DNA Microarray Grid Alignment and Foreground Separation Approaches

Peter Bajcsy

The National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, IL 61801, USA

This paper overviews DNA microarray grid alignment and foreground separation approaches. Microarray grid alignment and foreground separation are the basic processing steps of DNA microarray images that affect the quality of gene expression information, and hence impact our confidence in any data-derived biological conclusions. Thus, understanding microarray data processing steps becomes critical for performing optimal microarray data analysis. In the past, the grid alignment and foreground separation steps have not been covered extensively in the survey literature. We present several classifications of existing algorithms, and describe the fundamental principles of these algorithms. Challenges related to automation and reliability of processed image data are outlined at the end of this overview paper.

1. INTRODUCTION

The discovery of microarray technology in 1995 has opened new avenues for investigating gene expressions [1] and introduced new information problems [2, 3]. Researchers have developed several microarray data processing methods and modeling techniques that are specific to DNA microarray analysis [4] and with the objective to draw biologically meaningful conclusions [5–8]. However, the analysis of DNA microarray data consists of several processing steps [9] that can significantly deteriorate the quality of gene expression information, and hence lower our confidence in any derived research result. Thus, understanding microarray data processing steps [10] becomes critical for performing optimal microarray data analysis and deriving biologically meaningful conclusions. We present a simple workflow of microarray data processing steps in Figure 1 to motivate our overview.

The workflow in Figure 1 starts with raw image data acquired with laser scanners and ends with the results of data mining that have to be interpreted by biologists. The microarray data processing workflow includes issues related to (1) data management (e.g., MIAME compliant database [11]), (2) image processing (grid alignment, foreground separation, spot quality assessment, data quantification and normalization [12, 13]), (3) data analysis (identification of differentially expressed genes [14], data mining [15, 16], integration with other knowledge sources [17, 18], and quality and repeatability assessments of results [19]), and (4) biological interpretation (visualization [20]). The objective of this paper is to overview only DNA microarray grid alignment and foreground separation approaches. These two particular microarray processing steps have not been covered extensively in the past (see [5, 6, 12, 15]). In addition, the full coverage of all microarray data processing issues in sufficient details would not be permissible in a survey journal paper due to a page limit. The reader is referred to books for less recent but broader coverage of microarray processing steps [12].

Before presenting DNA microarray grid alignment and foreground separation approaches, we introduce the term "ideal" DNA microarray image in terms of its image content. The image content would be characterized by constant grid geometry, known background intensity with zero uncertainty, infinite spatial resolution, predefined spot shape (morphology), and constant spot intensity that (a) is different from the background, (b) is directly proportional to the biological phenomenon (up- or down-regulation), and (c) has zero uncertainty for all spots. For multichannel microarray images, the same characteristics of an ideal image apply to each image channel and the channels are perfectly aligned. One microarray image can also contain multiple subgrids. Figure 2 shows an example of such an ideal microarray image. While finding such an ideal cDNA image is probably a pure utopia, it is a good starting point for understanding image variations and possibly simulating them [21]. One can view the overview of multiple grid alignment and foreground separation approaches as a set of techniques that try to compensate for deviations from the "ideal" microarray image model.
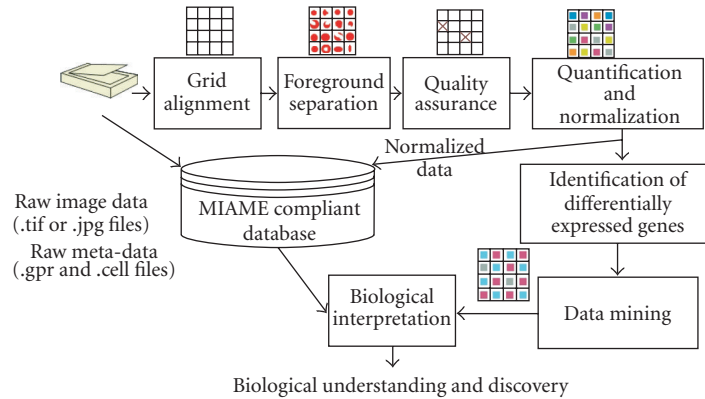
FIGURE 1: Microarray data processing workflow.

One could also mention that the grid alignment and foreground separation steps in cDNA processing do not occur in processing of oligonucleotide arrays, such the Affymetrix GeneChip (http://www.affymetrix.com). Oligonucleotide arrays contain only foreground and therefore the extracted descriptors represent absolute gene expression level. From an image processing viewpoint, the Affymetrix chips are easier to process since there is no background and the spot shape is rectangular. However, cDNA arrays are appropriate for detecting long DNA sequences while oligonucleotide arrays are designed for detecting only a short DNA sequence. Furthermore, the Affymetrix technology has been much more expensive than the technology with coated glass slides.

We present an overview of grid alignment techniques in Section 2, foreground separation methods in Section 3, and conclude our paper in Section 4. First, grid alignment methods are overviewed in terms of (1) automation as manual, semiautomated and fully automated, and (2) their underlying image analysis approaches as template-based and data-driven. Data-driven grid alignment algorithms are decomposed into (a) finding grid lines, (b) processing multiple channels, (c) estimating grid rotation, and (d) finding multiple grids. Next, foreground separation methods are described as those using (1) spatial templates, (2) intensity-based clustering, (3) intensity-based segmentation, and (4) spatial and intensity information.

2. GRID ALIGNMENT METHODS

A grid alignment (also known as addressing or spot finding [22] or gridding [23]) is one of the processing steps in microarray image analysis that registers a set of unevenly spaced, parallel, and perpendicular lines (a template) with the image content representing a two-dimensional (2D) array of spots [24]. The registration objective of the grid alignment step is to find all template descriptors. The template descriptors include line end point coordinates, so that pairs of perpendicular lines intersect at the center locations of a 2D array of spots in a microarray scan. Furthermore, this step has to identify any number of distinct grids of spots in one image.
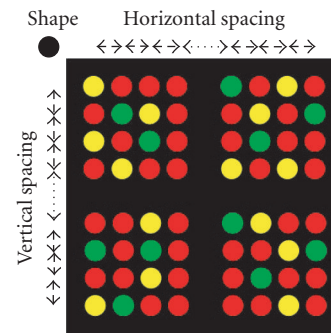
(a)
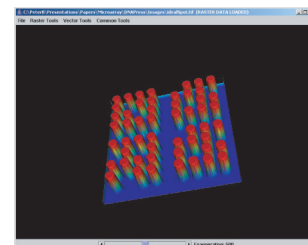
(b)
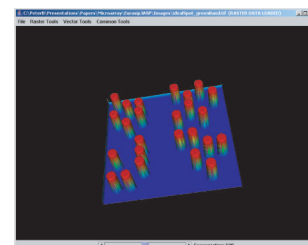
(c)

FIGURE 2: 2D illustration of an "ideal" microarray image (a) with constant shape, horizontal and vertical spacing parameters, and intensity profile. 3D visualization of the red (b) and green (c) channels. Both channels are characterized by the same parameters and are perfectly aligned.

There are two views on microarray grid alignment. First, grid alignment methods could be viewed in terms of automation as manual, semiautomated, and fully automated [15, Chapter 3], [25], [12, Chapter 6]. Second, grid alignment techniques could be viewed in terms of their underlying image analysis approaches as template-based and data-driven [24].

2.1. Automation level of grid alignment methods

Manual grid alignment methods

Given the fact that one expects a spot geometry to be very similar to a grid (or a set of subgrids), a manual alignment method is based on a grid template of spots. A user specifies dimensions of a grid template and a radius of each spot to form a template. Computer user interfaces like a computer mouse are available for adjusting the predefined grid template to match the microarray spot layout.

To compensate for many microarray image variations, one could possibly obtain "perfect" grid alignment assuming that human-computer interface (HCI) software tools are built for adjusting shape and location of each spot individually. It is apparent that this approach for grid alignment is not only very time consuming and tedious, but also almost impossible to repeat or use for high-throughput microarray image analysis.

Semiautomated grid alignment methods

In general, there are some parts of grid alignment that can be reliably executed by computers, but other parts are dependent on user's input. One example would be a manual grid initialization (selection of corner spots, specification of grid dimensions), followed by automated search for grid lines and grid spots [23]. The automated component can be executed by using either a grid template that is matched to the image content with image correlation techniques, or a data-driven technique that assumes intensity homogeneous background and heterogeneous foreground. The benefits of semiautomated grid alignment methods include reductions of human labor and time, and an increase of processing repeatability. Nevertheless, these methods might not suffice to meet the requirements of high-throughput microarray image processing.

Fully automated grid alignment methods

These methods should reliably identify all spots without any human intervention based on one-time human setup. The one-time setup is for incorporating any prior knowledge about an image microarray layout into the grid alignment algorithms in order to reduce their parameter search space. Many times, the challenge of designing fully automated grid methods is to identify all parameters that represent prior knowledge and quantify constraints for those parameters. Typically, these methods are data-driven and have to optimize internally multiple algorithmic parameters in their parameter search space to compensate for all previously described microarray image variations.

While it is everyone's ultimate goal to design fully automated grid alignment methods, one has to understand that these methods depend entirely on data content. For example, if there is a missing line of spots (spot color is indistinguishable from background), then an algorithm would not be able to find any supporting evidence for a grid line. One approach to this problem is the assignment of algorithmic confidence scores to each found grid. Grids with low confidence can be set aside for further human inspection whereas the grids with high algorithmic confidence can be processed without any human intervention. Another approach is to build into a microarray image some fiduciary spots that could guide image processing and provide a self-correction mechanism.

Finally, the question arises how much accuracy one can gain by automating alignment, and under what image variability conditions. This is an open research topic that requires a user study to quantify the accuracy, computational requirements, and consistency of alignment results. The user studies should also include the links between automation and the variations in cDNA microarray images.

2.2. Image analysis approaches to grid alignment

2.2.1. Template-based approaches

The template-based approach is the most prevalent in the previous literature and existing software packages, for example, GenePix Pro by Axon Instruments [26], ScanAlyze [27], or GridOnArray by Scanalytics [28]. Most of the currently available software packages enable manual template matching [26] (GenePix), [27] (ScanAlyze), [29] (Dapple), [30] (ImageGene). The procedure for manual template-based matching can be described as follows. Define the template by specifying number of subgrids, number of spots along rows and columns of a microarray image, spot diameter and spot spacing along rows and columns. Then adjust template location and all the above parameters to match the spots in a microarray image of interest. The quality of a match is assessed visually by maximizing the inclusion of spot pixels inside of one of the spots forming a template.

Some software products already incorporate an automatic template location adjustment (also called refinement) by searching for the best match between a fixed template and the microarray data [26] (GenePix), [31] (QuantArray), [32] (Array Vision). The refinement is executed by maximizing correlation of (1) an image template formed based on user's inputs and (2) the processed microarray image over a set of possible template placements (e.g., translated and rotated from the user defined initial position). If the parameters of the template become one part of the refinement search, then the approach is referred to as refinement with deformable templates. For example, it is possible to employ refinement with deformable templates based on Bayesian grid matching [33] to achieve certain data-driven flexibility into grid alignment.

The template-based approach is viewed as appropriate if the measured grid geometry does not deviate too much from the expected grid model as defined by a template [28].

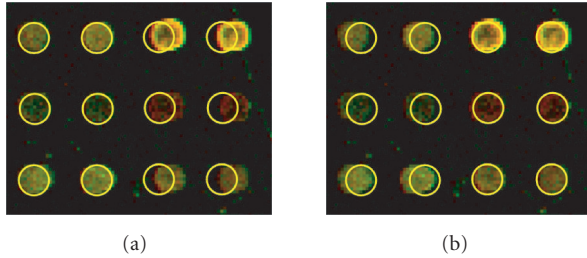(a)   (b)

FIGURE 3: Template-based alignment results obtained by visually aligning the left two columns (a) or the right two columns (b) of microarray spots.

If measured spot grids are unpredictably irregular, then this approach leads to (a) inaccurate results or (b) unacceptable costs for creating grid templates that would be custom-tuned to each batch of observed grid geometries. An example of alignment inaccuracies is shown in Figure 3. In this figure, the middle columns of spots have different spacing than the left two and the right two columns. A single template with a fixed spacing between columns leads to alignment errors illustrated in Figure 3. To increase accuracy of alignment, one would have to introduce multiple templates at the cost of larger number of parameters to adjust.

2.2.2. Data-driven approaches

There are several components of data-driven algorithms and each component solves one part of the grid alignment puzzle. We overview basic components of data-driven grid alignment algorithms that involve (1) finding grid lines, (2) processing multiple channels, (3) estimating grid rotation, and (4) finding multiple grids. We also present the algorithmic issues related to (1) tradeoffs between speed and accuracy, (2) repeatability and parameter optimization, and (3) incorporating prior knowledge about grids.

Finding grid lines

The first "core" component that finds grid lines is (a) based on statistical analysis of 1D image projections [34–37], or (b) used as part of image segmentation algorithms [38–40]. The algorithmic approach based on 1D image projections consists of the following steps [24, 37]. First, a summation of all intensities over a set of adjacent lines (rows or columns) is computed and denoted as a projection vector. Second, local extremes (maxima for bright foreground or minima for dark foreground) are detected within the projection vectors. These local extremes represent an approximation of spot centers. The tacit assumption is that the sought lines intersect a large number of high-contrast and low-contrast areas in contrary to the background that is assumed to be intensity homogeneous with some superimposed additive noise. Third, a set of lines is determined from the local extremes by incorporating input parameters (e.g., number of lines) and by finding consistency in spacing of local extremes. Fourth, all

intersections of perpendicular lines are calculated to estimate spot locations. The input microarray intensities can be preprocessed to remove dark-bright schema dependency (e.g., by edge detection [24]), or to enhance contrast of spots (e.g., by matched filtering or spot amplification [22]). Figure 4 illustrates 1D projections derived from a preprocessed image by Sobel edge detection algorithm [41]. After preprocessing the input image (Figure 4(a)), projection vectors are formed by summing adjacent rows (Figure 4(b)) or columns (Figure 4(c)). The graphs in Figures 4(a) and 4(b) show the dependency of the projection vector on the row or column location. The minima in these graphs refer to the locations with the smallest intensity change (in between spots) while the maxima refer to the locations with the maximum intensity change (across spots).

The other algorithmic approaches to finding grid lines that are based on image segmentation use primarily morphological processing [40, 42, 43] or Markov random field (MRF) models [38, 39, 44, 45] or graph models [25, 46]. In [40], adaptive thresholding and morphological processing steps are used to detect guide spots. The guide spots are defined as the locations of good quality spots (circular in shape, of appropriate size and intensity consistently higher than the background), for instance, the spots in Figure 5. With the help of guide spots and given the information about microarray layout, the final grid can be estimated automatically. The drawback of this approach is the assumptions about the existence of guide spots and the absence of spurious "spots" due to contamination. Another MRF segmentation-based approach reported in [38, 39] uses region growing segmentation to obtain partial grids that are then evaluated by grid hypothesis testing. The grid alignment problem is formulated as an MRF labeling problem, where subgrids are defined as sites, and placement hypotheses for the subgrids are labels. Finally, the graph-based grid alignment represents spots as ε-graphs, with "up," "down," "left," and "right" edges [46]. A block of spots is formed when neighboring vertices of edges are identified with ε-similar edge lengths.

Processing multiple channels

Given multichannel microarray images, the second component of data-driven methods tackles usually the problem of fusing image channels (also called bands). During multichannel microarray data acquisition, each channel is acquired at a different time and hence spatial misalignment of image channels might occur. Thus, this aspect of channel fusion requires cross-channel alignment (registration) that is usually approached by standard registration techniques.

Next, due to the different image content of each channel (bright and dark spots, as well as background variations per channel), grid alignment is dependent on an image channel. The fusion problem has to bring together either input channels for grid alignment or the results of grid alignment obtained for each channel separately. This aspect of channel fusion can be approached by performing a Boolean operation on all channels [24] or by linear combination of all
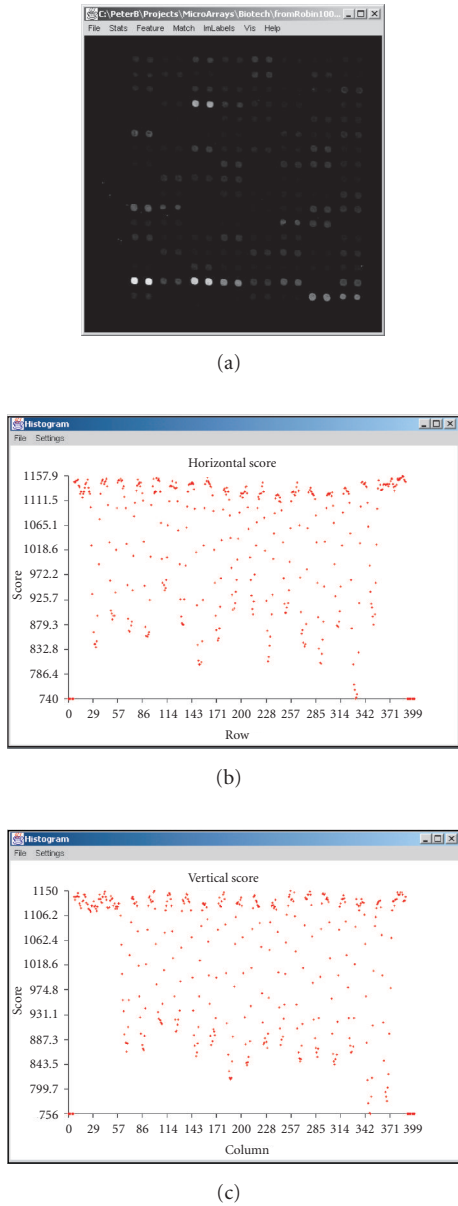
(a)

(b)

(c)

FIGURE 4: A microarray image (a) and its 1D projection scores (modified summations) derived from the original image after pre-processing by Sobel edge detection. 1D projections along rows (b) and along columns (c).

FIGURE 5: An example of guide spots as used in [40] for grid alignment.

(a)

(b)

(c)

FIGURE 6: Illustration of processing multiple channels. Microarray images of red (a) and green (b) channels that are fused by Boolean OR function before further processing (c).

channels weighted by the median values [23]. For instance, multiple channels could be fused by performing (channel1 OR channel2 OR channel3 OR . . .) at a pixel level, as illustrated in Figure 6 for two channels. The fusion of all channels with a Boolean OR operator will propagate foreground and background intensity variations into the grid alignment algorithm and increase its robustness assuming that there is little spurious variation in the background. The option of fusing channels beforehand reduces multichannel computation and avoids the problem of merging multiple grids detected per each channel.
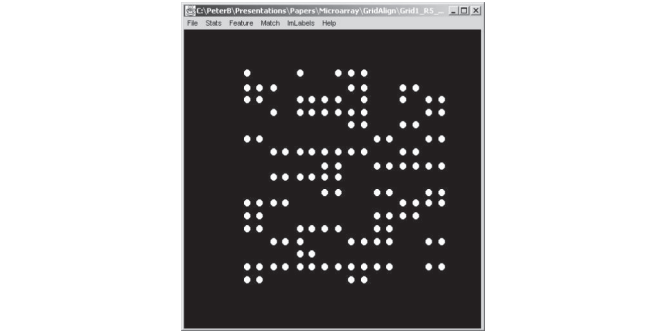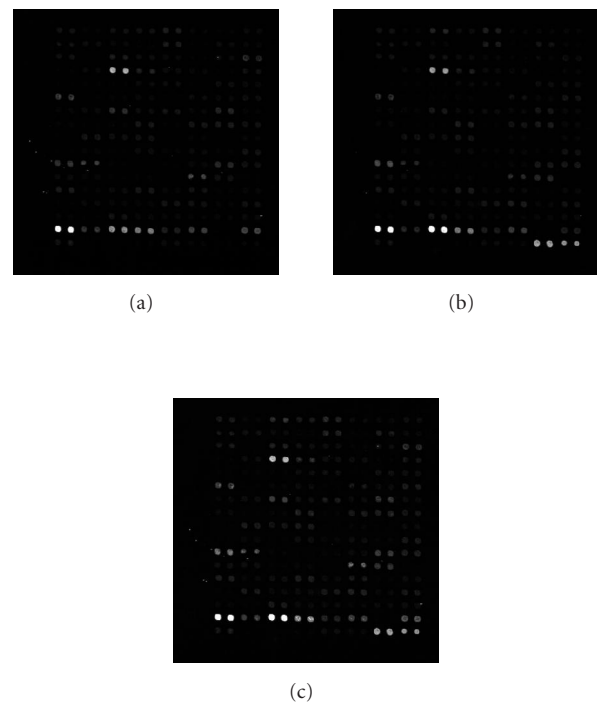
Estimating grid rotation

The third component of data-driven methods addresses the problem of grid rotation. This problem occurs due to the fact that the coordinate system of the robot printing the array may be slightly rotated with respect to the microarray image coordinate system [39]. One approach to this problem is an exhaustive search of all expected rotational angles [24]. This approach is motivated by the fact that the range of grid rotations is quite small, and therefore the search space is small. An initial angular estimate can be made by analyzing
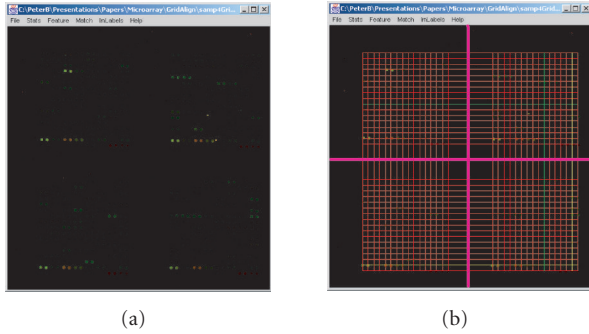
FIGURE 7: An example result of processing the original image (a) with the proposed algorithm and analyzing discontinuities in line spacing (b) to partition the original image into subimages containing one subarray per subimage.

four edges of a 2D array [37]. The disadvantage of this approach is that small angle image rotations introduce pixel distortions because rotated pixels with new noninteger locations are rounded to the closest integer location (row and column). Another approach to the grid rotation problem is the use of discrete Radon transformation [22]. In this case, the grid rotation angle is estimated by (a) performing projections in multiple directions (Radon transformation) and (b) selecting the maximum median projection value. While Radon transformation is computationally expensive, a significant speedup can be achieved by successive refinement of angular increments and limiting the range of angular rotations.

Finding multiple subgrids

Many times DNA microarray images contain multiple distinct 2D subarrays of spots (subgrids). The subgrids are separated by background and the subgrid edge-to-edge distance is larger than the intra-spot distances within each subgrid. The number of expected distinct subgrids can be defined by the number of subgrids along horizontal (row) and vertical (column) axes since distinct subgrids are also arranged in a 2D array format. The numbers of subgrids can be specified as input parameters since they are considered as part of our prior knowledge about microarray slides. Given the input parameters, the task is still to find image subareas that contain individual subgrids and then localize all spots in the subgrids. Due to the regular arrangement of printed subgrids and the approximate alignment of sub-grid edges with the image borders, one approach is to partition the original images into rectangular subareas based on the input parameters and then process each subarea separately.

If the input parameters are not available, then the problem can be approached by treating the entire image as one grid, searching for all irregular lines in the entire image, and then analyzing the spacing of all found mutually perpendicular grid lines [24]. Every large discontinuity in the line spacing will indicate the end of one and beginning of another sub-grid (2D arrays of spots). An example result is shown in Figure 7.

Speed and accuracy tradeoffs

Another optional component of data-driven methods could incorporate the speed and accuracy tradeoffs by image downsampling option. It is well known that the speed of most image-processing algorithms is linearly proportional to the number of pixels since every pixel has to be accessed at least once and processed in some way. To illustrate the processing requirements, let us consider two microarray images (image1 and image2) of the same pixel size and with the same content (intensity statistics per spot). Image1 and image2 contain N × M spots of radii R1 and R2, respectively, such that R1 < R2. The grid alignment processing of image2 could be performed faster without any loss of accuracy with respect to the alignment processing performed on image1 if image2 is subsampled by a factor of R1/R2. From this follows that the tradeoff between (a) speed (correlated with computational requirements) and (b) grid alignment accuracy is also a function of spot size (or radius R). In practice, downsampling (or local averaging) is preferred instead of subsampling in order to preserve local spot information that could be completely eliminated by subsampling.

Repeatability and parameter optimization

In order to introduce fully automated methods and hence microarray image processing repeatability, it is necessary to address the issue of algorithmic parameter optimization. The first part of this task is to discriminate one-time setup parameters, for example, number of grids or number of lines, from the data-dependent parameters, for example, size of spatial filters or noise thresholds. Next, it is beneficial to limit the ranges of parameters to be optimized by specifying their lower and upper bounds, for example, grid angular rotation. This step reduces any unnecessary computation cost during optimization. Finally, an optimization strategy has to be devised so that a global optimum rather than a local parameter optimum is found for a given "optimality" metric.

While the benefit of parameter optimization is a fully automated grid alignment tool, the drawback of optimization is the need for more computation and hence slower execution speed. From a system performance viewpoint, it is desirable to create optional user-driven inputs for algorithmic parameters in order to incorporate any prior knowledge about microarray image layout. Users that do not specify any microarray layout information will use more computational resources than users that partly describe input data. Nonetheless, the availability of optional algorithmic inputs and embedded parameter optimization techniques lets end users decide between the two application extremes, such as real-time performance with limited computational resources and offline processing with supercomputing resources.

Incorporating prior knowledge about grids

The most common prior knowledge about microarray layout includes number of grids (along rows and along columns), number of lines per grid, and perhaps spot radius. Other

inputs about corner spot locations, line spacing, grid rotation, or background characteristics should be easily incorporated into grid alignment algorithms. It is also possible that an irregularly spaced grid as detected by a data-driven method should be overruled by a strict regularity requirement on the final grid. For example, due to our prior knowledge about printing, the requirement to generate a grid with equally spaced rows could be incorporated into the final grid by (a) computing a histogram of distances between adjacent already detected rows, and (b) selecting the most frequent distance as the most likely correct row spacing [24]. One can then choose the row with the highest algorithmic confidence (score) as the initial location and place the final grid according to the regularity constraint.

The data-driven approaches are capable of finding irregular grids but are prone to misalignment due to spurious or missing spots. They are also dependent on many parameters. One can achieve significant cost savings with data-driven approaches when the majority of microarray slides meets certain quality standards and a fully automated algorithm flags images that are beyond its reliable processing capability.

3. FOREGROUND SEPARATION METHODS

The outcome of grid alignment is an approximation of spot locations. A spot location is usually defined as a rectangular image area enclosing one spot (also denoted as a grid cell). The next task is to identify pixels that belong to foreground (signal) of expected spot shape and to background. We refer to this task as foreground separation and it involves image segmentation and clustering.

The term image segmentation is associated with the problem of partitioning an image into spatially contiguous regions with similar properties (e.g., color or texture), while the term image clustering refers to the problem of partitioning an image into sets of pixels with similar properties (e.g., intensity, color, or texture) but not necessarily connected. The objective of segmentation inside of a grid cell is to find one segment that contains the foreground information. If a spot could be formed by a set of noncontiguous regions/pixels, then image clustering can be applied. While microarray image segmentation and clustering problems result in grouping pixels based on intensity similarities, it is quite frequent to use a spatial template-based separation, where the template follows a spot shape model. We should also mention foreground separation methods that assign foreground and background labels to pixels based on both intensities and locations.

We describe next the foreground separation methods using (1) spatial templates, (2) intensity-based clustering, (3) intensity-based segmentation, and (4) spatial and intensity information. We also address the issue of foreground separation from multichannel microarray images.

3.1. Foreground separation using spatial templates

This type of signal separation assumes that a spot is centered inside of a grid cell and it closely matches the expected spot morphology. The spatial template consists typically of two concentric circles, where the pixels inside of the smaller circle are labeled as foreground (signal) and the pixels outside of the larger circle are labeled as background (see Figure 8). All pixels in between of the two concentric circles are viewed as transition pixels and are not used. Clearly, this type of foreground separation will fail for spots with varying radii or spatial offsets from the grid cell center, and will include all pixels with artifacts (e.g., dust particles, scratches, or spot contaminants). The consequence of poor signal separation will lead to artificially increased background level and distorted signal-to-background ratio. A quantitative comparison of the results obtained from circular spots and segmented spots can be found in [36].

3.2. Foreground separation using intensity-based clustering

This type of signal separation boils down to a two-class image clustering problem (or image thresholding) [37]. Image thresholding is executed by choosing a threshold intensity value and assigning the signal label to all pixels that are above the threshold value (or below depending on a microarray image dark-bright scheme). The threshold value can be chosen by computing the expected percentage of spot pixels inside of a grid cell based on the knowledge about image resolution and spot radius. The thresholding approach can be viewed as clustering by determining a cluster separation boundary. Other clustering approaches use cluster intensity representatives, for instance, K-means or K-medoids [16], and the similarity between any intensity and the particular representative in order to assign pixel label (cluster membership). These methods can also be applied to the foreground separation problem [47].

Figure 9 shows examples of accurate and inaccurate foreground separation. In this example, we used an advanced K-means clustering algorithm [34] that iteratively reassigns foreground and background pixel labels until the cluster's centroid intensities do not change significantly.

3.3. Foreground separation using intensity-based segmentation

There are many segmentation methods available in the image processing literature [41, Chapter 6]. We will describe only those that have been frequently used with microarray images, such as seeded region growing, watershed segmentation, and active contour models.

Seeded region growing segmentation starts with a set of input pixel locations (seeds) [23, 35]. The segmentation method groups simultaneously pixels of similar intensities with the seeds to form a set of contiguous pixels (regions). The grouping is executed incrementally for a decreasing similarity threshold. The segmentation is completed when all pixels have been assigned to one of the regions grown from the initial seeds. In the case of microarray images, the foreground seed could be chosen either as the center location of a grid cell or as the maximum intensity pixel inside a grid cell.
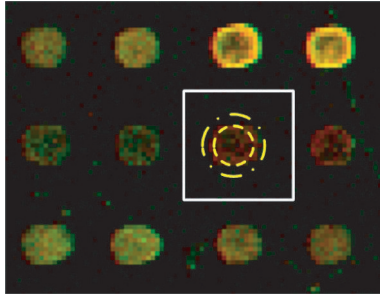
FIGURE 8: Illustration of a grid cell and the separation using spatial concentric circular templates.
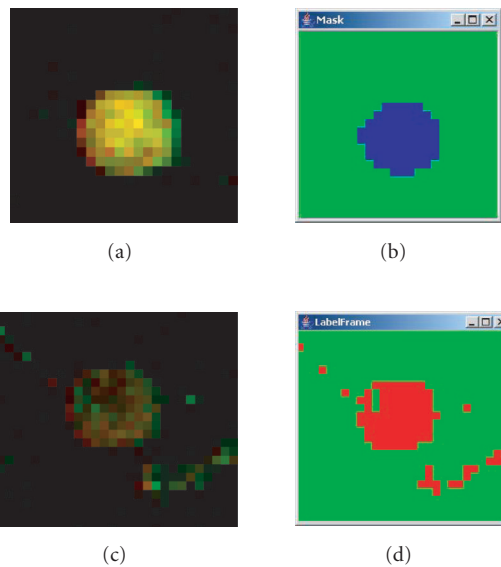

FIGURE 9: Examples of accurate ((a) original image, and (b) label image) and inaccurate ((c) original image, (d) label image) foreground separation using intensity-based clustering. The results were obtained using the Isodata (advanced K-means) algorithm [34].
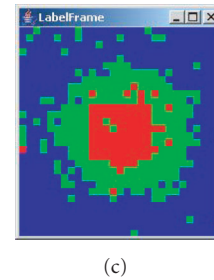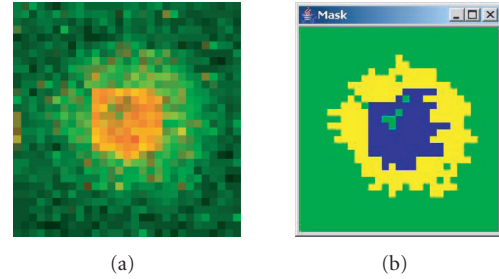

FIGURE 10: An example of pros and cons of foreground separation using intensity-based clustering and segmentation: (a) original image; (b) segmentation result; and (c) clustering result. The results were obtained using the Isodata (advanced K-means) [48] and region growing algorithms [34].

Similarly, the background seed could be selected either as the middle point between two spots or as the minimum intensity pixel inside a grid cell.

Morphological segmentation by watershed transformation is based on image operators derived from mathematical morphology [42]. There are two basic operators, dilation and erosion, and two composite operators, opening and closing. These operators are frequently used for filtering light or dark image structures according to a predefined size and shape. In the case of microarray images, morphological operators can filter out structures that deviate too much from the expected shape and size of a spot. Segmentation by watershed transformation can be viewed as the analysis of a grid cell intensity relief consisting of (a) no peak (missing spot), (b) one peak (clear spot), and (c) multiple peaks (vague spot). The case of multiple peaks is treated by searching for peak separation boundaries with the morphological operators that

mimic watersheds (flooding image areas below peaks). The outcome of the segmentation step is the region that corresponds to the most likely spots according to the morphological analysis of grid cell image intensities.

Active contour [39] and multiple snake [13] models start with an initial contour model and by deforming it the objective is to minimize some predefined energy functional. The initial contour is usually represented by a polygon in a digital domain. The energy functional is composed of several global and local constraints on the contour deformation (e.g., individual, group, and constraint energy as in [13], or spring chain constraints as in [39]). Some preprocessing is usually necessary to address the problems with touching spots, large spot size variation, and convergence of greedy algorithms to local minima.

The main difference between foreground separation using clustering and using segmentation is illustrated in Figure 10. If a spot segment (region) is correctly identified, then the segmentation approach will exclude dark pixels from the foreground assuming that they are surrounded by a connected set of pixels. In contrary, the clustering approach will include to the foreground cluster pixels that belong to the background or the intensity transitioning area. These pros and cons can be seen in the middle and right images in Figure 10.

Another issue to consider while choosing the most appropriate foreground separation technique is the priority order for selecting correct foreground pixels. There are certain grid cells where multiple interpretations are plausible as illustrated in Figure 11. If two segments of approximately the same size are detected inside of a grid cell (see Figure 11),

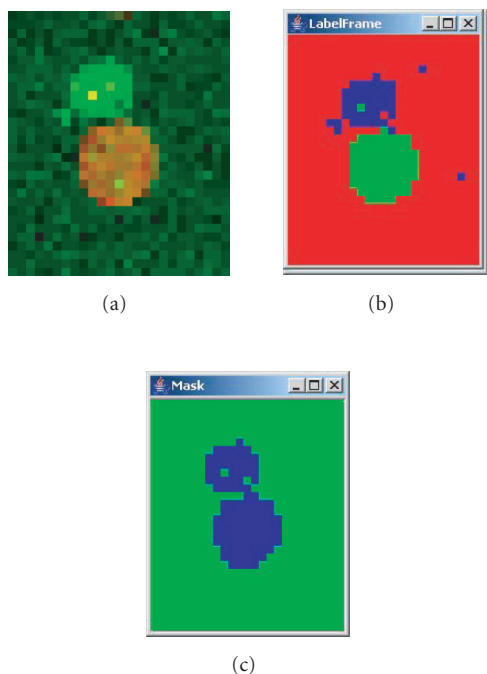(a)   (b)

(c)

FIGURE 11: Multiple interpretations of the original grid cell image (a). The interpretation can vary based on prior region intensity and/or location and/or morphology information: (b) two distinct foreground clusters characterized by similar intensities; (c) one foreground contiguous region.

then should we select (a) the brighter segment, (b) the segment with less irregular shape, or (c) the segment closer to the grid center? If a scratched spot consisting of two half disks is considered as a valid spot, then should we include into foreground all segments of the same intensity that are close or connected to the main segment positioned over the grid center? These decisions require ordering priorities in terms of expected region intensity, location, and spot morphology.

3.4. Foreground separation using spatial and intensity information (hybrid methods)

Several foreground separation methods try to integrate the prior knowledge about spot morphology (spatial template), spot location, and expected intensity distribution. These methods could be viewed as a sequence of steps consisting of segmentation or clustering image partitions, spatial template image partitions, statistical testing, and foreground/background trimming.

Spatially constrained segmentation and clustering

For instance, foreground separation using segmentation leads to a connected region that is fitted to a spatial template [40]. If the best-fitted circle deviates too much from the template, then the spot is labeled as invalid. It is also possible to apply repeatedly clustering and mask matching [49] by which intensity and shape features are integrated. Another example

would be foreground separation using clustering with additional minimization constraint on cluster dispersion [47]. The particular choice of clustering could be the partitioning method based on K = 2 medoids (PAM) with Manhattan distance as the similarity metric. This method in [47] was reported to be robust to the presence of noise in microarray images.

Mann-Whitney statistical testing

This foreground separation algorithm is executed by randomly selecting N pixels from the background and N pixels with the lowest intensities from the foreground over an expected spatial template of a spot [50]. Next, the two sets of pixels are compared according to the Mann-Whitney test [51, Test 12] with critical values of 0.05 or 0.01. The Mann-Whitney nonparametric test is a technique designed for evaluating a hypothesis whether or not two independent samples represent two populations with different median values. Iteratively, the darkest foreground pixels are replaced with those pixels that have not yet been chosen, and evaluated until the Mann-Whitney test satisfies the statistical significance criteria. The foreground separation is then achieved by selecting all pixels with higher intensities than the background pixels that passed the statistical significance test. It is apparent that this method relies on good selections of background pixels but incorporates our prior knowledge about spot template and expected intensity distributions. Unfortunately, this method cannot detect the presence of artifacts that bias the foreground separation results.

The results of this statistical method are dependent on the number of available samples within a grid cell and hence image resolution. A high statistical confidence in microarray measurements would be obtained from a digital image with very high spatial resolution (very large number of pixels per grid cell). However, the cost of experiments, the limitations of laser scanners in terms of image resolution, storage limitations, and other specimen preparation issues are real-world constraints that have to be taken into account. Thus, the results of this method, as well as many other statistical methods [51], are always reported with the number of foreground and background pixels that defines the statistical confidence of the derived results.

Spatial and intensity trimming

This approach is based on analyzing intensity distributions of foreground and background pixels as defined by a spatial template and then discarding those pixels that are classified as distribution outliers [15, Chapter 3]. Spatial trimming is achieved by initial foreground and background assignments over a spot template while intensity trimming is accomplished by removing pixels with intensity outliers with respect to foreground and background intensity distributions. The goal of spatial and intensity trimming is to remove (a) contamination pixels (e.g., dust or dirt) in foreground and background regions, and (b) artifact pixels (e.g., doughnut spot shape) in foreground region. Figure 12 illustrates
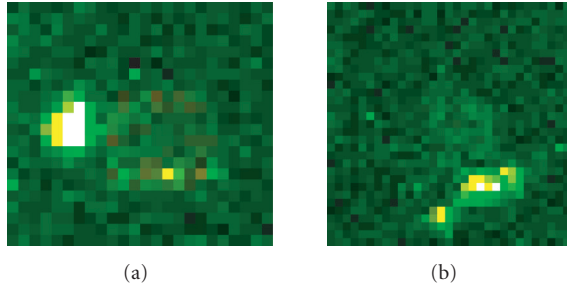
FIGURE 12: A couple of grid cell examples where contamination pixels (the very bright pixels) have to be trimmed.
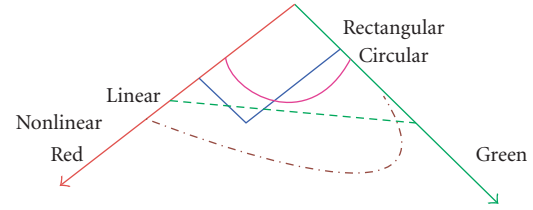

FIGURE 13: Possible foreground separation boundaries for two-channel input data. The two perpendicular axes denote intensities in red and green channels. All other curves illustrate shapes of boundaries that separate foreground and background (e.g., for dark background, the points between a boundary and the intersection of red and green axes are labeled as "background" and all other points as "foreground.")

a couple of examples where contamination pixels would skew the resulting gene expressions if they would not be trimmed off.

The trimming approach is similar to Mann-Whitney statistical testing but the statistical testing of the trimming method is applied to foreground and background pixels (intensity distribution analysis) instead of only to background pixels in the case of Mann-Whitney statistical testing. The spatial trimming can be improved by using two concentric circles that define foreground, background, and transient pixels. The transient pixels are eliminated from the analysis since they are not reliable. During intensity trimming, the choice of intensity threshold values that divide distribution outliers from other intensities depends on a user and the values are related to a statistical confidence. Empirically, a good performance is obtained when the threshold values eliminate approximately 5–10% of each, foreground and background, cumulative distributions [15, Chapter 3]. However, this approach should not be used when a spot size is very small (3-4 pixels in diameter) since the underlying statistical assumption of this analysis is the use of a sufficiently large number of samples (pixels). For example, for a spot of the radius equal to two pixels, there would be only $\pi*2^2 = 12.57$ foreground pixels, and the number of foreground outliers would be $5\%*\pi*2^2 = 0.63$ pixel.

3.5. *Foreground separation from multichannel microarray images*

For the case of multichannel images, the choices of foreground separation approaches have to be explored [52]. The goal in this case is to assign a label "foreground" or "background" to each pixel based on a vector of intensities. For example, the red and green input image channels from a cDNA slide form a two-dimensional vector of intensities at each pixel. Foreground separation can be achieved by processing red and green intensities separately or together.

Let us consider the foreground separation using intensity thresholding. The foreground separation threshold values can be computed by considering (1) Euclidean distances to each pixel represented as a two-dimensional intensity vector (circular separation), (2) intensities for red and green channels at each pixel separately (rectangular separation), (3) correlated intensities for red and green channels (linear

separation), or (4) intensities of pixels after fusing red and green channels with some nonlinear operators (nonlinear separation, e.g., fusing channels with the Boolean OR operator). Depending on the choice of thresholding approach, the foreground separation boundary for a two-channel microarray image will lead to circular, rectangular, linear, or nonlinear curves as illustrated in Figure 13.

Each of the aforementioned separation boundaries leads to a different set of spot and background labels. One should be aware of different statistical assumptions about a joint PDF of multiple channels associated with each separation boundary. A few examples of the results obtained using multiple boundary types are shown in Figure 14. As expected, the total count of foreground pixels varies based on the multichannel separation method; circular-15913, rectangular-509, linear-15877, nonlinear AND -13735, and nonlinear OR -16045 (400×400 image size, two bytes per pixel).

4. SUMMARY

Microarray technology and the data acquired from it form a new way of learning about gene expression using sophisticated visualization tools [53]. We have overviewed DNA microarray grid alignment and foreground separation approaches to summarize our current understanding about these two basic microarray processing steps. The importance of these two processing steps lies in the fact that they are the first operations performed with any raw microarray images. Challenges related to automation and reliability of processed image data remain still open questions.

For example, automation is important to guarantee microarray image processing repeatability. Assuming that an algorithm is executed with the same data, we expect to obtain the same results every time we perform an image processing step. In order to achieve this goal, algorithms should be "parameter-free" so that the same algorithm can be applied repeatedly without any bias with respect to a user's parameter selection. Thus, for instance, any manual positioning of a grid template is not only tedious and time-consuming but also undesirable since the grid alignment step cannot then be repeated easily. A concrete example of the repeatability

(a)   (b)

(c)   (d)
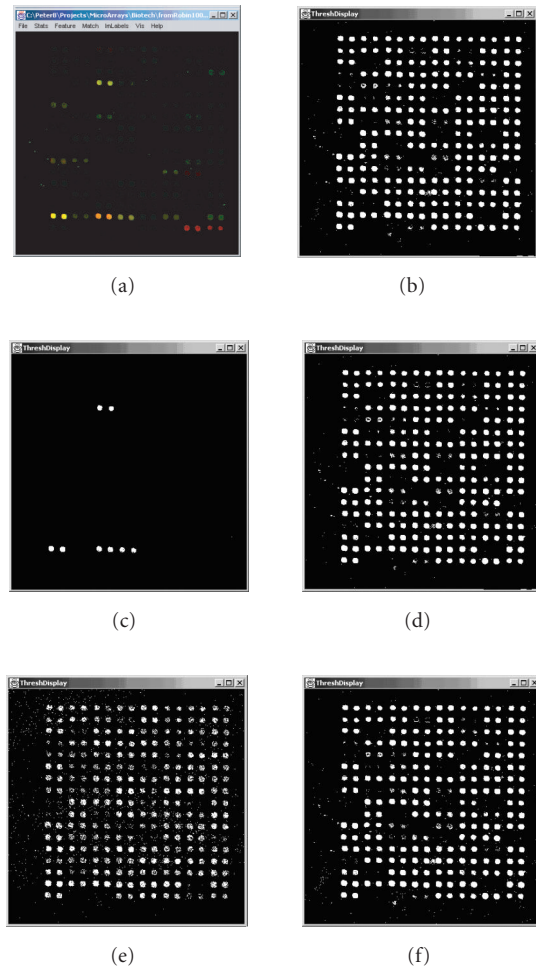
(e)   (f)

FIGURE 14: Examples of the results for spot versus background separation obtained from the two-channel input image shown in the top row (a) with multiple boundary types; circular (b), rectangular (c), linear (d), nonlinear after AND operation (e), and nonlinear after OR operation (f).

issues is presented in [54], where authors compared results obtained by two different users from the same slide (optic primordial dissected from E11.5 wild-type and aphakia mouse embryos) while using the ScanAlyze software package [27]. Each user provided the same input about grid layout first, and then placed multiple grids independently and refined the spot size and position. The outcome of the comparison led up to two-fold variations in the ratios arising from the grid placement differences.

Furthermore, the amount of microarray image data is growing exponentially and so one is concerned about preparing sufficient storage and computational resources to meet the requirements of end users. For example, finding a grid of spots can be achieved much faster from a subsampled microarray image (e.g., processing one out of 5×5 pixels), but the grid alignment accuracy would be less than if the original microarray image had been processed. There are clearly tradeoffs between computational resources (memory and speed/time) and alignment accuracy given a large number of

microarray images [24]. While this issue might be resolved without any accuracy loss by using either supercomputers or distributed parallel computing with grid-based technology [55, 56], it might still be beneficial to design image processing algorithms that could incorporate such resource limitations [55].

One could speculate about the future of cDNA microarray image processing in terms of automation, processing reliability, storage, and computational requirements as described above. It is possible to envision a parallel between the future microarray technology and the past semiconductor technology advancements. The semiconductor industry has gone through several decades of technological improvements with respect to wafer materials, data processing, and automation to achieve the current state. This could become a model for the microarray technology advancements. In addition, it could be foreseen that current standardization efforts [11] will enable more automation and higher reliability, introductions of new substrate materials will lead to higher quality of data [57] and the continuing algorithmic work with the supporting efforts to build a bioinformatics cyber-infrastructure [58] will lead to very high-throughput microarray image processing and eventually to much better understanding of biological phenomena.

ACKNOWLEDGMENTS

REFERENCES

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with complementary DNA microarray," *Science*, vol. 270, pp. 467–470, 1995.

[2] D. Fenstemacher, "Introduction to bioinformatics," *Journal of the American Society for Information Science and Technology*, vol. 65, no. 5, pp. 440–446, 2005.

[3] W. J. MacMullen and S. O. Denn, "Information problems in molecular biology and bioinformatics," *Journal of the American Society for Information Science and Technology*, vol. 65, no. 5, pp. 447–456, 2005.

[4] J. Quackenbush, "Computational analysis of microarray," *Computational Analysis of Microarray*, vol. 2, no. 6, pp. 418–427, 2001.

[5] P. Bajcsy, J. Han, L. Liu, and J. Young, "Survey of bioData analysis from data mining perspective," in *Data Mining in Bioinformatics*, J. T. L. Wang, M. J. Zaki, H. T. T. Toivonen, and D. Shasha, Eds., chapter 2, pp. 9–39, Springer, New York, NY, USA, 2004.

[6] P. Baldi and S. Brunak, *Bioinformatics, The Machine Learning Approach*, The MIT Press, Cambridge, Mass, USA, 2nd edition, 2001.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene

expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[8] S. K. Moore, "Understanding the human genome," *IEEE Spectrum*, vol. 37, no. 11, pp. 33–42, 2000.

[9] A. B. Goryachev, P. F. MacGregor, and A. M. Edwards, "Unfolding of microarray data," *Journal of Computational Biology*, vol. 8, no. 4, pp. 443–461, 2001.

[10] P. Bajcsy, "An overview of microarray image processing requirements," in *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), the Workshop on Computer Vision Methods for Bioinformatics (CVMB)*, San Diego, Calif, USA, June 2005.

[11] A. Brazma, P. Hungamp, J. Quackenbush, et al., "Minimum information about a microarray experiment (MIAME) - toward standards for microarray data," *Nature Genetics*, vol. 29, no. 4, pp. 365–371, 2001.

[12] G. Kamberova and S. Shah, Eds., *DNA Array Image Analysis - Nuts and Bolts. Data Analysis Tools for DNA Microarrays*, DNA Press LLC, Salem, Mass, USA, 2002.

[13] T. Srinark and C. Kambhamettu, "A microarray image analysis system based on multiple-snake," *Journal of Biological Systems*, vol. 12, no. 4, 2004, Special issue.

[14] H. Yue, P. S. Eastman, B. B. Wang, et al., "An evaluation of the performance of cDNA microarrays for detecting changes in global mRNA expression," *Nucleic Acids Research*, vol. 29, no. 8, p. e41-1, 2001.

[15] S. Draghici, *Data Analysis Tools for DNA Microarrays*, CRC Mathematical Biology and Medicine Series, Chapman & Hall, London, UK, 2003.

[16] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, Calif, USA, 2001.

[17] L. H. Saal, C. Troein, J. Vallon-Christersson, S. Gruvberger, Å. Borg, and C. Peterson, "BioArray software environment: a platform for comprehensive management and analysis of microarray data," *Genome Biology*, vol. 3, no. 8, 2002, software 0003.1-0003.6.

[18] H. Samartzidou, L. Turner, and T. Houts, "Lucidea Microarray ScoreCard: An integrated tool for validation of microarray gene expression experiments, Innovation Forum, Microarrays," Life Science News 8, 2001 Amersham Pharmacia Biotech.

[19] D. Rocke and B. Durbin, "A model for measurement error for gene expression arrays," *Journal of Computational Biology*, vol. 8, no. 6, pp. 557–569, 2001.

[20] J. Seo and B. Shneiderman, "Interactively exploring hierarchical clustering results," *IEEE Computer*, vol. 35, no. 7, pp. 80–86, 2002.

[21] Y. Balagurunathan, E. R. Dougherty, Y. Chen, M. L. Bittner, and J. M. Trent, "Simulation of cDNA microarrays via a parameterized random signal model," *Journal of Biomedical Optics*, vol. 7, no. 3, 2002.

[22] N. Brandle, H. Bischof, and H. Lapp, "Robust DNA Microarray image analysis," *Machine Vision and Applications*, vol. 15, no. 1, pp. 11–28, 2003.

[23] C. W. Whitfield, A. M. Cziko, and G. E. Robinson, "Gene expression profiles in the brain predict behavior in individual honey bees," *Science*, vol. 302, pp. 296–299, 2003.

[24] P. Bajcsy, "Gridline: automatic grid alignment in DNA microarray scans," *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 15–25, 2004.

[25] H.-Y. Jung and H.-G. Cho, "An automatic block and spot indexing with k-nearest neighbors graph for microarray image analysis," *Bioinformatics*, vol. 18, no. 2, pp. S141–S151, 2002.

[26] Axon Instruments Inc., "GenePix Pro, Product Description," http://www.axon.com/GN_Genomics.html.

[27] M. Eisen, "ScanAlyze," Product Description at http://rana.lbl.gov/EisenSoftware.htm.

[28] Scanalytics Inc., "MicroArray Suite," Product Description at http://www.scanalytics.com/product/hts/microarray.html.

[29] J. Buhler, T. Ideker, and D. Haynor, "Dapple: improved techniques for finding spots on DNA microarrays," Tech. Rep. UWTR 2000-08-05, UV CSE, Seattle, Wash, USA.

[30] Biodiscovery Inc., "ImaGene Product description," 2005, http://www.biodiscovery.com/imagene.asp.

[31] Packard BioChip Technologies, LLC, "Quant Array Analysis Software," Product Description at http://las.perkinelmer.com/Content/RelatedMaterials/ReflectionTechNote.pdf.

[32] Imaging Research Inc., "Array Vision," Product Description at http://www.imagingresearch.com/products/Genomics_Software.asp.

[33] K. Hartelius and J. M. Cartstensen, "Bayesian grid matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 162–173, 2003.

[34] P. Bajcsy, "Image To Knowledge (I2K)," Software Documentation at http://isda.ncsa.uiuc.edu/i2kmanual/.

[35] CSIRO Mathematical Informational Sciences, "SpotImage Analysis Software," Product Documentation at http://experimental.act.cmis.csiro.au/Spot/index.php.

[36] A. N. Jain, T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, and D. Pinkel, "Fully automated quantification of microarray image data," *Genome Research*, vol. 12, no. 2, pp. 325–332, 2002.

[37] M. Steinfath, W. Wruck, H. Seidel, H. Lehrach, U. Radelof, and J. O'Brien, "Automated image analysis for array hybridization experiments," *Bioinformatics*, vol. 17, no. 7, pp. 634–641, 2001.

[38] M. Katzer, F. Kummert, and G. Sagerer, "Robust automatic microarray image analysis," in *Proceedings of the International Conference on Bioinformatics: North-South Networking*, Bangkok, Thailand, 2002.

[39] M. Katzer, F. Kummert, and G. Sagerer, "Methods for automatic microarray image segmentation," *IEEE Transactions on Nanobioscience*, vol. 2, no. 4, pp. 202–212, 2003.

[40] A. W.-C. Liew, H. Yan, and M. Yang, "Robust adaptive spot segmentation of DNA microarray images," *Pattern Recognition*, vol. 36, no. 5, pp. 1251–1254, 2003.

[41] J. Russ, *The Image Processing Handbook*, CRC Press LLC, Boca Raton, Fla, USA, 3rd edition, 1999.

[42] J. Angulo and J. Serra, "Automatic analysis of DNA microarray images using mathematical morphology," *Bioinformatics*, vol. 19, no. 5, pp. 553–562, 2003.

[43] R. Hirata, J. Barrera, R. F. Hashimoto, and D. O. Dantas, "Microarray gridding by mathematical morphology," in *Proceedings of 14th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '01)*, pp. 112–119, Florianopolis, Brazil, October 2001.

[44] G. Antoniol and M. Ceccarelli, "A markov random field approach to microarray image gridding," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, Cambridge, UK, August 2004.

[45] O. Demirkaya, M. H. Asyali, and M. M. Shoukri, "Segmentation of cDNA microarray spots using Markov random field modeling," *Bioinformatics*, vol. 21, no. 13, pp. 2994–3000, 2005.

[46] H.-J. Jin, B.-K. Chun, and H.G. Cho, "Extended epsilon regular sequence for automated analysis of microarray images,"

in *The IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), the Workshop on Computer Vision Methods for Bioinformatics (CVMB)*, San Diego, Calif, USA, June 2005.

[47] D. Bozinov and J. Rahnenfuhrer, "Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering," *Bioinformatics*, vol. 18, no. 5, pp. 747–756, 2002.

[48] J. T. Tou and R. C. Gonzales, *Pattern Recognition Principles*, Addison-Wesley, Reading, Mass, USA, 1974.

[49] J. Rahnenführer and D. Bozinov, "Hybrid clustering for microarray image analysis combining intensity and shape features," *BMC Bioinformatics*, vol. 5, no. 1, p. 47, 2004.

[50] 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, no. 4, pp. 364–374, 1997.

[51] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall CRC, London, UK, 2nd edition, 2000.

[52] R. Lukac, K. N. Plataniotis, B. Smolka, and A. N. Venetsanopoulos, "An automated multichannel procedure for cDNA microarray image processing," *Lecture Notes in Computer Science*, vol. 3212, pp. 1–8, 2004.

[53] R. M. Adams, B. Stancampiano, M. McKenna, and D. Small, "Case study: a virtual environment for genomic data visualization," *IEEE Transactions on Visualization*, vol. 1, 2002, October 27-November 1, 2002, Boston, Mass, USA (published as CD).

[54] N. D. Lawrence, M. Milo, M. Niranjan, P. Rashbass, and S. Soullier, "Reducing the variability in cDNA microarray image processing by Bayesian inference," *Bioinformatics*, vol. 20, no. 4, pp. 518–526, 2004.

[55] I. Foster and C. Kesselman, "Computational grids," in *The Grid: Blueprint for a New Computing Infrastructure*, chapter 2, Morgan-Kaufman, San Francisco, Calif, USA, 1999.

[56] M. Karo, C. Dwan, J. Freeman, J. Weissman, M. Livny, and E. Retzel, "Applying grid technologies to bioinformatics," in *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPDC '01)*, pp. 441–442, San Francisco, Calif, USA, August 2001.

[57] C. M. Strom, D. D. Clark, F. M. Hantash, et al., "Direct visualization of cystic fibrosis transmembrane regulator mutations in the clinical laboratory setting," *Clinical Chemistry*, vol. 50, no. 5, pp. 836–845, 2004.

[58] Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, http://www.nsf.gov/od/oci/reports/toc.jsp.

Peter Bajcsy has earned his Ph.D. degree from the Electrical and Computer Engineering Department, University of Illinois at Urbana-Champaign, Ill, 1997, and M.S. degree from the Electrical Engineering Department, University of Pennsylvania, Philadelphia, Pa, 1994. He is currently with the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, Ill, working as a Research Scientist on problems related to automatic transfer of image content to knowledge and X-informatics, where X stands for bio, hydro, medical image, and sensor. In the past, he had worked on real-time machine vision problems for semiconductor industry and synthetic aperture radar (SAR) technology for government contracting industry. He has developed several software systems for automatic feature extraction, feature selection, segmentation, classification, tracking, and statistical modeling from multispectral and microscopy data sets. Dr. Bajcsy's scientific interests include informatics, image and signal processing, statistical data analysis, data mining, pattern recognition, novel sensor technology, and computer and machine vision.

Special Issue on

Cross-Layer Design for the Physical, MAC, and Link Layer in Wireless Systems

Call for Papers

Motivation

Cross-layer design is a common theme for emerging wireless networks, such as cellular systems and ad hoc networks, though the objectives and requirements can be very different. A cross-layer approach is particularly important when designing protocols at the physical (PHY) and medium access control (MAC)/link layer for several reasons. The past decade gave birth to innovative techniques such as opportunistic communication, rate adaptation, and cooperative coding, which all require close interaction between layers. For example, optimizing the rate-delay tradeoff in a multiuser wireless system requires knowledge of PHY parameters such as channel state information (CSI) as well as delay information from the MAC. As another example, the notion of link in wireless relay systems is generalized towards a collection of several physical channels related to more than two nodes. The potential for novel solutions through cross-layer design is further seen in the fact that it motivates collaboration among researchers of different specialties, such as antennas and propagation, signal processing, error-control coding, scheduling, and networking (MAC) protocols.

This special issue is devoted to research efforts that use cross-layer design to jointly optimize the lowest two layers of the wireless communication systems. Such cross-layer solutions will have decisive impact in future wireless networks.

Contributions on the following and related topics are solicited:

- Random access protocols with multipacket reception (MPR), multiantenna MPR protocols
- MAC/ARQ protocols for multiantenna and wireless relay systems
- MAC-level consideration of time-varying channels: channel estimation, channel state information (CSI), and link adaptation
- MAC-level interference mitigation via management of PHY-level multiuser networks: for example, interfering multipleaccess (MA) channels

- MAC/PHY protocols and analysis of cognitive radio networks
- MAC protocols for PHY-restricted devices/circuitry (e.g., RFID)
- PHY-aware scheduling and resource allocation
- Analysis of fundamental tradeoffs at the PHY/MAC interface: rate-delay, energy-delay
- PHY/MAC/link aspects of network coding in wireless networks
- PHY/MAC issues in wireless sensor networks and distributed source co
- Cross-layer optimization of current standards (WiMAX, W-CDMA, etc.)
- Interaction of MAC/PHY design with the circuitry and the battery management

Authors should follow the EURASIP Journal on Advances in Signal Processing manuscript format described at http://www.hindawi.com/journals/asp/. Prospective authors should submit an electronic copy of their complete manuscript through the EURASIP Journal on Advances in Signal Processing Manuscript Tracking System at http://mts.hindawi.com/, according to the following timetable:

Manuscript Due	February 1, 2008
First Round of Reviews	May 1, 2008
Publication Date	August 1, 2008

GUEST EDITORS:

Petar Popovski, Center for TeleInFrastructure, Aalborg University, Niels Jernes Vej 12, A5-209, 9220 Aalborg, Denmark; petarp@kom.aau.dk

Mary Ann Ingram, School of Electrical and Computer Engineering, Georgia Institute of Technology, 777 Atlantic Drive NW, Atlanta, GA 30332-0250, USA; maryann.ingram@ee.gatech.edu

Christian B. Peel, ArrayComm LLC, San Jose, CA 95131-1014, USA; chris.peel@ieee.org

Shinsuke Hara, Graduate School of Engineering, Osaka City University, Osaka-shi 558-8585, Japan; hara@info.eng.osaka-cu.ac.jp

Stavros Toumpis, Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus; toumpis@ucy.ac.cy

WoSPA 2008

An International Workshop on Signal Processing and its Applications

18 – 20 March 2008, University of Sharjah, Sharjah, U.A.E.

General Chair
B. Boashash
University of Sharjah, UAE
Univ. of Queensland, Australia

Chair, Organizing Committee
K. Abed-Meraim
University of Sharjah, UAE
Telecom-Paris, France

Co-Chairs, Technical Program
M. Bettayeb
University of Sharjah, UAE
M. Gabbouj
Tampere Univ. of Technology, Finland
M. Al-Mualla
Etisalat University College, UAE

Publications and Web
M. Saad
University of Sharjah, UAE

Publicity
I. Shahin
University of Sharjah, UAE

Sponsorship
M. Samarai
University of Sharjah, UAE

Local Arrangements
Q. Nasir & N. Brahimi
University of Sharjah, UAE

Finance & Registration
C. B. Yahya
University of Sharjah, UAE

Social Events
I. Kamel
University of Sharjah, UAE

Industry Liaison
A.-K. Hamid
University of Sharjah, UAE

International Liaisons
A. Beghdadi, Europe
Université Paris XIII, France

S. Hussain, Asia
UTM University, Malaysia

M. Cheriet, North America
Université du Québec, Canada

A. Benazza, Africa
Supcom, Tunisia

V. Chandran, Australia
Queensland Univ. of Technology, Australia

Call For Papers

The 5th WoSPA is part of a series of workshops dedicated to signal processing theory and applications. It started in 1990 as part of the International Symposium on Signal Processing and its Applications (ISSPA-90). WoSPA was first held as a separate workshop in 1993 in Brisbane, Australia. It was followed by WoSPA97, WoSPA2000, WoSPA2002, and in 2005 it was merged with CCSP (International Conference on Communication, Computer and Signal Processing), which was held in Kuala Lumpur, Malaysia.

WoSPA provides a forum for engineers and scientists engaged in research and development in different fields and applications, with the common interest of Signal Processing, to discuss common objectives and applications, and to interact with leading specialists in the field. WoSPA encourages collaboration between industry, academia and government institutions.

This international workshop is organized by the University of Sharjah, UAE, and will take place on its premises on 18–20 March 2008.

Workshop Format: The workshop program is comprised of both oral and poster presentations. As in previous WoSPAs, it is intended that the morning session be mostly devoted to keynote and invited speakers, whereas the afternoon is for poster presentations, subject to the review of final submissions.

This fifth WoSPA will emphasize *multidisciplinary research* works involving signal processing in different engineering and fundamental fields. For that, four special sessions will be organized on:

- ✓ Multi-disciplinary research in Signal Processing
- ✓ Signal Processing for Civil and Architectural Engineering
- ✓ Emerging fields in Signal Processing
- ✓ Signal Processing for Renewable Energies.

Papers are invited in, but not limited to, the following topics:

- **Methods:** Statistical Signal and Array Processing, Time-Frequency/Time-Scale Analysis, Non-linear System Theory, Image and Multi-dimensional Signal Processing (SP), Independent Component Analysis, Wavelets and Multi-rate SP, Fractals, Machine Learning and Pattern Recognition, Digital Filter Design, Algorithms and Implementation, Sensor Networks, Genomics SP, Software Computing, Cross-Layer Design, others.

- **Applications:** Biomedical Signal Processing, Audio and Video Signal Processing, Radar and Sonar, Machine Vision, Coding, Encryption and Watermarking, Biometrics, Object Recognition, Detection and Tracking, Wireless Communications, Cooperative Communications, Guidance & Control, Speech Recognition & Speaker Identification, Multimedia Signal Processing, others.

For more details see**: www.sharjah.ac.ae/wospa**

Important Deadlines:

Full Paper Submission:
December 15, 2007
Notification of Acceptance:
February 01, 2008
Final Accepted Paper Submission:
February 15, 2008

Conference Secretary

H. Yahyaoui
University of Sharjah, UAE
Tel : +971 6 505 0918
Fax :+971 6 505 0872
E-mail: hyahyaoui@sharjah.ac.ae

IEEE

EURASIP