Adaptive techniques for microarray image analysis with related quality assessment

Sebastiano Battiato
Gianpiero Di Blasi
Giovanni Maria Farinella
Giovanni Gallo
Giuseppe Claudio Guarnera
University of Catania
Image Processing Laboratory
Catania, Italy
E-mail: battiato@dmi.unict.it

Abstract. *We propose novel techniques for microarray image analysis. In particular, we describe an overall pipeline able to solve the most common problems of microarray image analysis. We propose the microarray image rotation algorithm (MIRA) and the statistical gridding pipeline (SGRIP) as two advanced modules devoted to restoring the original microarray grid orientation and to detecting, the correct geometrical information about each spot of input microarray, respectively. Both solutions work by making use of statistical observations, obtaining adaptive and reliable information about each spot property. They improve the performance of the microarray image segmentation pipeline (MISP) we recently developed. MIRA, MISP, and SGRIP modules have been developed as plug-ins for an advanced framework for microarray image analysis. A new quality measure able to effectively evaluate the adaptive segmentation with respect to the fixed (e.g., nonadaptive) circle segmentation of each spot is proposed. Experiments confirm the effectiveness of the proposed techniques in terms of visual and numerical data.* © 2007 SPIE and IS&T. [DOI: 10.1117/1.2816445]

1 Introduction

DNA microarray[1,2] is a fundamental biotechnology for gene expression profiling and biomedical studies. Image analysis has found applications in microarray technology because it is able to extrapolate new and nontrivial knowledge that is partially hidden in the images. In a typical microarray experiment, two 16-bit TIFF images are obtained using microarray scanners. The two images are conventionally assigned with a red and a green channel to elaborate them within conventional image processing software. The processing pipeline for these input data is summarizable in the following steps:[3] gridding, segmentation, intensity extraction, and quality measures.[4,5] Gridding and segmentation are crucial steps: they have a potentially large impact on subsequent analysis (e.g., clustering or identification of differentially expressed genes[6]). In the last decade, many academic[7] and commercial microarray image analysis software and methods have been developed. Some of them[5,8] improve a primordial simple solution[9,10] with heuristic strategies. Nevertheless, human interaction is still usually required to obtain a high level of accuracy. Microarray images contain a set of grids that are organized at two levels (e.g., 4×4, 16×16, etc.). As in Refs. 11 and 12, we focus our attention on inner grids, since several authors have successfully addressed the problem of locating and segmenting the outer grid.

In this paper we introduce a novel algorithm, the microarray image rotation algorithm (MIRA), to take into account typical rotation problems of microarray images. MIRA is aimed at correcting in a preprocessing phase, rotation problems in the microarray images, ensuring that successive pipeline steps are not affected by such geometrical distortions. MIRA uses statistical analysis on the rows and columns of a binary map to infer a correction angle. The microarray images obtained by MIRA are hence processed by the statistical gridding pipeline (SGRIP) and the microarray image segmentation pipeline (MISP[13]), to perform all the other involved steps in microarray image analysis: gridding, segmentation, and data/quality measure extraction.

In particular, the gridding process is realized by the SGRIP module: Histogram analysis on the rows and the columns of a binary mask of input data obtains statistical information about the signal's spatial distribution. Starting from an initial guess, the final grid mask is refined according to local considerations about typical spot acquisition problems (e.g., spot overlap, comet tails, etc.). SGRIP realizes a fully unsupervised gridding and leads to an accurate grid where each single spot is correctly addressed. MISP starting from the robust gridding provided by SGRIP reliably performs data extraction and quality measure evaluation. MIRA, MISP, and SGRIP together allow us to automatically detect the microarray grid, to correct the rotation angle, to assign coordinates to each spot, to discriminate among the foreground, background, and local background, to calculate intensity, and to extrapolate quality measures. To test our algorithms, we have developed a framework called microarray image analysis framework (MIAF). The

Paper 06056RR received Apr. 5, 2006; revised manuscript received Jun. 7, 2007; accepted for publication Jun. 23, 2007; published online Dec. 17, 2007. This paper is a extension of a paper presented at the SPIE Conference on Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, Jan. 2006, San Jose, Calif. The paper presented there appears (unrefereed) in SPIE Proceedings Vol. 6064.
1017-9909/2007/16(4)/043013/20/$25.00 © 2007 SPIE and IS&T.

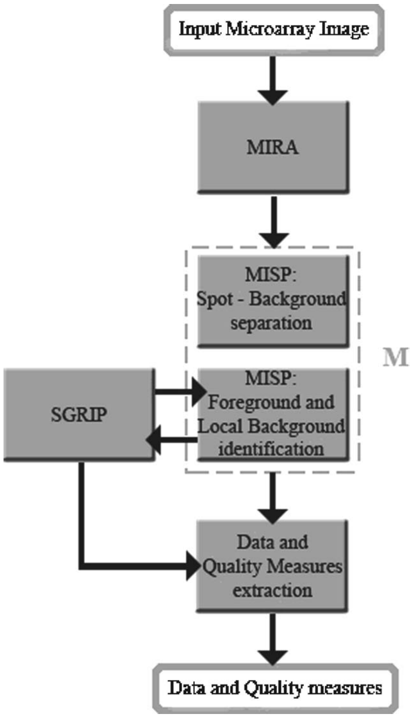


Fig. 1 The global architecture of the proposed MIAF framework.

overall architecture of the system is plug-in-based (Fig. 1). The pipeline has been validated through an extensive experimental phase.

The paper is organized as follows: Section 2 synthetically reviews microarray image analysis, focusing on the state of the art for rotation and gridding algorithms. Section 3 describes the preprocessing step performed by MIRA, and Section 4 summarizes the global MISP pipeline. SGRIP is described in detail in Section 5. A quality measure used for comparison of segmentation algorithms is presented in Section 6. Finally, experiments and conclusions are reported in Sections 7 and 8, respectively.

2 Related Works

A microarray typically consists of several blocks with the same layout, and the print-tips on the arrayer are normally arranged in a regular array.[14] Assuming ideal conditions, the spots in each block are located in an evenly spaced lattice corresponding to the layout of the print tips. The ideal microarray image has the following properties:[15]

1. all grids are of the same size;
2. subgrids are regularly spaced;
3. the spots are centered on the intersections of the subgrid lines;
4. the spots' size and shape are perfectly circular for all spots;
5. grid geometry does not change within a given type of slide;
6. no dust or contamination affects the slide;
7. there is minimal and uniform background intensity across the image.

In most of the real cases, the microarray images violate some of these conditions. Variation during the printing of the array produces a variation of the exact locations of spots in an unpredictable way. Other problems can be related to the spot position, irregularities on the spot (shape and size), and contamination.

In the following subsections, we briefly review the capabilities of existing microarray image analysis tools, specifying in detail the proposed solution to manage visualization, segmentation, and intensity and quality measure extraction.

Rotation and gridding processes are reported in separate subsections to better analyze and compare existing solutions with SGRIP and MISP capabilities. Also, we briefly compare the overall characteristics and major drawbacks.

2.1 *Microarray Image Analysis Tools*

Scanalyze[9] starts by converting each pair of input *TIFF* images into a single RGB image as follows:

$$R = \frac{Gain \times \left(\frac{Cy5\_pixel}{256}\right)}{\text{(Normalization)}},$$

$$G = Gain \times \left(\frac{Cy3\_pixel}{256}\right), \quad B = 0. \tag{1}$$

The brightness is controlled by the *Gain* parameter, while *Normalization* is a balancing parameter. The *Scanalyze* gridding phase relies on user intervention to specify various parameters (e.g., separation between rows and columns). *Scanalyze* uses the fixed-circle segmentation method. The circle size has to be user-provided and leads to the gridding phase. All spots are considered valid even if they do not enclose a signal; such spots can be manually excluded from the successive information extraction phase. To estimate the background local intensity, the median values are used. The ratio G/R of the average foreground is obtained by taking into account the correction of the background. The software produces a set of information and several quality values (e.g., channels correlation) for each spot.

Genepix[8] uses a square root transformation to reduce the dynamic range of input images. It is also possible to manually select the 8 bits to be saved, or to use the predefined options to save high, low, or central bits. Spot addressing is automatic. In the first versions, a circle-based segmentation with variable dimension was used, and the spot could be classified as "not found" according to some conditions (e.g., "spot diameter less than 6 pixels," "spot position overlaps another spot," or "spot diameter is outside the planned options limits"). In more recent releases, this segmentation method has been replaced by irregular spot segmentation. Local and global methods are used to compute the background and foreground intensities for each spot. Spot circularity is also adopted as a quality measure.

Spot[6] uses a linear combination weighted by the median values to obtain a single image, $I = G' + (mG/mR) \times R'$, where G' and R' are the images obtained from G and R using a square root transformation and mG and mR are the corresponding median values. Spot addressing is based on batch processing over a collection of microarray images with the same geometric structure. Successive steps are automatic and produce two estimated grids:

Table 1 Tool's characteristics.

		Segmentation		Gridding		
Software	Correction of Grid Rotation	Manual/Automatic	Type	Manual/Automatic	Parameters Required	Iterative/Single Step
Scanalyze[9]	No	Manual	Fixed circle, adaptive circle	Manual	Yes	Manual refinement
Genepix[8]	No	Automatic	Adaptive circle, seeded region growing	Automatic	Yes	Single step
Spot[6]	No	Automatic	Seeded region growing	Automatic	Yes, batch procedure	Iterative
Angulo and Serrà[7]	No	Automatic	Morphological operators and watershed transformation	Automatic	No	Single step
Matarray[17]	No	Automatic	Fixed circle	Manual	Yes	Iterative
MAGIC[10]	No	Automatic	Fixed/adaptive circle, seeded region growing	Automatic	Yes	Single step
MIAF	Yes	Automatic	*Ad hoc* technique (MISP)	Automatic	No	Single step

- fitted foreground grids: horizontal and vertical lines passing for the centers of the estimated spots;
- fitted background grids: horizontal and vertical lines passing through the gaps of the estimated centers between spots.

In the segmentation phase, a seeded region-growing algorithm is used.[16] The seeds are chosen according to the estimated grids. Background, foreground intensity, and quality measures are computed similarly to GenePix.

Angulo and Serrà[7] combined input images using a linear combination weighted by median. The overall pipeline combines addressing and gridding techniques, making use of morphological operators together with classical segmentation algorithms; the overall performances are evaluated in terms of segmentation accuracy without providing quality measures.

Matarray[17] uses a combination of intensity and spatial information for spot detection and signal segmentation. The anchor point and grid dimension are specified by the user. Starting from a first draft identification of the spot centers, the overall area is split in patches defining a circular mask for each patch used for spot segmentation. An iterative process is then achieved, calculating the signal intensity and local background to improve detection. The combined quality index described in Ref. 17 is used for quality assessment.

MicroArray Genome Imaging and Clustering Tool (MAGIC[10]*)* analyzes all types of gene expression data on all major operating systems. Visualization is performed by a weighted linear combination. Gridding requires the number of grids and the number of rows and columns for each grid; moreover, an indication of the coordinates of the top left spot, top right spot, and bottom row is required. Segmentation is performed by choosing one of three algorithms: fixed circle, adaptive circle, and seeded region growing. The fixed circle is centered in the grid square, with a user-specified radius. The adaptive circle algorithm analyzes the signal in each square of the grid to determine the most appropriate center and radius (within a user-specified range) for each circle. Finally, the adaptive circle's center is set to contain the largest number of "on" pixels. A seeded region-growing algorithm connects each pixel to a background or foreground region until all pixels have been properly labeled. A user-specified threshold and geometric considerations determine which pixels may be used to "seed" the regions. The user can choose to consider the background in computation of a green to red ratio signal. Each spot can be ignored using manual flag selection. MAGIC creates an "Expression file" containing foreground and background spots' intensity for each channel and channel ratio intensity.

Only a few tools are able to consider and properly manage the overall involved microarray image analysis steps. Table 1 shows the main characteristics of each reviewed tool, also reporting the presence or absence of fundamental modules such as rotation, quality measure extraction, etc. To better highlight differences and similarities with the proposed approach, we also report MIAF at the end of Table 1.

2.2 *Rotation*

Microarray image analysis can be affected by several errors when input grids are slightly rotated. Despite this, only a few authors have tried to address and solve this problem.

In Refs. 11 and 12, the two input images are filtered according to the orientation matching transform (OM), which is aimed at detecting the candidate points for the spot centers. To compute the angles α and β between the grid directions and the axis, the Radon transform (RT) of the images is filtered by the OM, whose peaks are analyzed. The directions of the projection by integrating the space variable s in the RT are used. The algorithm computes the two main peaks of the function below:

$$\Gamma(\phi) = \int_s \Re^2\{\mathrm{OM}\{I\}(s,\phi)dS\}. \tag{2}$$

It is important to notice that the OM transformation used in Refs. 11 and 12 requires the minimum and maximum spot radii as input parameters, which are usually hard to know in real case.

In our algorithm, we compute the peaks of a function $f(I)$, but dissimilarly from Refs. 11 and 12, the algorithm does not require parameters. In Ref. 19, the rotation detection is done after a preprocessing that estimates some global variables for gridding. Subarray rotation identification is achieved by the examination of the intensity projection profile along the x- and y-axes of a black-and-white binary image obtained from previous steps. A subarray is identified as a "rotated region" if the size of the block is greater than the average subarray's width and height. To detect if the rotation is clockwise or counterclockwise, the rotation directions are compared to the intensity sum of the top one-third region with the bottom one-third region along the horizontal and vertical axes in the rotated region. To calculate the rotation angle, the authors of Ref. 19 iteratively rotate the region by a quarter degree until its projection profile is matched to the normal one. The method proposed in Ref. 19 is affected by two main problems:

1. If a subgrid is affected by some acquisition problems such as wide areas with very high noise level or weak spot signals, its profile will be different from others also in absence of a rotation. In this case, the subgrid may be identified as rotated and, as shown in Ref. 19, the iterative refinement may enter in a loop, since at each iteration the subgrid profile will be different from others'.
2. If the problems indicated above are located in the small regions of the subgrid used for this purpose or those regions are empty (no spot signal), the results of the detection may be wrong.

2.3 *Gridding*

One of the main challenging problems in the context of microarray image analysis is the gridding step. The gridding process assigns the coordinates to each microarray spot. This phase may be carried out manually or automatically. Automatic addressing increases the speed of the analysis, but few of the common available software offer this option.

We decided to consider in a separate section the gridding process, because even if almost all tools provide some heuristic solution, a lot of advanced and reliable *ad hoc* solutions have to be mentioned in detail.

Some of the proposed methods require user intervention for setting the grid anchor points, grid dimension in terms of rows and columns, etc. A good gridding method must hence be fully automatic and fast, simple, and adaptive with respect to real microarray structure and fluctuations of the parameters.

In Ref. 14, the spots are located by finding a rectangle containing pixels of each spot and using it as a valid mask for gridding. Two main steps are performed: First, the algorithm sums up the intensities across the pixels in each row (column); next, it finds the local minima of the summed intensities using a sliding window whose span is approximately equal to the width of a typical spot. Although this method does not require human interaction, some parameters have to be known in advance: the number of spots in each row, the number of the spot column, and the size of the sliding window. In any case, the final gridding results do not take into account local spot irregularities or different spots' sizes and shapes.

Local information approximating spot size and shape with advanced segmentation strategies is indeed a crucial step to derive reliable gridding information: Our proposed solution tries to move from these considerations.

The gridding algorithm proposed in Ref. 18 is fully automatic. In the first stage, the algorithm is applied to the whole image in order to find the positions of the subgrid. Then it is used again on each subgrid to find the position of the spots. The first step computes the average intensities row by row and column by column on the whole image. To remove the noise, a low-pass filter is applied. Taking into account the regularity of the structure of the microarray image, it assumes that the distance between adjacent cells containing the spots should be approximately equal. The initial "guess" is obtained by finding the minima of the average intensity: An iterative refinement adjusts the initial guess. The initial guess may not form a regular grid, which is instead obtained after the final refinement, aimed at removing extra lines due to noise contamination and to adding missing lines due to low-intensity rows and columns. Spot borders are identified using adaptive circles. The method assumes that the axes of grids are parallel to the borders of the image and that no rotation of the grids has occurred during the digitization process. In this case, only regular grids without considering spot irregularity are obtained. Just to overcome such a problem, our approach is based on statistic observations of subgrid parameters that lead, in a first phase, to an orthogonal subgrid that is approximately regular. Extra or missing lines are removed or added by means of a single-step correction algorithm that uses the median distance between adjacent rows or columns as a reference parameters rather than using the average distance as in Ref. 18. A second phase allows us to obtain an adaptive subgrids, in which spot centers are correctly addressed, taking into account local variations and problems such as spot merging. We do not assume that subgrid axes are parallel to the border of the image, since MIRA performs a preprocessing phase to restore grid rotation. We identify the spot border using MISP, which performs an adaptive shape segmentation.

In Ref. 11, a gridding algorithm using the Radon transformation (RT) to compute the parameters of a regular grid is proposed. By analyzing the RT peaks, some parameters are properly estimated. In particular, the (x_0, y_0) coordinates of the upper left spot, the directional angles α, β, and the grid spacing Δx, Δy are successively used to determinate each grid point.

Such an algorithm has been tested on synthetic and real microarray images where the original real position on the grids is known in advance. In Ref. 12, such work has been extended with some new considerations. The first step, based on the radon transformation, is aimed at generating a grid hypothesis, while the second step accounts for local grid deformations. To refine the grid hypothesis, a Bayesian approach is used, to maximize the posterior probability (MAP estimate). The observed datum is the input image, a raw visual representation of an ideal grid with a well-defined organization. The MAP grid estimate of the most likely grid gives the unobserved image. A further refinement is achieved by means of an iterative meta-heuristic approach.

In Ref. 19, a three-step algorithm is used to detect the information related to the grid: preprocessing, rotation detection, and local gridding refinement. Some global parameters are first estimated by a simple preprocessing heuristic. For the rotation detection, the microarray image is iteratively rotated by a quarter degree until its projection profile closely matches to the normal one. The gridding refinement uses the global values obtained in the preprocessing step to have some guess about the location of each subarray. The parameter estimation is simple mainly when applied on subarray structures.

In Ref. 20, an automatic iterative algorithm is proposed. The algorithm assumes that spot centers deviate from a sequence of similarity transformations whose parameters vary smoothly. Using this assumption, the authors can formulate the spot center gridding problem as a constrained optimization problem combining a quantitative criterion that measures gridding result correctness with some constraints that reduce local parameter variation. The problem is solved by analyzing the cause of the deviation of the spot centers, assuming that spot center deviations can be modeled by the following parameters: scaling, rotation, and translation. The mean squared error e_e of all matched centers is defined. Also, a smoothness constraint by minimizing variation is introduced together with an error measure e_s of the smoothness. The problem is solved by searching the solutions that minimize a weight sum $e_e + \lambda e_s$, where λ is a nonnegative parameter, by a numerical iterative algorithm. It assumes that each block in the analyzed microarray has the same rotation angle, both in the initial distortion estimation and in a tree-based outlier correction, so that a subblock with a different rotation angle is considered an "outlier" and relative parameters are consequently adjusted. Each step is iteratively performed. Our approach is based on the idea that every grid in a microarray may have a different rotation angle, which occurs independently from the neighboring grid. Our methods for rotation detection and gridding use a single-step algorithm, because it assumes that neighboring grid parameters are independent.

The gridding method proposed in Ref. 21 uses a scheme that combines global and local segmentation mechanisms for defining the boundaries of each microarray spot. It initially creates *global* boundaries, using the middle point of two successive peaks related to the sums of R and G intensities along the rows and columns of the microarray image. In the next step, the global boundaries are refined as follows. The horizontal (vertical) final boundary between two spots is refined by locating the minimum of the sum of the rows (columns) and taking into account only the area left out the global boundary grid of these spots. Working directly on pixel values, this approach may be affected by perturbation induced by the presence of noise. A more robust solution should be based on the binary guide mask obtained by effective and accurate spot segmentation.

3 Preprocessing: A Microarray Image Rotation Algorithm

In this section, we formalize MIRA, an algorithm based on histogram analysis able to automatically detect and correct rotation problems of microarray grids. The technique is designed for orthogonal grids, the most common type for microarrays. It can be formalized as follows. Let I be a black-and-white binary image. $f: I \rightarrow N$ is defined as

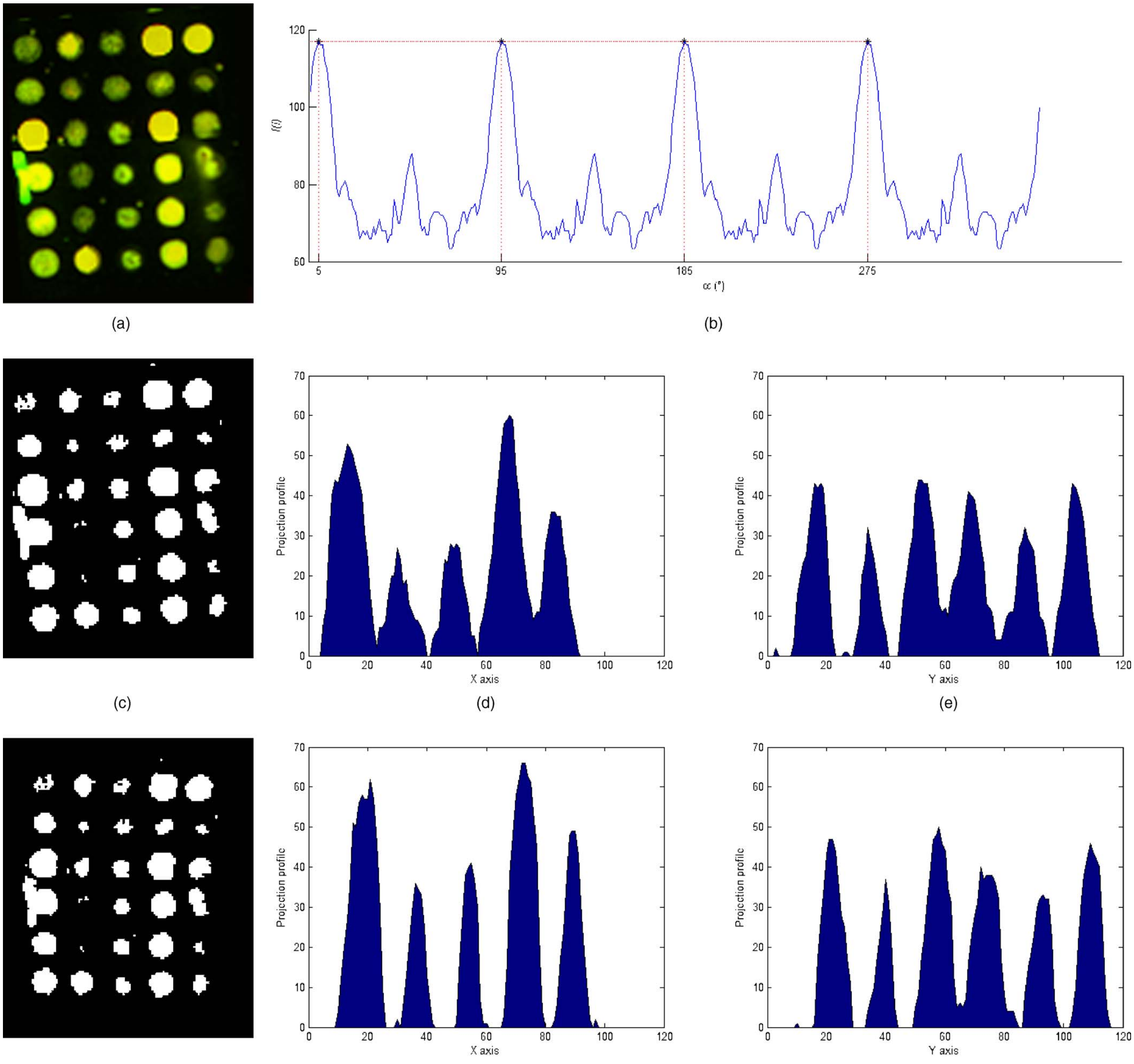
$$f(I) = \max(h(I)) + \max(v(I)), \tag{3}$$

where $h(I)$ and $v(I)$ are, respectively, the integral projections profile[22] along the x- and y-axes of I, obtained by summing up the binary value of each pixel in I (0 for black and 1 for white) [Fig. 2(d) and 2(e)]. Let M be a binary map [Fig. 2(c)] of a microarray image [Fig. 2(a)] that captures where the spot pixels are approximately located in both input channels. One way to obtain M is to partition each original microarray channel into two classes (using the K-means algorithm[23]) and then combining the resulting binary images by the logical OR.

Let M_α be the map obtained by rotating M of α radians. Since $f(M_\alpha) = f(M_{(\alpha + k * \pi/2)})$, $\alpha_{\max} = \pi/4$. The estimate correction angle α^* is defined as

$$\alpha^* = \underset{-\frac{\pi}{4} < \alpha < \frac{\pi}{4}}{\operatorname{argmax}} \{f(M_\alpha)\}. \tag{4}$$

In order to find the main direction of the grid, we just consider the directions of the projections. This allows us to select the direction having the maximum value of $f(I)$ [Fig. 2(b)], corresponding to the angle where the maximum number of aligned spot centers is located. The pseudo-code of the algorithm is



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 2 Main steps involved in the MIRA algorithm.

```
Input: a binary map I of input microarray;
$\max_f = f(I)$, $\alpha^* = 0$;
For $\alpha = \text{from} - \alpha_{\max}$ to $\alpha_{\max}$ do
    $I' = \text{Rotate}(I, \alpha)$
    If $f(I') > \max_f$ then
        $\max_f = f(I')$;
        $\alpha^* = \alpha$;
```

M is hence affine-transformed by a rotation of angle α^* around the image's center [Fig. 2(f)–2(h)]. A simple bilinear interpolation is used to reconstruct the signal after rotation. We safely assume that both input channels have the same rotation angle.

MIRA is based only on the value of $f(I)$, which depends only on the entire grid profile, without comparison with other profiles. This property makes the system reliable, since an error in detecting a grid rotation will not affect the others and there is no risk of entering a loop. A rotation test is always done, since there is no way for MIRA to know *a priori* if the grid profile shows no rotation.

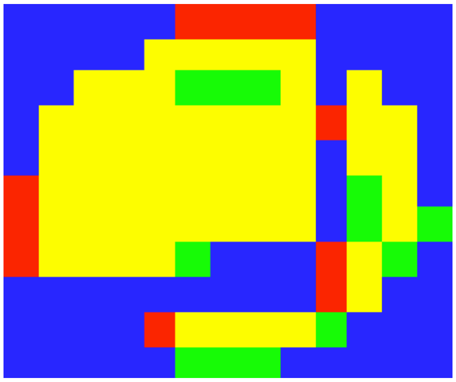


Fig. 3 Microarray image semantic color region. Background (*black*), local background (*blue*), red channel foreground (*red*), green channel foreground (green), red channel and green channel foreground (yellow).

4 Segmentation Pipeline

In this section, we briefly describe the main steps of the image segmentation pipeline MISP.[13] The proposed process is fully automatic. The technique processes each microarray image to produce five semantic regions (Fig. 3):

- background,
- local background,
- red channel foreground,
- green channel foreground,
- red channel and green channel foreground.

The pipeline can be ideally subdivided into two sequential modules (Fig. 4):

- spot-background separation (Fig. 5),
- foreground and local background identification (Fig. 6).

The *Spot-Background separation* module identifies the spot signal pixels from the background. Using statistical region merging (SRM[24]) on each channel, it is possible to extract the spot shape by making use of the local mean intensity rather than the single pixel value intensity. Further processing is devoted to better distinguish involved signals by making use of *ad hoc* γ LUT[25] and k means clustering.[26] The shape is further refined at the intensity of edges by taking into account the deviation of edge pixels from the local mean. Two binary masks, *GBin* and *RBin*, are the output of the *Spot-Background separation* module and become the input of the next module.

The second module, *Foreground and Local Background identification*, identifies *GBin* and *RBin* with *Red Mask Foreground* (*RMF*) and *Green Mask Foreground* (*GMF*). It also builds a *Spot Guide Mask (SGM)* as the logical OR of these two maps.

Moreover, the set of pixels belonging to *SGM* but not to *RMF* (*GMF*) are said to be the internal background relative to the red (green) channel for the spot.

Let *Grid Guide Mask* (*GGM*) be the minimum square containing *SGM*. The difference between *GGM* and *SGM* forms the *RGBackMask*. The local background relative to

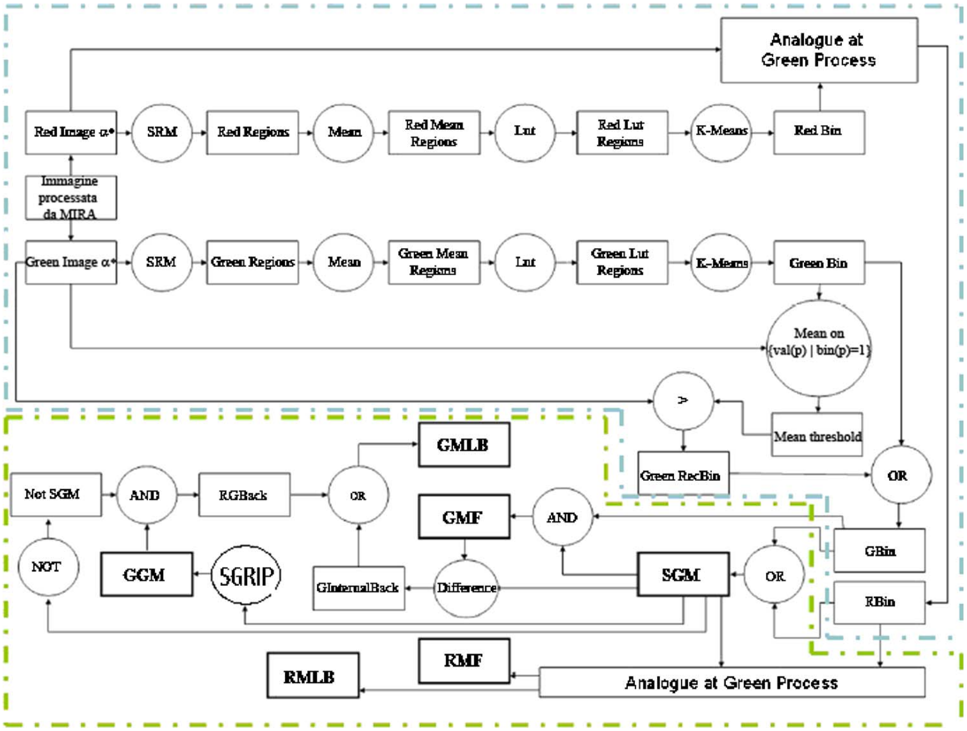


Fig. 4 MISP: microarray image segmentation pipeline. Cyan dashed line refers to the steps involved in the spot-background separation module introduced in Sec. 4. Green refers to the steps involved in the foreground and local background identification module introduced in Section 4 (see Ref. 13 for more details).

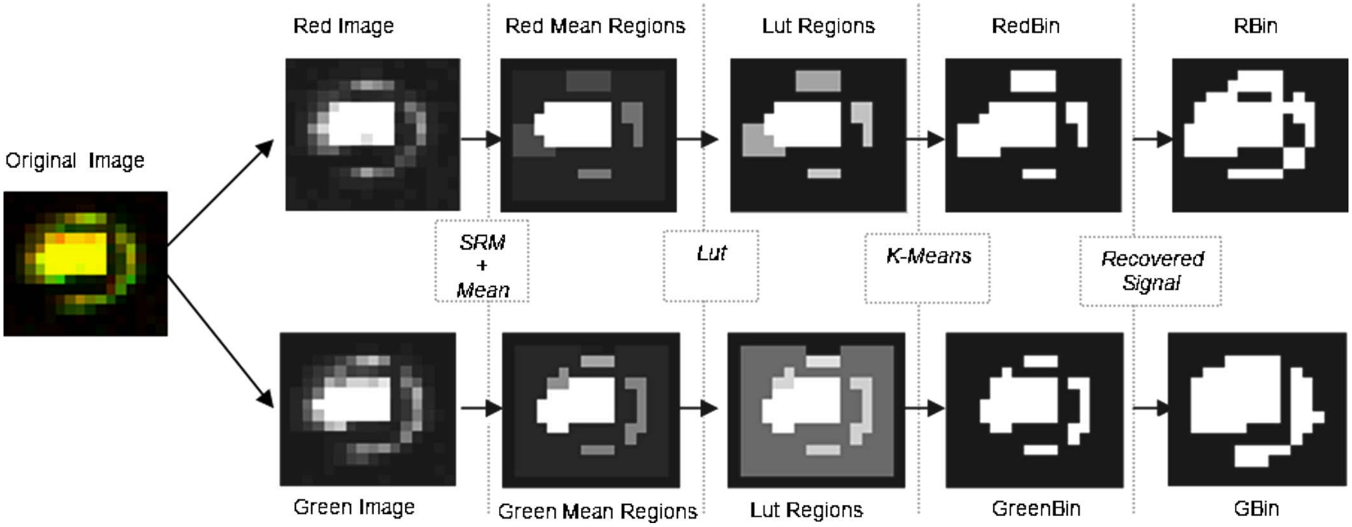


Fig. 5 Spot–background separation.

the red channel is obtained by augmenting *RGBackMask* with the pixels belonging to the internal background relative to the red channel. The local background relative to the green channel is obtained similarly.

Figure 7 summarizes the overall process together with the involved binary masks. Major details can be found in Ref. 13.

We point out that the SGRIP module described in the following section provides a more accurate detection of GGM than the one presented in Ref. 13, which is not able to deal with the neighboring spot merging problem or large, noisy areas. A variety of quality measures and useful data may be obtained by gathering information inside the different masks created so far. *SGM* is used to derive quality measures for each spot (e.g., spot area measure). *GGM* is used to assign coordinates to each spot. Other masks are used to characterize the pixel belonging to *foreground/background/local background*, to calculate intensity,[3,27] and to extrapolate quality measures for each spot.[4,5,17]

5 Statistical Gridding Pipeline

The authors have proposed a simple gridding technique.[13] That algorithm is supplanted by a more effective approach that has been implemented into the new module called SGRIP. The pipeline for SGRIP includes two phases (Fig. 8):

- *Grid Finding—Correction*,
- *GGM Creation—Refinement*.

In the first phase, *Grid Finding* approximates the spot centers. It works on *SGM* by assuming a local homogeneous background. The final output is obtained after a *Correction* step to recover spot centers that have been missed so far. The spot center prototypes are stored in an $m \times n$ matrix P, where m and n are, respectively, the inferred number of rows and columns in the array.

The second phase is able to produce a *GGM*, starting from the data in matrix P. *GGM Creation* uses P and *SGM*

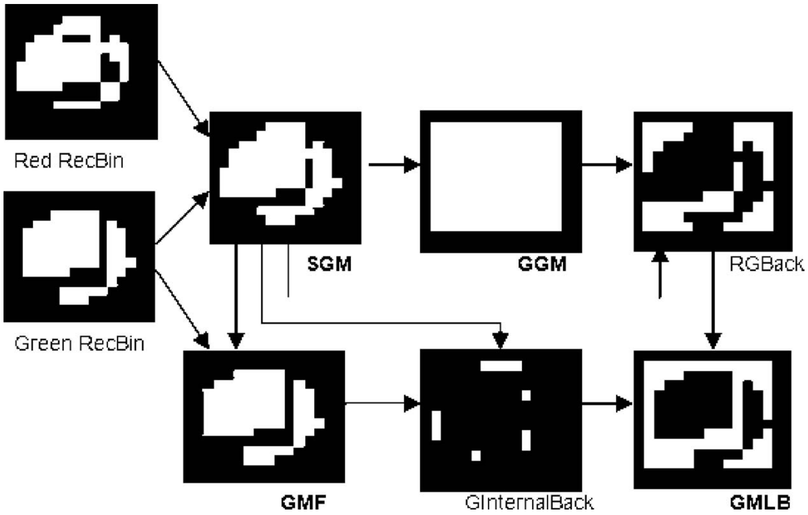


Fig. 6 Channel foreground and local background identification.

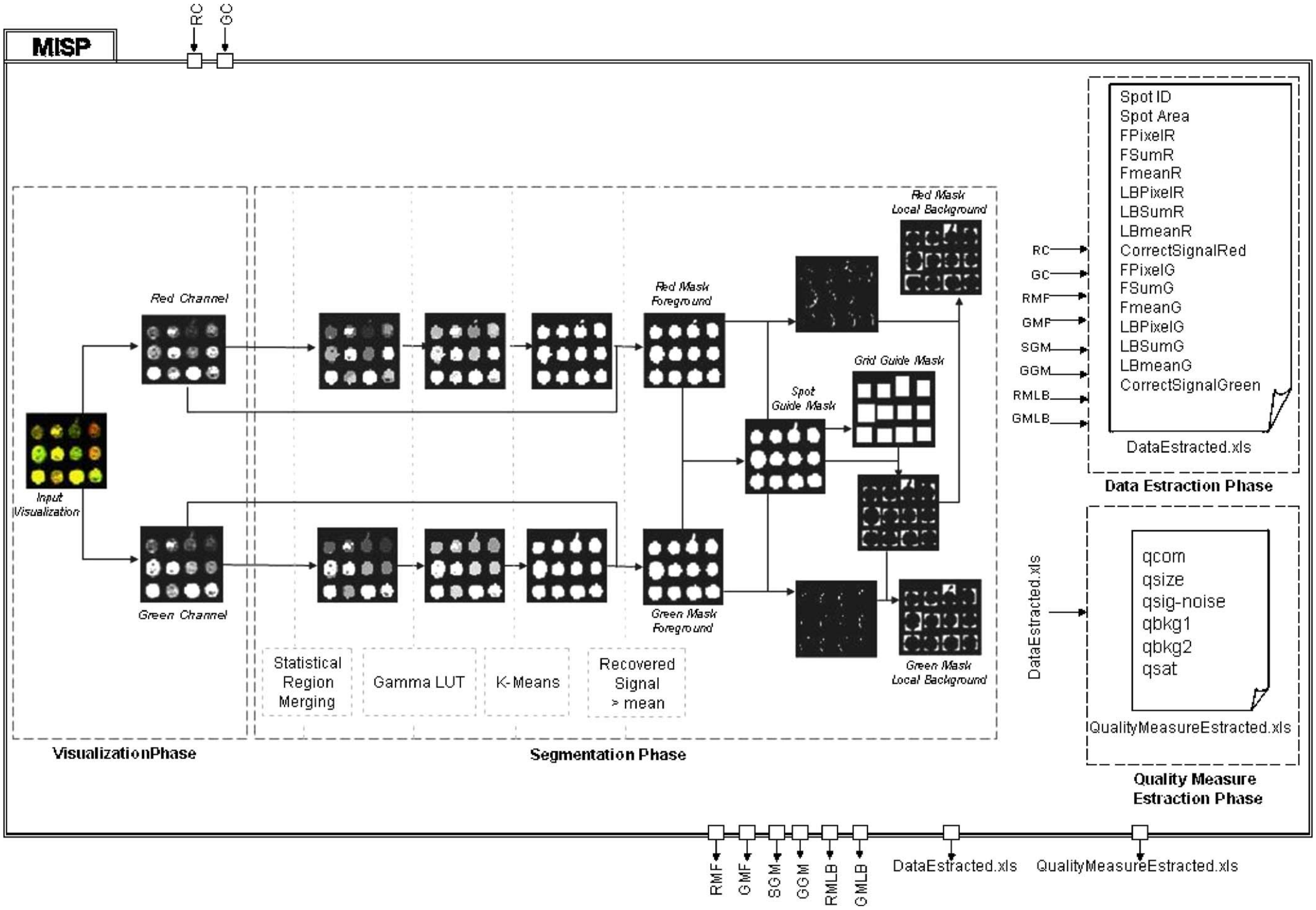


Fig. 7 MISP: software prototype architecture. Involved details are described in Ref. 13.

to create a first approximation: to each simple connected component in *SGM* is assigned the minimum rectangular region containing the component. The final step (*GGM Refinement*) separates spots erroneously merged with others. We assume that SGRIP is performed on previously correctly rotated images. In the following subsections, we describe in greater detail the *Grid Finding*, *Grid Correction*, *GGM Creation*, and *Refinement* steps.

5.1 *Grid-Finding Algorithm*

Grid Finding detects the grid location and the number of spot rows and columns in the grid. We assume that both horizontal and vertical histograms of *SGM* have the typical shape of an almost regularly spaced sequence of peaks separated by a valley. In the following, the term “Expected Values” refers to the expected value of a Gaussian-like distribution. The typical shape of a single sequence can be easily approximated by a Gaussian-like distribution (e.g., doubly truncated.[28])

The algorithm is:

Input: a binary map *SGM* of input microarray;

1. Let *Hh* and *Vh* be, respectively, the horizontal and vertical cumulative histogram of *SGM*;
2. Let *MHh* and *MVh* be, respectively, the mean of *Hh* and *Vh*;
3. Let $CHh = Hh - MHh$;
4. Let $CVh = Vh - MVh$;
5. Let HG_{SGM} and VG_{SGM} be the family of peaks, respectively, in *CHh* and *CVh*;
6. Let $EHG_{SGM} = \{\text{Expected Values}(d) : d \in HG_{SGM}\}$;
7. Let $EVG_{SGM} = \{\text{Expected Values}(d) : d \in VG_{SGM}\}$;

Output: *X*, the set of couple (i,j), where $i \in EHG_{SGM}$ and $j \in EVG_{SGM}$.

Step 3 in the algorithm separates each peak in the sequence from the others. A side effect of steps 3 and 4 is that if a row or column contains just a few spots, the corresponding peak could be lost (Fig. 9); this leads to the need for the next correction block.

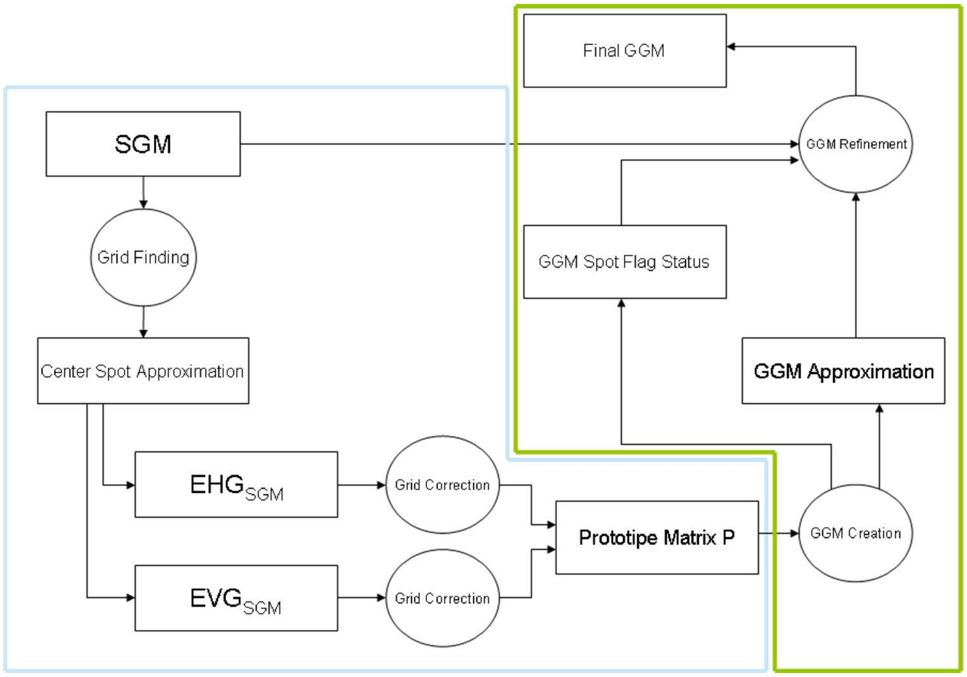


Fig. 8 SGRIP pipeline: The cyan line refers to the Grid Finding-Correction phase introduced in Section 5 and detailed in Sections 5.1 and 5.2, while the Green line refers to the GGM Creation-Refinement phase introduced in Section 5 and detailed in Sections 5.3 and 5.4.

5.2 *Grid Correction*

A *Correction* algorithm is used to reconstruct the missed spots and to correctly infer the number of spot rows and columns. This algorithm is applied separately for row and column coordinates. The algorithm applies the simple idea that if a row (column) has been missed, then the distance between two successive peaks has to become larger than the median gap between. We adopt the following simple rule to identify missed spots: If the distance between one peak and the successive one is greater than $K \times m$, with m equal to the median of all the gaps between the spots in the sequence, then there is a missing row (column). The rationale of the above rule is the follows. Ideally, each microarray grid is composed of regularly distributed rows and columns and spots are perfectly circular. Considering horizontal and vertical cumulative histograms, in an ideal case one can observe a typical pattern of an almost regularly spaced sequence of peaks separated by a valley. Each column contains the same number of spots and is equidistant by previous and successive columns ($\Delta x = constant$); hence, each column histogram, $i=1,\ldots,\#columns$, may be represented with a doubly truncated Gaussian distribution.[28] In reality, many uncontrollable factors are involved in shaping the signal distribution for microarray experiments, so each double-truncated Gaussian distribution can be approximated by a Gaussian distribution $N_i(\mu_i,\sigma)$, where $\mu_i = \mu_1 + (i-1) \times \Delta x$. Each Gaussian can be obtained by shifting the previous mean μ by Δx and using the same variance σ.

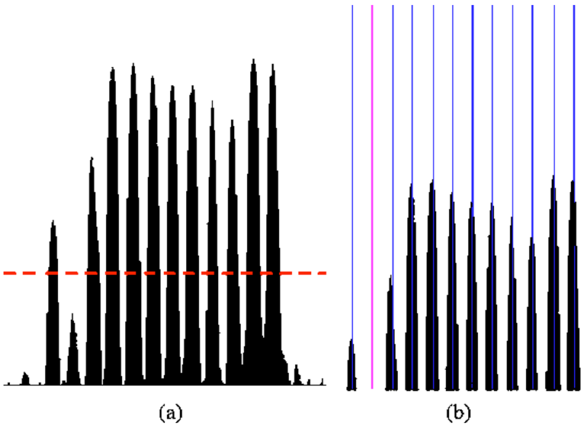


Fig. 9 (a) An example of the horizontal cumulative histogram of SGM (*Hh*). Red dashed line indicates the mean value (*MHh*). In this example, one of the Gaussians is below the mean value. (b) The corresponding family of the Gaussian HG_{SGM}. The expectation value for each Gaussian in HG_{SGM} corresponds to the central point of the spots column (blue lines). The purple line corresponds to the one unlocalized column that will be referred to successively.

The parameters μ and σ are not constant, due to microarray problems; however, when a grid has a large number of spots, it is possible to approximate Δx with the median m of the distances between columns. Hence, $\mu_i = \mu_1 + (i-1) \times m$.

We observe that a random variable $\chi \approx N(\mu,\sigma)$ has $P(\mu - 3\sigma \le \chi \le \mu + 3\sigma) \approx 1$; taking into account the above consideration, we can approximate this value with

$$P\left(\mu - \frac{K \times m}{2} \le \chi \le \mu + \frac{K \times m}{2}\right) \approx 1.$$

The *Grid Correction* algorithm uses the parameter K (line 6) below with the following meaning. If in the interval $[\mu - K \times m/2, \mu + K \times m/2]$ computed for each couple of

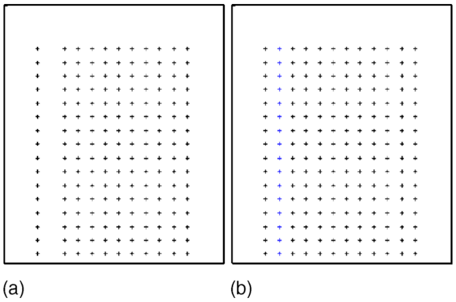


Fig. 10 Spot centers (a) before and (b) after the *Grid Refinement* step. In blue are the inferred positions for an unlocalized spot column in the *Grid Finding* step.

columns found by the *Grid Finding* algorithm, no column is inside the interval, then with high probability a column will be reconstructed using the median value *m*. This is justified by the above considerations. The algorithm can be synthetically sketched as follows:

```
Input: an array $Coord$ of sorted coordinates (e.g., $EVG_{SGM}$)
For each couple $\zeta$ of successive coordinates in $Coord$
   Calculate $d_{ij}(\zeta)=Coord_{i+1}-Coord_i$
   Insert $d_{ij}(\zeta)$ in a vector $D$ with position $i$;
$i=0$;
While i $<|D|$
   If $D_i > K^*$median($D$)
      $|Coord|=|Coord|+1$;
      Insert a new element in $Coord$ between $Coord_i$ and
      $Coord_{i+1}$ with value $Coord_i$+median($D$);
      $|D|=|D|+1$;
      Insert a new element in $D$ between $D_i$ and $D_{i+1}$ with value
      $D_i$-median($D$);
      $D_i$=median($D$);
   $i=i+1$;
```

The output is a new matrix *P* of pairs mapping the final prototype spot centers (Figs. 9 and 10). In our experiments, the *K* parameter has been estimated to be equal to 1.8 using the least-squares method on a training microarray data set. The position of the missing row (column) is restored by using the median as an *ad hoc* guess. We use the median in order to reconstruct the rows (columns), because it is more robust than the mean value, which is typically unstable when the number of missing rows (columns) is large. Moreover, using the median rather than the mean, we obtain a single-step method to estimate the positions of missing spots; we do not need to reestimate the guess after inserting a missed spot. Analog considerations can be done for the GGM refinement algorithm reported in Section 5.4.

5.3 *GGM Creation*

The *Grid Guide Mask Creation* algorithm starts with a matrix *P* of pixel coordinates. These coordinates are used as starting points to assign to each detected spot the minimum rectangular region containing the spot itself computed over the SGM (Fig. 11).

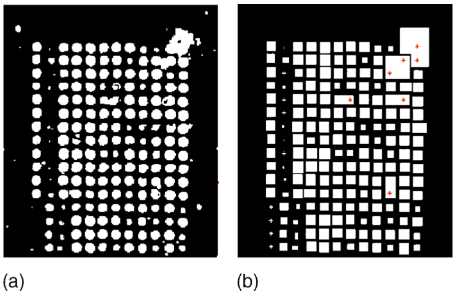


Fig. 11 GGM creation: (a) SGM input obtained by the segmentation pipeline and (b) the corresponding output after the GGM creation step.

The *Grid Guide Mask Creation* algorithm can be described as follows:

```
Input: an $m \times n$ matrix $P$ of pixel coordinates and $SGM$;
For each row $i$ of $P$
   For each column $j$ of $P$
      Let $(x,y)$ be the coordinates in $P(i,j)$
      Initialize $count$ with the number of the neighboring signal
      pixels of $P(i,j)$ having zero value;
      If $SGM(x,y)>0$, then
         push in a stack $S$ the coordinates $(x,y)$;
         Initialize the corners $(\min_x,\max_x)$, $(\min_y,\max_y)$ of the
         smallest rectangular region related to $P(i,j)$ at value
         zero;
         while nonEmpty($S$)
            $tmp$=pop($S$);
            $count=count+1$;
            if $(\min_x,\max_x)$, $(\min_y,\max_y)$,changes with respect to
            $tmp$, update the corner information;
            Mark the coordinates $(tmp_x,tmp_y)$ as $controlled$;
            Let
            $R=\{p:p$ is a pixel such as the distance from $P(i,j)$ is
            less than $Range$, $p$ is $uncontrolled$ and
            $SGM(p_x,p_y)>0\}$
            If $R\neq\emptyset$, then mark as $uncontrolled$ all $p\in R$ and
            push them in $S$;
         If $count>1$
   GGM$(i,j)=[(\min_x,\min_y),(\max_x,\max_y)$, $found$']
         else GGM$(i,j)=[(x,y),(x,y)$,'$not\ found$'];
```

For each seed in $P(i,j)$, we create a record in position (i,j) in the GGM matrix containing the coordinates of the

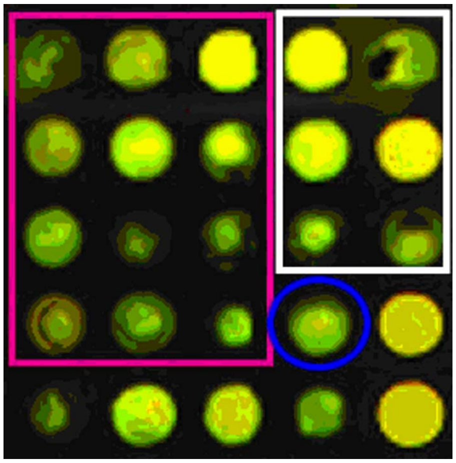


Fig. 12 Regions involved in GGM refinement. The blue circle indicates the s_n spot. The white line encloses the $R_{n\_ne}(s_n)$ region, while the magenta line encloses the $R_{w\_nw}(s_n)$ region.

top left and the bottom right corners of the minimum rectangular region that contains the spot guide, and the status of the corresponding spot ("*found*" or "*not found*"). The search of the minimum region containing the spot is carried out only within a square area of side equal to $2\times Range + 1$. This is useful to include spots that exhibit doughnut shapes or that are split into two or more connected components. Note that when the GGM rectangle stops growing, there are only two mutually exclusive motivations:

1. no more foreground spot pixels are outside the rectangle (this includes the special case of spot absence);
2 the rectangle is merged with some previously processed rectangle.

This means that the rectangle hull of two or more spots is not disjoint.

Observe now that since spot center coordinates are processed in left-to-right, top-to-bottom fashion, the merging of the rectangle relative to the spot with the center (x_n, y_n) may happen only with rectangles relative to spots with center (x_f, y_f) and the following holds:

$$((x_f < x_n)\mathrm{AND}(y_f \leq y_n))\mathrm{OR}((x_f \geq x_n)\mathrm{AND}(y_f < y_n)). \quad (5)$$

This property is used ahead to restore spots erroneously merged with others.

5.4 GGM Refinement

The final *GGM* is obtained by a *refinement* algorithm that is designed to solve the overmerging problem. Observe that one rectangle may be merged only with a rectangle located W, N, NW, or NE of it. The refinement strategy is based on the following claim: If a spot s_n is marked as "*not found*," but a foreground region that can be assigned to s_n exists, then the s_n region has been assigned to another spot s_f, where $s_f \in C$, with $C = R_{w\_nw}(s_n) \cup R_{n\_ne}(s_n)$, where $R_{w\_nw}(s_n)$ and $R_{n\_ne}(s_n)$ are the two regions (Fig. 12). These two regions are in correspondence with the two disjoint clauses in Eq. (5). Following the same order (left to right and top to bottom), we can restrict C to the region in which the spots of the row preceding s_n or the spot located at the left of s_n. In the following pseudo-code, this region is denoted by $Cr(s_n)$ for northwest merging case. The other three cases are similar.

```
Input: SGM, GGM and spot status flag

1.  Let $Wset=\{w: w$ is value of width of a region in GGM with status="found"$\}$;
2.  Let $Hset=\{h: h$ is value of height of a region in GGM with status="found"$\}$;
3.  Let Wmedian and Hmedian be the medians of Wset values and Hset values, respectively;
4.  Let Wvar and Hvar be the variances of Wset and Hset values, respectively;
5.  $\forall$ spot $s_n$: status($s_n$)="not found" do
6.    If $\exists s_f \in C_r(s_n): s_{n\_area} \subseteq s_{f\_area}$;
7.      If $s_f$ is a northwest neighbor of $s_n$
8.        Let $Hs_f$ and $Vs_f$, respectively be the horizontal and vertical cumulative histogram of the $s_f$region
9.        Let $p$ be the point in SGM in which $Hs_f$ and $Vs_f$ assume minimum values
10.       If$(|p_x - s_{fx}| < K\times Wvar/2)$ and $(|p_y - s_{fy}| < K' \times Hvar/2)$
11.         Split $s_f$ region into four parts considering $p$, top left and bottom right corners
12.         Update $s_f$ region with NW subregion obtained in step 11
13.         Assign to $s_n$ the SE sub region obtained in step 11
14.       Else
15.         assign to $s_f$ and $s_n$ two regions whose dimensions are Wmedian·H median center to $s_f$ and $s_n$
16.     If $s_f \in$ other case
17.       process is similar way for northwest case
```

The values K and K' are obtained by variance analysis on *Hset* and *Vset* of different microarrays. In our experiment, $K=1.2$ and $K'=1.3$.

Figure 13 shows the final output of the overall SGRIP pipeline applied to the input image SGM reported in Fig. 11.

6 Quality Measures

One of the main drawbacks to microarray imaging tools and algorithms is the difficult task of evaluating the real performances of each involved technique. Various quality measures have been proposed in the literature.[4,5,17] Here we propose a new quality index defined as follows:

$$q_{\mathrm{index}}(IDSpot) = \frac{q_{com2R}(IDSpot) + q_{com2G}(IDSpot)}{2}, \quad (6)$$

where

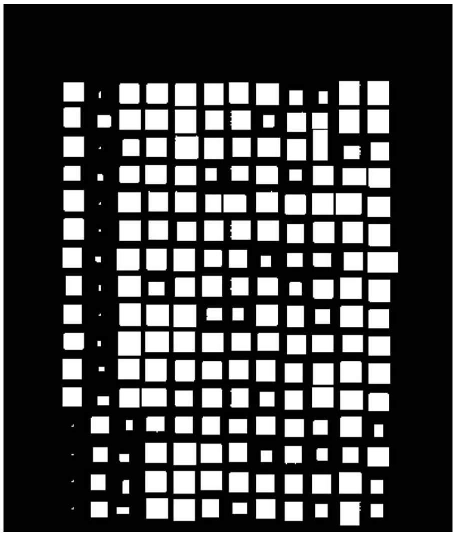


Fig. 13 GGM refinement output.

$$q_{com2} = (q_{sig-noise} \times q_{bkg1} \times q_{bkg2})^{1/3}. \quad (7)$$

q_{index} is the mean of q_{com2} computed for each channel. q_{com2} is partially derived from the *combined quality index* (see Ref. 4 for details):

$$q_{com} = (q_{size} \times q_{sig-noise} \times q_{bkg1} \times q_{bkg2})^{1/4} \times q_{sat}. \quad (8)$$

The *Combined quality index* (q_{com}) encloses the size of the spot (q_{size}), the signal-to-noise-ratio ($q_{sig-noise}$), the local background variability (q_{bkg1}), excessively high local background (q_{bkg2}), and saturation in photo intensity detection (q_{sat}). q_{size} assesses the irregularities of spot size, $q_{sig-noise}$ is a measure for the signal-to-noise ratio, q_{bkg1} quantifies the variability in local background, q_{bkg2} is the level of the local background, and q_{sat} indicates if the percentage of the saturated pixel is less than 10% for each spot.

The original *combined quality index* q_{com} has been modified to be used for software comparison rather than only as a measure for flag checking associated with each spot. Preliminary results demonstrate that the measures q_{size} and q_{sat} involved in q_{com} may produce errors in evaluation and comparison of different segmentation methodology because of the following considerations:

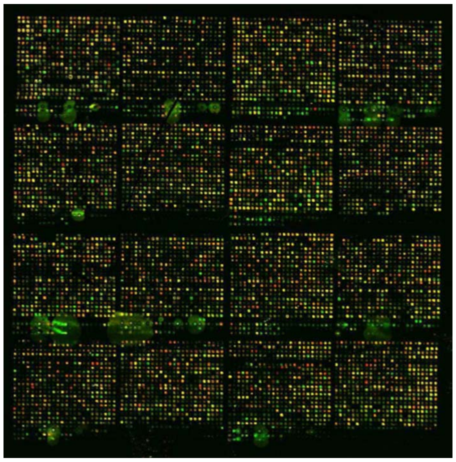


Fig. 14 Microarray image from the Stanford microarray database, ExptID 15739.[34]

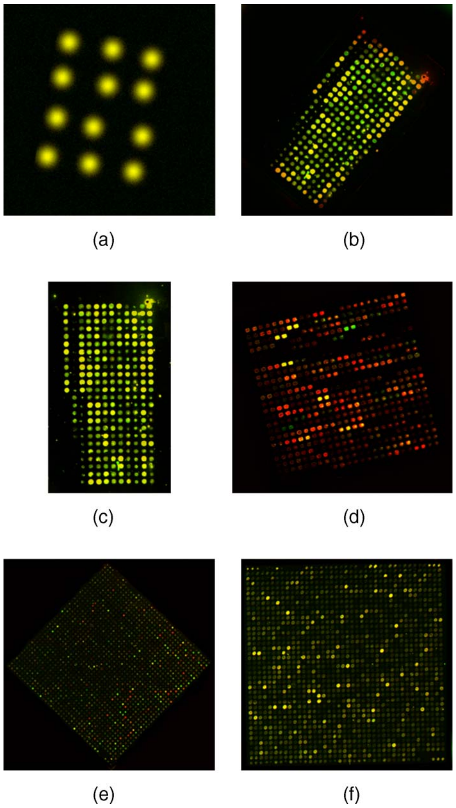
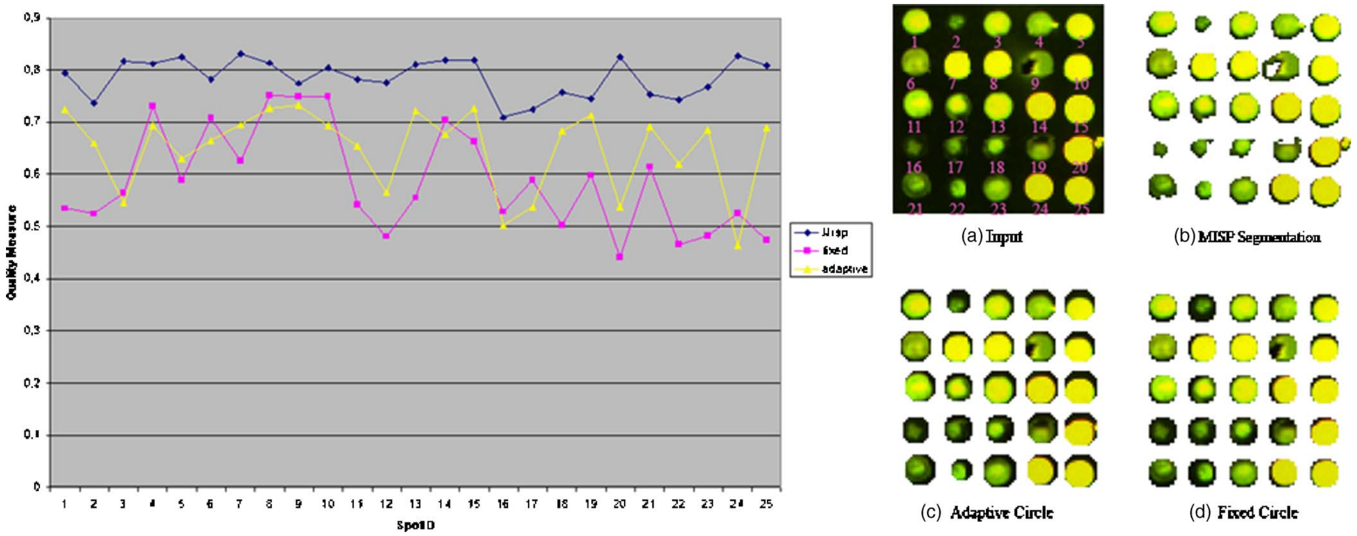


Fig. 15 Examples of microarray used to test MIRA.

- q_{size} is related to the regularity of the spot with respect to an ideal spot whose dimension equals the mean of the spots in the microarray. Fixed- and variable-circle segmentation produce foreground masks in which the background is included or the irregular foreground is discarded. In the “fixed” case, each spot in the mask has the same dimension, hence, the q_{size} for each spot in the microarray. Adaptive techniques instead produce irregular foreground masks, and the background is discarded; hence, $q_{size}<1$ even if the method is more reliable than a “fixed” circle. q_{size} is useful for comparison of segmentation techniques that use the same methodology (fixed or adaptive).
- Analogously, q_{sat} penalizes the adaptive techniques even if they are better than fixed- by or variable-circle techniques. Example: Let s be a spot with 4 saturated pixels. Suppose that the spot is segmented using the fixed-circle methodology with a circle area of 50 pixels in which 10 pixels belong to the background. In this case, $q_{sat}=1$ because the saturated pixels are less than 10% (5 pixels) of the spot area. Suppose that the same spot is segmented correctly using an adaptive technique. In this case, the area of the spot is 40 pixels and $q_{sat}=0$. Hence, q_{sat} is not useful for comparison of adaptive segmentation techniques.



(a) Input

(b) MISP Segmentation

(c) Adaptive Circle

(d) Fixed Circle

Fig. 16 Results obtained using different methods of segmentation on the microarray M_7. The plot of the corresponding q_{index} relative to different segmentation methods. In (a) the input microarray is reported. In (b) the segmentation obtained by MISP is shown. (c) and (d) are obtained using *Scanalyze*.

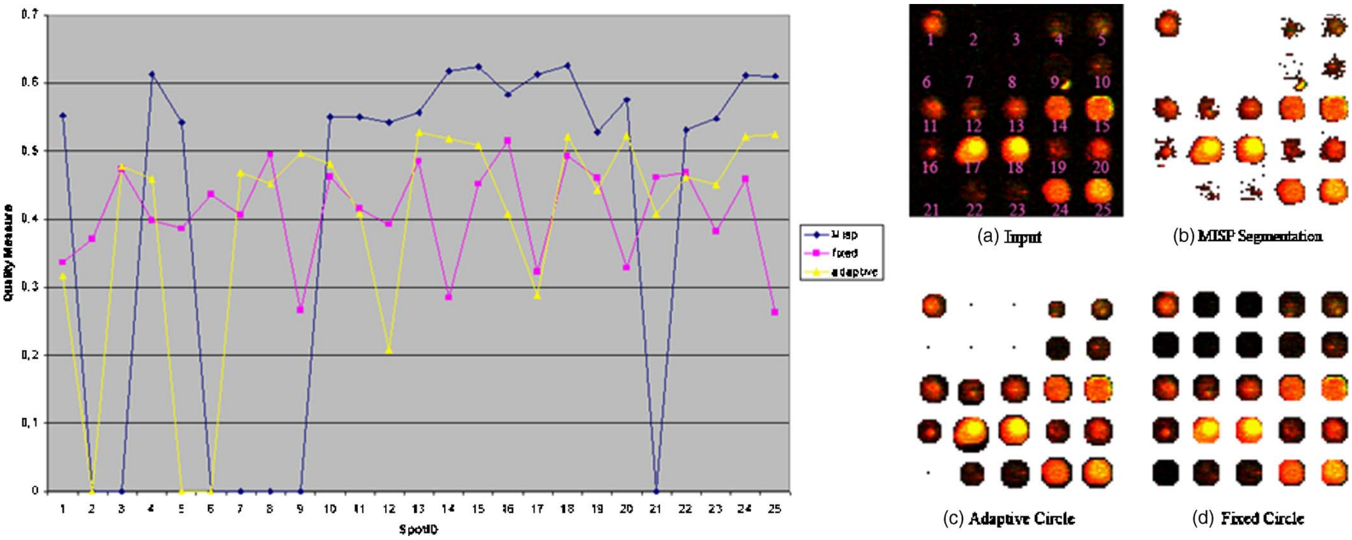
The other measures, $q_{sig-noise}, q_{bg1}, q_{bg2}$, are calculated for each channel and report the goodness in terms of foreground-background separation.

7 Experiments and Discussion

The first image data set used for testing each involved module in MIAF refers to the "Whole Yeast Genome" microarrays, freely downloadable at the MAGIC Website.[29] Microarray data denoted by M_i, $i=1,\ldots,18$, addressed to specific problems (rotation, segmentation, gridding), have been selected in order to exemplify the MIAF performance. Comparison with the output obtained with *Scanalyze*[9] has been carried out. We believe that the main strength of our adaptive approach is revealed when it is compared with techniques based on circle segmentation. Only by using adaptive segmentation strategies the real amount of gene expression for each spot be effectively managed. For fairness of comparison, hence, care has been given in order to use *Scanalyze* with the best possible choice of user-selected parameters. In particular, parameters have been tuned to obtain optimal quality measures. MIAF is able to perform quality measurement also on an imported *Scanalyze* grid: This makes comparison easier. MISP segmentation performances have been compared with *Scanalyze* also using a calibration data set[30] accessible from the U.S. National Cancer Institute, generated by Incyte Genomics for the purpose of assessing quality assurance parameters within microarray experiments. To test the gridding performed by MIAF (SGRIP) more accurately, we also consider the microarray data set used in Ref. 31, related to the whole genome of *Saccharomyces cerevisiae* and freely downloadable at Pat Brown's lab homepage.[32] Further testing has been carried out referring to the collection of microarray images available in the Stanford Microarray Database (SMD).[33] Using SMD, researchers are able to store, retrieve, display, and analyze the complete raw data produced with one of the interactive image processing platforms compatible with SMD. In particular, we refer to the experiments ExpID 15739[34] (Fig. 14) and ExpID 51509.[35] We compare our results with the results obtained in Ref. 12, where Spot[6] has been used on the same data set. We apply

Table 2 Experimental rotation assessment of MIRA on M_1, M_2, M_3, M_4, M_5, and M_6.

Microarray	Rows	Columns	Absent Spots in %	Real Rotation Angle α (in degrees)	Angle Estimate by MIRA	Error
M_1	4	3	0	−10°	−10°	0°
M_2	24	12	6.25	−38°	−39°	1°
M_3	24	12	6.25	1°	1°	0°
M_4	23	24	14.7	14°	13°	1°
M_5	42	40	0.71	−44°	−44°	0°
M_6	40	40	1.44	1.2°	1°	0.2°



(a) Input (b) MISP Segmentation

(c) Adaptive Circle (d) Fixed Circle

Fig. 17 Results obtained using different methods of segmentation on the microarray M_8. The absent spots are correctly identified by our processing pipeline: The corresponding q_{index} is equal to zero.

the proposed pipeline algorithms to each inner grid.

7.1 *Rotation Tests*

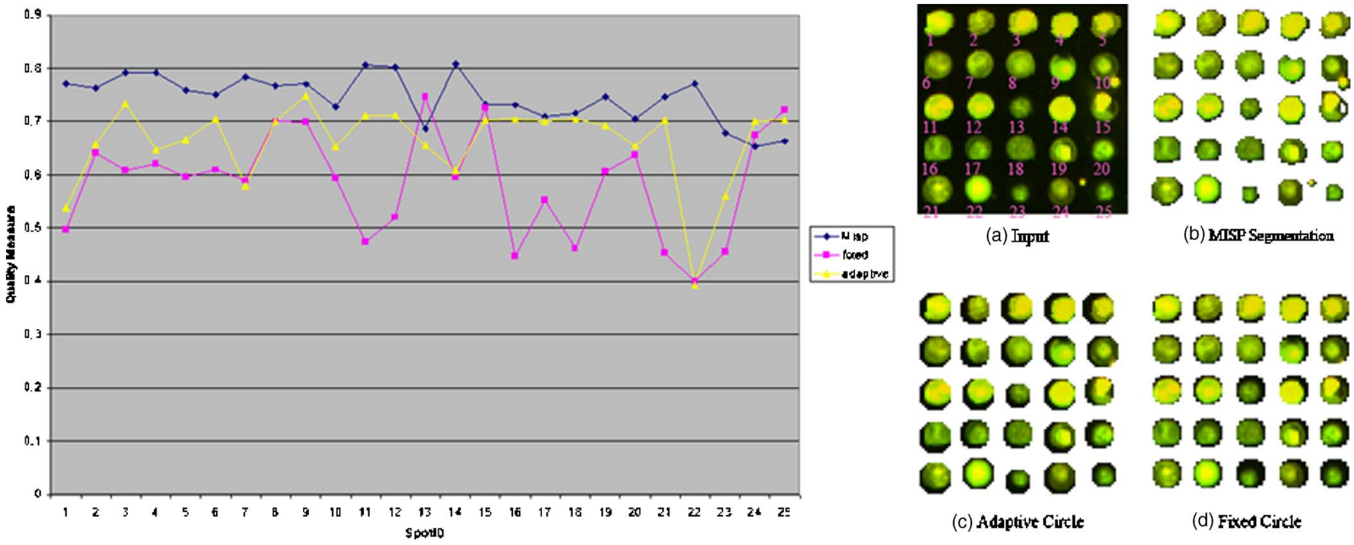
To test MIRA, we use a data set in which each microarray has been previously rotated with a (known) global rotation angle. More precisely, microarrays have been manually rotated by angles in the range $[-44°, +44°]$ (Fig. 15). Some results obtained for microarray M_1–M_6 are reported in Table 2; they confirm the good performances of the proposed technique (error mean: 0.36; error std: 0.49) in case of both large and small rotations. The method is not sensitive to the number of involved spots; however, for grids having thousands of spots, the MIRA performances benefit from the major statistical robustness of $h(I)$ and $v(I)$.

7.2 *Segmentation Tests*

To evaluate MISP algorithm performance, we have performed accurate tests for both visual and numerical assessment. The tests are performed taking into account 114 different spots that can be classified as follows:

1. 42 *spots* with good signal intensity and a clear circularity shape;
2. 34 *spots* with irregular shape and good signal intensity;
3. 38 *spots* with low signal intensity and shape variability.

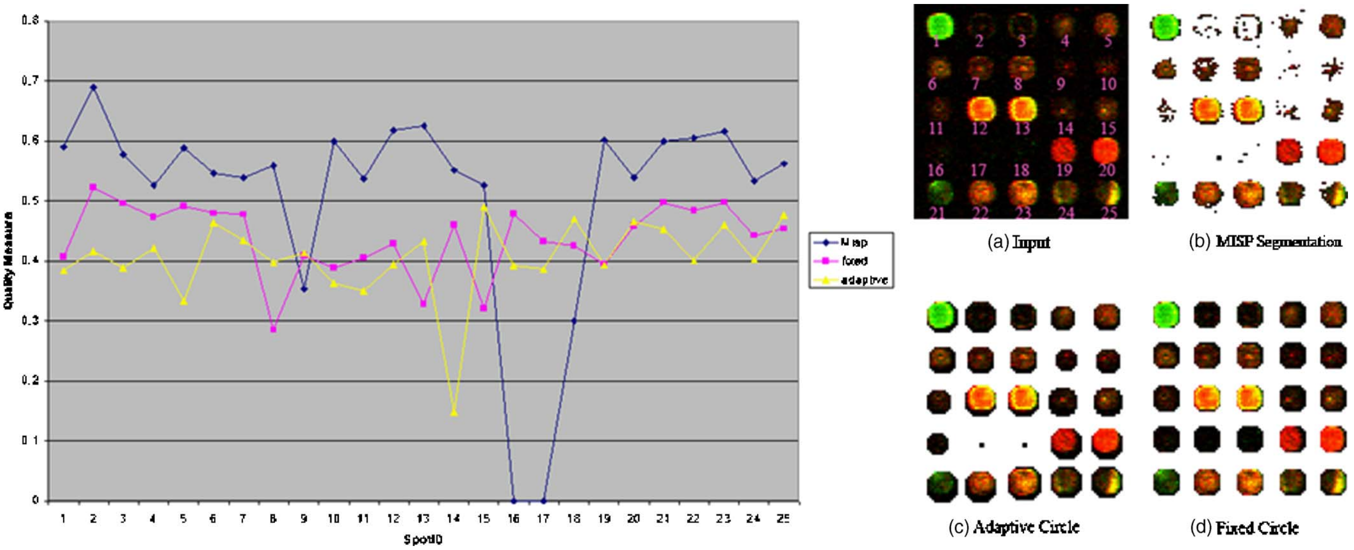
Figures 16–20 show the input microarray and the results obtained using different methods of segmentation. For each



(a) Input (b) MISP Segmentation

(c) Adaptive Circle (d) Fixed Circle

Fig. 18 Results obtained using different methods of segmentation on the microarray M_9. The plot reports the corresponding q_{index} relative to different segmentation methods. (c) and (d) are obtained using *Scanalyze*. Our solution (b) is able to outperform in almost all cases.



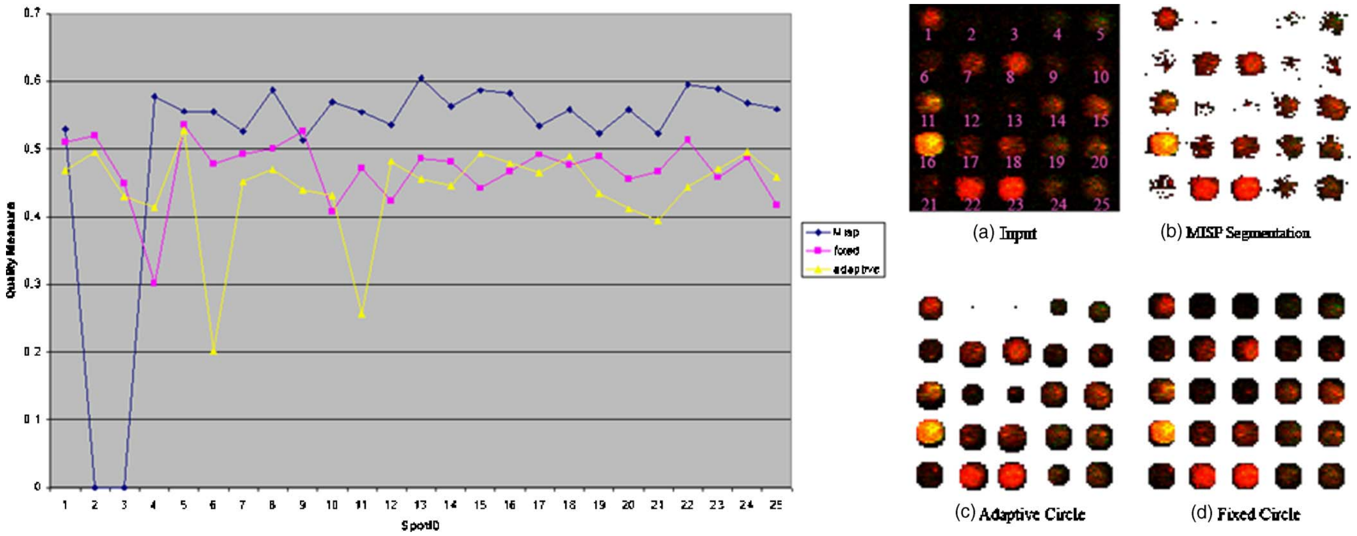
(a) Input (b) MISP Segmentation

(c) Adaptive Circle (d) Fixed Circle

Fig. 19 Results obtained using different methods of segmentation on the microarray M_{10}. The plot reports the corresponding q_{index} relative to different segmentation methods. In (a) the input microarray is reported. In (b) the segmentation obtained by MISP is shown. (c) and (d) are obtained using *Scanalyze*.

spot, the plot of the corresponding q_{index} relative to different segmentation methods is also reported. The results in Fig. 16(c) and 16(d) are obtained using *Scanalyze*. Remember that *Scanalyze* allows the user to manually fix the dimension and the center of the circle enclosing each spot; in our experiment, we refer to this technique as "adaptive circle" [Fig. 16(c)]. The default *Scanalyze* segmentation is denoted as a "fixed circle" [Fig. 16(d)]. Our solution is able to outperform in almost all cases this *ad hoc* heuristic as well as shown by visual and numerical results of q_{index}, which confirm the effectiveness of MISP with respect to *Scanalyze* for fixed and adaptive cases.

To confirm the superiority of our segmentation strategy, we compare the extracted data using a scatterplot method. To evaluate the segmentation strategy, we take into account the data using MISP and *Scanalyze*. Scatterplots have a specific purpose since they show how much one variable is affected by another. The relationship between two variables is denoted the *correlation*. Scatterplots usually consist of a large body of data, and the closer the data points come when plotted to making a straight line, the higher the correlation between the two variables, or the stronger the relationship. If the data points make a straight line going from the origin out to high x- and y-values, then the variables are



(a) Input (b) MISP Segmentation

(c) Adaptive Circle (d) Fixed Circle

Fig. 20 Results obtained using different methods of segmentation on the microarray M_{11}. The plot reports the corresponding q_{index} relative to different segmentation methods. In (a) the input microarray is reported. (c) and (d) are obtained using *Scanalyze*. Visual and numerical results of q_{index} confirm the performance of MISP (b) with respect to *Scanalyze* for both fixed and adaptive cases.

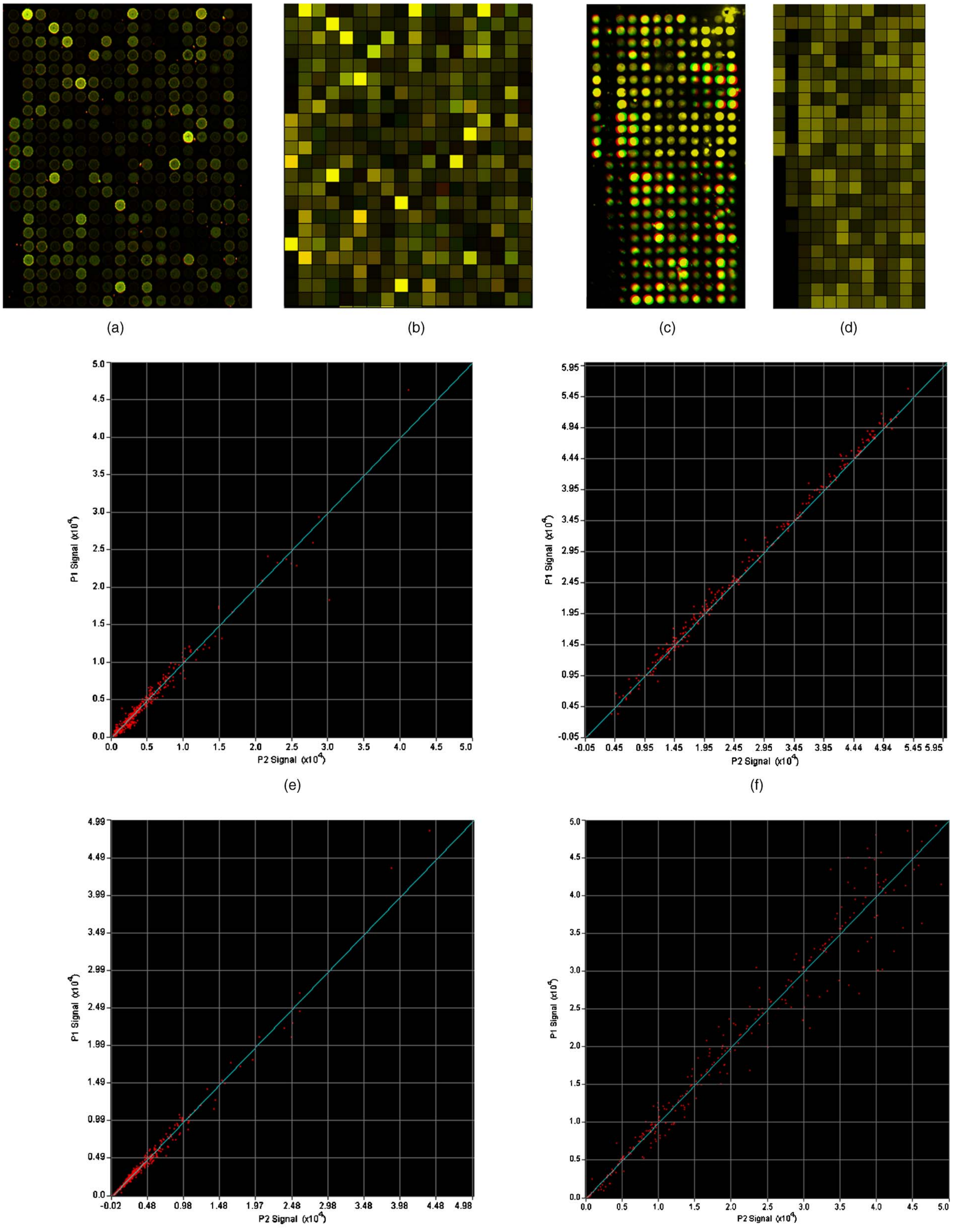


Fig. 21 Scatterplot results. (a) The microarray M_{12}. (b) The corresponding virtual image generated by using AVA.[36,37] (e), (g) The scatterplot obtained with MISP and *Scanalyze*, respectively. (c), (d) The microarray M_{13} and its virtual representation. (f), (h) MISP and *Scanalyze* scatterplots, respectively. MISP performs better than *Scanalyze* because the ratio values are clearly more homogenous.

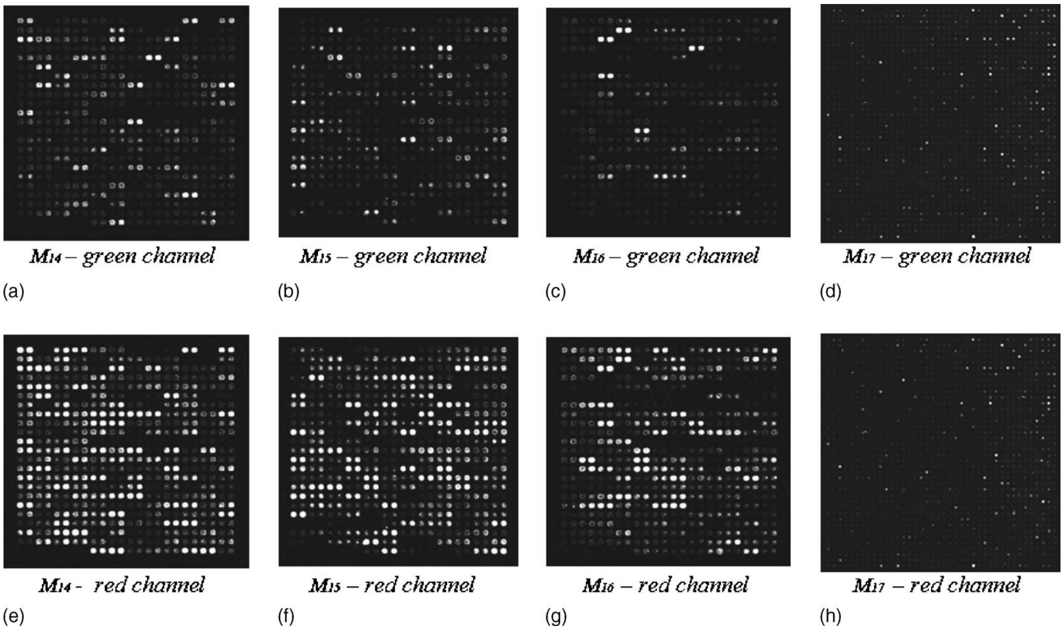


Fig. 22 Microarrays involved in the gridding test.

said to have a positive correlation. A perfect positive correlation is given to the value of 1. The rationale of this test is the following: Knowing the precise red and green channel values for each spot and, therefore, the ratio, supposing to have highly correlated data by means of a scatterplot, it is possible to easily evaluate and compare the performances of the segmentation phase (the final correlation should be the same). The relative effectiveness of the various approaches is evident without considering (displaying) the original scatterplot, because "good" segmented data must lie very close to the main diagonal.

In Fig. 21, two microarrays used for scatterplot tests are shown. The microarray M_{12} was obtained by selecting from microarrays in the Incyte data set[30] a grid in which about 85% of spots have a red/green ratio close to 1 (ratio range, 0.99–1.25).

M_{12} is part of a microarray set used for scanner calibration. In particular, it presents high-quality spots approximately circular, and there is no evident problem with contamination of image acquisition. In this case, fixed-circle and adaptive techniques show very similar performances (no evident background information is collected by the involved segmentation pipelines).

The microarray M_{13} is obtained by duplicating a single-channel image and applying in the copy a small random variation for each spot center location, to simulate typical acquisition problems (the range is ±2 pixels in both the horizontal and vertical axes). Since there are no changes in pixel value, the ratio for each spot is equal to 1. The original single-channel image shows some typical problems with spot size and shape, grid rotation, and nonhomogeneous background. Scatterplots of M_{13} show the superiority of adaptive techniques. Note that, without the introduced alterations, the ratio would have been equal to 1 in the fixed-circle case, although intensity extraction would have been erroneous due to spot shape problems.

Data from each microarray channel and the red/green ratio have been extracted using both MISP and *Scanalyze*. The segmentations have been obtained by considering MISP results where the relative q_{index} (see Section 6) has been measured to guarantee a high-quality value. *Scanalyze* results have been obtained by positioning (manually) the relative circle for each spot in order to capture just the real signal. The data have been normalized using the mean value of each channel and the mean of the red/green ratio.

The scatterplots have been obtained using *AVA—Array Visual Analyzer*,[36,37] a tool to analyze microarray data. The scatterplots in Fig. 21 show that MISP performs better than *Scanalyze* because the ratio values are clearly more homogenous.

7.3 *Gridding Tests*

The input microarray used to test this step are shown in Fig. 22. M_{14} and M_{15} have good-quality spots in a 24×23 grid, with some absent spots. In M_{16} the spots are arranged in a 24×23 grid, but more of them are missing or are irregular spots affected by noise. A 40×40 grid characterizes M_{17} with even more absent and low-signal spots.

Test results for each microarray are reported in Table 3, where each column shows one of the following values:

1. number of real printed spots;
2. number of spots effectively present;
3. number of present spots correctly localized;
4. missing spots detected;
5. number of present spots with low signal;
6. spurious signals amiss belonging to the grid.

Figure 23(a) shows the microarray M_{18} at the first grid of ExpID 15739[34] and the corresponding gridding results obtained by SGRIP [Fig. 23(b)]. For comparison, we report the results of the approach used in Ref. 12 [Fig. 23(c)], where a comparison with the Spot[6] gridding is made [Fig. 23(d)]. We point out that in Ref. 12 [Fig. 23(c) and 23(d)]

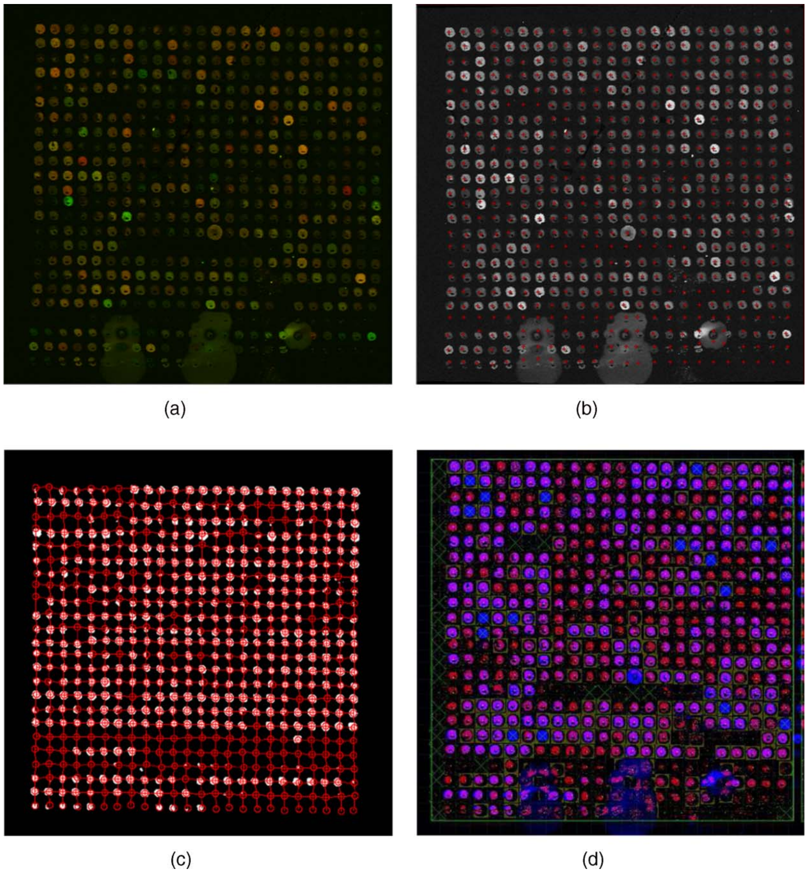


Fig. 23 (a) The microarray M^{18} of the first grid of ExpID 15739.[31] (b) Corresponding gridding results obtained by SGRIP. (c), (d) Results of the approach used in Refs. 12 and 6, respectively.

the generated grid is overlapped to the segmented image, while in 23(b) the grid is overlapped to the input image, corrected from rotation.

The results confirm the effectiveness of the proposed pipeline to manage real cases including missing or partially missing spots. Finally, the proposed strategy is also able to determine the number of missing spots producing the inferred position.

8 Conclusions

In this work, we have proposed an integrated framework called MIAF for microarray image analysis that improves upon previous solutions. Adaptive techniques have been applied for both gridding and segmentation, obtaining effective and reliable information about input data even in the presence of noise and irregular spot distribution. The overall process is fully unsupervised.

We have also introduced a modification of the quality index proposed in Ref. 4 to take into account capabilities of adaptive spot segmentation modules. Experimental results show that the proposed pipeline captures in a reliable way the underlying signal distribution of input data. The proposed solution for gridding and rotation detection further improves the MISP[13] producing a set of binary masks used to derive accurate spot information and quality measures. Future works will include the possibility of using *ad hoc* techniques for common acquisition problems (e.g., noise reduction[38]). Major details and experiments can be found at http//www.dmi.unict.it/~iplab.

Table 3 Experimental gridding results on data set M_{14}, M_{15}, M_{16}, and M_{17}.

Microarray	1	2	3	4	5	6
M_{14}	552	515	515	37	21	2
M_{15}	552	482	482	70	44	0
M_{16}	552	467	467	85	30	0
M_{17}	1600	1329	1329	271	600	0

References

1. NCBI, "Microarray: Chipping away at the misteries of science and medicine," http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html (2004).
2. D. Stekel, *Microarray Bioinformatics*, Cambridge University Press, New York (2003).
3. Y. H. Yang, M. J. Buckley, and T. P. Speed, "Analysis of cDNA microarray images," *Briefings Bioinf.* **2**(4), 341–349 (2001).
4. U. Sauer, C. Preininger, and S. R. Hany, "Quick & simple: quality control of microarray data," *Bioinformatics*, *Advance Access* (2004).
5. K. Groch, A. Kuklin, A. Petrov, and S. Shams, "Image segmentation and quality control measures in microarray image analysis," *J. Assoc. Lab. Autom.* **6**(3), 73–76 (2001).
6. Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," *J. Comput. Graph. Stat.* **11**(1), 108–136 (2002).

7. J. Angulo and J. Serra, "Automatic analysis of DNA microarray images using mathematical morphology," *Bioinformatics* **19**(5), 553–562 (2003).
8. Axon Instruments Inc., *GenePix Pro User's Guide*, http://www.axon.com/, Software and Documentation (2001).
9. M. Eisen, *Scanalyze*—Software and Documentation, http://rana.lbl.gov/EisenSoftware.htm(1999).
10. L. Heyer, D. Z. Moskowitz, J. A. Abele, P. Karnik, D. Choi, A. M. Campbell, E. E. Oldham, and B. K. Akin, "MAGIC tool: Integrated microarray data analysis," *Bioinfor. Appl. Note* **21**(9), 2114–2115 (2005).
11. G. Antoniol, M. Ceccarelli, and A. Petrosino, "Microarray image addressing based on the radon transform," *IEEE Int. Conf. Image Process.* (*ICIP 2005*) 1, 13–16 (2005).
12. G. Antoniol and M. Ceccarelli, "A deformable grid-matching approach for microarray images," *IEEE Trans. Image Process.* **15**(10), 3178–3188 (2006).
13. S. Battiato, G. Di Blasi, G. M. Farinella, G. Gallo, and G. C. Guarnera, "*Ad hoc* segmentation pipeline for microarray image analysis," *Proc. in IS&T-SPIE Electronic Imaging*, San Jose, CA (2006).
14. Q. Li, C. Fraley, R. E. Bumgarner, K. Y. Yeung, and A. E. Raftery, "Donuts, scratches and blanks: robust model-based segmentation of microarray images," Technical Report no. 473, Department of Statistics, University of Washington(2005).
15. A. Kuklin, "Laboratory automation in microarray image processing," *Am. Lab. (Shelton, Conn.)*, 64–67 (2000).
16. R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(6), 641–647 (1994).
17. X. Wang, S. Ghosh, and S. W. Guo, "Quantitative quality control in microarray image processing and data acquisition," *Nucleid Acids Res.* **29** (2001).
18. S. Lonardi and Y. Luo, "Gridding and compression of microarray images," *IEEE Comput. Syst. Bioinfor. Conf. (CSB'04)*, Stanford, CA (2004).
19. Y. Wang, F. Y. Shih, and M. Q. Ma, "Precise gridding of microarray images by detecting and correcting rotations in subarrays," in *Proc. 8th Joint Conf. on Infor. Sci.*, Salt Lake City, pp. 1195–1198 (2005).
20. J. Ho, W. L. Hwang, H. H. Lu, and D. T. Lee, "Gridding spot centers of smoothly distorted microarray images," *IEEE Trans. Image Process.* **15**(2), 342–353 (2006).
21. K. Blekas, N. P. Galatsanos, A. Likas, and I. E. Lagaris, "Mixture model analysis of DNA microarray images," *IEEE Trans. Med. Imaging* **24**(7), 901–909 (2005).
22. J.-S. Kim, "A fast feature-based block matching algorithm using integral projections," *IEEE J. Sel. Areas Commun.* **10**(5), 968–971 (1992).
23. A. Jain, M. N. Murthy, and P. J. Flynn, "Data clustering: A review," *ACM Comp. Rev.* (1999).
24. R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(11), 1452–1458 (2004).
25. R. C. Gonzales and R. E. Woods, *Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ (2002).
26. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York (2006).
27. R. Nagarajan, "Intensity based segmentation of microarray images," *IEEE Trans. Med. Imaging* **22**(7), 882–889 (2003).
28. A. C. Johnson and N. T. Thomopoulos, "Characteristics and tables of the doubley-truncated normal distribution," in *Proc. Production & Oper. Manage. Soc. (POMS) High Tech.* (2002).
29. http://www.bio.davidson.edu/projects/magic/magic.html.
30. http://dc.nci.nih.gov/dataSets/geawQCandIA.
31. J. L. DeRisi, V. R. Iyer, and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science* **278**(5338), 680–686 (1997).
32. http://brownlab.stanford.edu/.
33. J. Gollub, "The stanford microarray database: Data access and quality assessment tools," *Nucleic Acids Res.* **31**, 94–96 (2003).
34. M. Arbeitman, E. Furlong, F. Imam, E. Johnson, B. Null, B. Baker, M. Krasnow, M. Scott, R. Davis, and K. White, "Gene expression during the life cycle of drosophila melanogaster," *Science* **2957**, 2270–2275 (2002).
35. M. Bredel, C. Bredel, D. Juric, G. R. Harsh, H. Vogel, L. D. Recht, and B. I. Sikic, "High-resolution genome-wide mapping of genetic alterations in human glial brain tumors," *Cancer Res.* **65**(10), 4088–4096 (2005).
36. Y. Zhou and J. Liu, "AVA: Visual analysis of gene expression microarray data," *Bioinformatics* **19**(2), 293–294 (2003).
37. Y. Zhou, *Array Visual Analyzer (AVA) Manual* (2002).
38. A. Bosco, M. Mancuso, S. Battiato, and G. Spampinato, "Temporal noise reduction of Bayer matrixed video data," in *Proc. IEEE ICME02*, Switzerland, pp. 681–684 (2002).

Sebastiano Battiato received his degree in computer science (summa cum laude) in 1995 and his PhD degree in computer science and applied mathematics in 1999 at the University of Catania. From 1999 to 2003 he led the "Imaging" team at STMicroelectronics in Catania. Since 2004 he has worked as a Researcher in the Department of Mathematics and Computer Science of the University of Catania. His research interests include image enhancement and processing and image coding. He has published more than 80 papers in international journals and conference proceedings. He is coinventor of about 15 international patents. He is reviewer for several international journals and has participated in many international and national research projects. He is a senior member of the IEEE.

Gianpiero Di Blasi received his degree in math in 1999 at the University of Palermo and his PhD degree in computer science in 2006 at the University of Catania. He is currently a Post Doc in computer science at the University of Calabria. His research interests include image analysis, processing, and enhancement, computer graphics algorithms, nonphotorealistic rendering techniques, SVG applications, and Java/Java3D programming.

Giovanni Maria Farinella received his degree in computer science (summa cum laude) from Catania University, Italy, in 2004. He is currently a PhD student in computer science and an internal member of the IPLAB research group at the same university. His interests lie in computer vision and pattern recognition, application in biomedical informatics, biometric systems, and image/video processing and recognition.

Giovanni Gallo received his doctorate degree in mathematics from Catania University, Catania, Italy in 1990, and his PhD degree in computer science from New York University in 1992. He has taught formal methods for computer science and computer–human interaction at Catania University since 1992. His research interests include computer algebra, image processing, image database, medical image processing, and scientific visualization.

Giuseppe Claudio Guarnera received his BS degree in computer science with honors (summa cum laude) from the University of Catania, Italy, in 2006. He is currently an MS degree student in computer science. His research interests include image processing and statistics.