

# Ad-Hoc Segmentation Pipeline for Microarray Image Analysis

S. Battiato, G. Di Blasi, G. M. Farinella, G. Gallo, G. C. Guarnera  
{battiato, gdibiasi, gfarinella, gallo}@dmi.unict.it  
g.guarnera@studenti.unict.it

IPLab – Image Processing Laboratory  
<http://www.dmi.unict.it/~iplab>  
Dipartimento di Matematica e Informatica  
University of Catania, Via Andrea Doria 6 – 95125, Catania (Italy)

## ABSTRACT

Microarray is a new class of biotechnologies able to help biologist researches to extrapolate new knowledge from biological experiments. Image Analysis is devoted to extrapolate, process and visualize image information. For this reason it has found application also in Microarray, where it is a crucial step of this technology (e.g. segmentation). In this paper we describe MISP (Microarray Image Segmentation Pipeline), a new segmentation pipeline for Microarray Image Analysis. The pipeline uses a recent segmentation algorithm based on statistical analysis coupled with K-Means algorithm. The *Spot* masks produced by MISP are used to determinate spots information and quality measures. A software prototype system has been developed; it includes visualization, segmentation, information and quality measure extraction. Experiments show the effectiveness of the proposed pipeline both in terms of visual accuracy and measured quality values. Comparisons with existing solutions (e.g. Scanalyze [1]) confirm the improvement with respect to previously published works.

**Keywords:** Image Analysis, Microarray, Image Segmentation, Bioinformatics

## 1. INTRODUCTION

Image analysis has found application in Microarray technology, because it is able to extrapolate new and not trivial knowledge often hidden in the images. Microarray is a small solid support on which sequences of DNA of hundreds or thousands of different genes are spotted in fixed positions; the order of positioning is useful to the biomedical researcher to identify a specific genes sequence. Microarray technology has changed the operating way of the biologists, allowing the scientific observation of a great number of genes under several conditions simultaneously, in a single experiment. The Microarray technology consists of different sequential phases: biomedical questions, experimental design, microarray experiment, image and data analysis, biological verification. In each experiment (e.g. comparison between two tissue samples), two 16-bit TIFF images are obtained by using Microarray scanners capturing the fluorescence (Cy3, 510-550 nm, *Green* and Cy5, 630-660 nm, *Red*). The corresponding intensity images is proportional to the observed fluorescence. For sake of visualization, the images are combined together to derive a single 24-bit RGB image in which blue channel is set to zero while R-G image compression is usually used.

For each spot, the relative intensity of the channels *Red* and *Green* is determined [4]:

- prevalence of the red intensity denotes greater expression in the mutant sample;
- prevalence of the green intensity denotes greater expression in the reference sample;
- yellow denote a parity of expression.

Image analysis is a crucial aspect of Microarray experiments. It has a potentially large impact on subsequent analysis such as clustering or identification of differentially expressed genes [5]. In the Microarray technology context, images are usually processed by the following steps [6]:

- Addressing or Gridding: spot coordinates assignment;
- Segmentation: pixel classification in terms of foreground, signal of interest, and background;
- Intensity extraction and quality measures: spot evaluation of red and green foreground/background intensity pairs and quality measures [7], [8], [9].

Many academic ([1], [2], [3], [10]) and commercial [11] Microarray image analysis software are available (see [5] for a useful comparison).

Microarray image segmentation is fundamental to derive reliable information. Robust segmentation is important for correct classification of genes' expression and to extrapolate a variety of spot quality measures.

In this paper we propose a new advanced segmentation pipeline called MISIP (Microarray Image Segmentation Pipeline). The method is mainly based on a recent statistical segmentation technique [12] able to automatically detect regions containing connected and similar intensity pixels.

The overall MISIP process produces the following binary masks: SGM (Spots Guide Mask), RMF (Red Mask Foreground), GMF (Green Mask Foreground), RMLB (Red Mask Local Background), GMLB (Green Mask Local Background) and GGM (Grid Guide Mask).

SGM is used to derive quality measure for each spot (e.g. spot area measure). GGM is used to assign the effective coordinates to each spot. Other masks are used to characterize pixel belonging to foreground/background/local background, to calculate intensity and to extrapolate quality measures. Experiments confirm the effectiveness of the proposed pipeline, outperforming state-of-art methods.

The rest of the paper is organized as follows: Section 2 describes the Microarray image analysis and related works. In Section 3 we discuss our proposed ad-hoc MISIP pipeline, while implementation details and experimental results are presented in Section 4. Finally, conclusion and future works are summarized in Section 5.

## 2. MICROARRAY IMAGE ANALYSIS

Image analysis is the first Microarray technology processing step; it has a strong impact in the successive phases of analysis (clustering and identification of differently expressed genes). The input is composed by two 16-bit TIFF images. Usually, input images are properly scaled by reducing the overall dynamic range to 8-bit obtaining a single RGB image (for sake of visualization). The reduction is obtained by a square root transformation ( $\sqrt{2^{16}} = 2^8$ ).

Alternatively it is possible to select only 8 of the overall 16 bits. In Genepix [11], for example, it is possible to manually select 8 bits or to use the predefined options (high, low or centre bits). In the final 24-bit RGB image the blue channel is set to zero, red and green values came respectively from the 630-660 nm and 510-550 nm Microarray scanning. Microarray images analysis can be subdivided in three tasks ([5], [6]): addressing or gridding, segmentation, intensity and quality measure extraction.

### 2.1 Addressing or gridding

Addressing process assigns the coordinates to each Microarray spot. Such phase is carried out manually or automatically. Automatic addressing increases the speed of the analysis, but only a few softwares offer this option. Usually, only the manual addressing is supported by requiring the user interaction to choice among related parameters (e.g. grid size and orientation).

### 2.2 Segmentation

Segmentation classifies pixels in foreground (spot signal) or background. Automatic segmentation performances are strongly affected by defects due to the "source" (e.g. wrong concentrations of DNA) or acquisition system problems (e.g. scanners amplification). Existing methods for Microarray spots segmentation can be classified in four groups [5]:

- Fixed circle: all image spots are enclosed in circles of constant diameter;
- Adaptive circle: circle diameter is estimated separately for each spot;
- Adaptive shape: no restriction on the spots shape;
- Histogram: shape is based on statistical signal distribution without using spatial information.

Histogram segmentation uses a "target" mask (a region larger than any spot) and estimates foreground/background intensity for each spot from the pixel values histogram inside the mask. The approaches based on fixed or adaptive circular spot mask fail in capturing signal information; better result are obtained using adaptive shape detector, because the circular spot signals are quite rare. Two methods used for adaptive shape segmentation are based on watershed [13] and seeded region growing [14]. Both methods require the specification of starting pixels (seeds). Automatic seed specification, especially for spot segmentation, usually provides wrong results.

### 2.3 Intensity extraction and quality measures

After addressing and segmentation, intensities and quality measures have to be extracted. Usually, information extracted are spot foreground intensity (e.g. mean and/or median of pixel values within the segmented spot mask) and local background intensity (e.g. mean of pixel values inside the spot mask not belonging to the foreground) [5]. Typically, the target intensity is defined as the difference between foreground and background intensity [15]. Various quality measures have been proposed in literature ([7], [8], [9]). In the rest of the paper we use the *combined quality index* proposed in [7]. This quality measure index is robust and accurate. Combined quality index ( $q_{com}$ ) encloses size of the spot ( $q_{size}$ ), signal-to-noise-ratio ( $q_{sig-noise}$ ), 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 local background and  $q_{sat}$  indicates if the percentage of saturated pixel is less than 10% for each spot. All quality measures fall in the range [0, 1]. The measures are calculated for each dye channel as follows ([7], [8]):

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

$$q_{size} = \exp\left(-\frac{|A - A_0|}{A_0}\right) \quad (2)$$

where  $A$  is the number of pixel for each spot (spot area) and  $A_0$  is the average number of pixel for all spots (average spot area);

$$q_{sig-noise} = \frac{Fmean}{Fmean + Bmean} \quad (3)$$

where  $Fmean$  is the mean of foreground pixel intensity for each spot and  $Bmean$  is the mean of local background pixel intensity for each spot;

$$q_{bkg1} = \frac{f_1}{CV_{bkg}} \quad f_1 = \frac{1}{\text{Max}\left[\frac{BSD}{Bmean}\right]} \quad CV_{bkg} = \frac{BSD}{Bmean} \quad (4)$$

where  $CV_{bkg}$  is the variation coefficient of the local background for each spot,  $BSD$  is standard deviation of local background for each spot, and  $f_1$  is a normalization constant that satisfies  $\text{max}(q_{bkg1})=1$ ;

$$q_{bkg2} = f_2 \times \left(\frac{bgk_0}{bgk_0 + Bmean}\right) \quad f_2 = \frac{1}{\text{Max}\left[\frac{bgk_0}{bgk_0 + Bmean}\right]} \quad (5)$$

where  $bgk_0$  is the global average of background;

$$q_{sat} = \begin{cases} 1 & \text{if saturated spot pixels} < 10\% \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$q_{sat}$  threshold is defined by using the mean of total pixel intensity values. A pixel is saturated if its grey value is greater than the average of total pixel intensity values.

## 2.4. Related works

This section describes a few image analysis software recently proposed in literature: Scanalyze [1], GenePix [11], Spot [5], Angulo-Serra's system [9], Matarray [7] and Magic [2]. Primary goal of almost all software is properly process the input data extracting information and quality measures for each spot in Microarray.

Scanalyze [1] processes each input *TIFF* images obtaining a single RGB image as follows:

$$R = \frac{Gain \times \left( \frac{Cy5\_pixel}{256} \right)}{Normalization} \quad G = Gain \times \left( \frac{Cy3\_pixel}{256} \right) \quad B = 0 \quad (7)$$

Brightness is controlled by *Gain* parameter while *Normalization* is a balancing parameter. Scanalyze addressing phase is manual. Various parameters for grid construction have to be specified (e.g. separation between rows and columns). Scanalyze uses the fixed circle segmentation method. Size dimension of circles is manual and occurs contextually in addressing phase. All spots are considered valid even if they do not enclose signal; such spots can be excluded manually from the information extraction phase. To estimate background local intensity the median values are used. The ratio G/R of the average foreground is calculated possibly taking into account the correction of the background. The software produces for each spot a set of information and several quality values (e.g. channels correlation).

*Genepix* [11] uses a square root transformation to reduce the dynamic range of input images. It is also possible to select manually the 8 bits to be saved, or use the predefined options to save high, low or central bits. Addressing is automatic. In the previous version, circle based segmentation with variable dimension was used, and the spot was classified as "not found" according to some conditions (e.g. "spot diameter less then 6 pixel", "spot position overlap another spot" or "spot diameter is outside the planned options limits"). In the last version this segmentation method has been replaced by irregular spot segmentation. Local and global methods are used to compute background and foreground intensities for each spot. Spot circularity is also used as measure of quality.

*Spot* [5] uses linear combination weighted by the median values to obtain an single image  $I = G' + (mG/mR) * R'$  where  $G'$  and  $R'$  are the images obtained from  $G$  and  $R$  using squared root transformation and  $mG$  and  $mR$  are the corresponding median values. Spot addressing is based on a batch processing over a collection of Microarray images with the same geometric structure. Successive steps are automatic and produce two estimated grids:

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

In the segmentation phase a seeded region growing [14] algorithm is used. The seeds are chosen according to the estimated grids. Background, foreground intensity and quality measures are computed similarly to GenePix.

Angulo and Serrà [9] combine input images using linear combination weighted by median [5]. The overall pipeline combines addressing and gridding techniques making use of morphological operators together with classical segmentation algorithms [13]; the overall performances are evaluated in terms of segmentation accuracy without providing quality measures.

*Matarray* [7] 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 centres, the overall area is splitted in patches defining a circular mask for each patch used for spot segmentation. An iterative process is then achieved calculating signal intensity and local background to improve the detection. The combined quality index (1) is used for quality assessment.

*MicroArray Genome Imaging and Clustering Tool (MAGIC)* [2], [3] analyze all types of gene expression data on all major operating systems. Visualization is performed by linear combination weighted. Gridding does not require grid and feature dimensions or spacing. Segmentation is performed with one of three algorithms: fixed circle, adaptive circle or seeded region growing. The fixed circle is centered in the grid square, with a user specified radius. The adaptive circle algorithm examines the signal in each grid square to determine the most appropriate center and radius (within a user-specified range) for each circle. Finally, the adaptive circle's center are set to be those containing the largest number of 'on' pixels. A seeded region growing algorithm connects each pixel to a background or foreground region, continuing until all pixels are assigned. A user-specified threshold and geometric considerations determine which pixels are used to 'seed' the regions. The user can choose if consider background in computation of green/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 not flagged spot) for each channel and channels ratio intensity.

### 3. MISP: MICROARRAY IMAGE SEGMENTATION PIPELINE

An image segmentation is a partitioning technique which subdivides an image into a set of non overlapping regions conveying some specified features and whose union is the entire image. In the Microarray technology context the segmentation decompose the image (obtained by scanning) into regions that are meaningful in terms of spot signal and/or background. Correct segmentation is critical for an accurate signal pixel classification of each spot; a good quality segmentation allows using several quality measures based on both geometric properties and signal features of the spot, such as *Spot Signal Area to Spot Area Ratio* [9]. Different methods have been proposed for microarray images segmentation. In [17] an adaptive shape segmentation method based on K-means clustering algorithm is proposed. In [15] intensity-based segmentation techniques are used to generate a binary pixel partition inside a grid; two clustering techniques (Partitioning Around Medoids, PAM, and K-means) are used, while in [5] a seeded region growing algorithm is used. Spots circular mask approaches (fixed or adaptive circle segmentation) are not able to identify a well spot signal shape. Adaptive shape segmentations, are better than the previous ones but their performances are strictly related with specification (manually or not) of the number and position of initial seeds. In this section we describe the main details of a new advanced image segmentation pipeline called MISP. The overall process is synthetically sketched in Figure 1. MISP is an adaptive technique and does not require initial seeds. The overall process is automatic and the parameters setting is simple. MISP process produces the following binary masks: SGM (Spots Guide Mask), RMF (Red Mask Foreground), GMF (Green Mask Foreground), RMLB (Red Mask Local Background), GMLB (Green Mask Local Background) and GGM (Grid Guide Mask).

#### 3.1 Algorithms and operators

In this section we describe the algorithms and operators used in MISP. Next subsection will show how to merge all together in order to obtain final reliable results.

##### 3.1.1 Statistical region merging algorithm

Statistical Region Merging [12] is a recent segmentation technique able to capture the main structural components of a digital image using a simple but effective statistical analysis. According to theoretical analysis described in [12], [16], in Microarray imaging it is possible to estimate the distribution of ideal spot signal to be segmented by using an ad-hoc regions growing technique. Statistical Region Merging (SRM) is based on two components:

1. *Merging Predicate*, used to establish if two regions have to be merged;
2. *Merging Order*, used to establish the order used to test the *Merging Predicate*;

The pseudo-code of the algorithm is reported below (see [12], [16] for major details):

1	<i>Input: an image I (a microarray image channel)</i>
2	Let $S_I$ be the set of the 4-connexity couples of adjacent pixels in image $I$ .
3	$S'_I = \text{Order\_increasing}(S_I, f)$ ;
4	For $i=1$ to $ S'_I $ do
5	If $R(p_i) \neq R(p'_i)$ and $P(R(p_i), R(p'_i)) = \text{true}$ then
6	Union( $R(p_i), R(p'_i)$ )

$f(p, p')$  is a real-valued function, with  $p$  and  $p'$  adjacent (4-connexity) pixels in  $I$ .  $f$  is used to establish the *Merging Order*. In particular  $f$  approximates the following invariant: when any test between two true regions occurs, that means that all tests inside each of the two true regions have previously occurred. Different choices can be done both for *Merging Factor*  $Q$  used by *Merging Predicate* and  $f$  function ([12], [16]). In our experiments we heuristically choose  $Q=1024$  and  $f(p, p') = |p - p'|$  where  $p$  and  $p'$  are pixel channel values. The original algorithm has been extended to work with 16-bit images setting the *Nlevels* parameter to 65536.

##### 3.1.2 K-Means algorithm

K-means [18], [19] is a partition clustering algorithm; it starts with an initial partition and assigns patterns to clusters to reduce the overall squared-error. Partitions are updated iteratively by reassigning patterns minimizing the global error.

K-means algorithm can be generalized as follows:

- 1 *Input: Set of patterns;*
- 2 Select an initial partition with  $k$ -clusters;
- 3 *Repeat the steps 4 through 7 until the cluster membership stabilizes;*
- 4 Generate a new partition by assigning each pattern to its closest cluster center;
- 5 Compute new cluster centers as the centroids of the cluster;
- 6 *Repeat step 4 and 5 until an optimum value of the criterion function is found;*
- 7 Adjust the number of clusters by merging and splitting existing clusters or by removing small, or outlier clusters;

An initial partition can be created by first specifying a set of  $K$  seed points. Seed points can be randomly chosen from all patterns. Partitions are updated by reassigning pattern to clusters. A K-means step is the assignment of all patterns to the closest cluster center. The center of the obtained clusters is recomputed after each new assignment, by through the mean values of the clusters. Step 7 is usually used to recover from poor initial partitions selecting a “suitable” number of clusters. A cluster is splitted if it has too many patterns with large variance value along the feature with largest spread. Two clusters are merged if their centers are sufficiently close. Several strategies and heuristics can be applied to execute the above steps algorithm ([17], [18], [19]).

### 3.1.3 Other operators

The overall process requires a series of classical punctual and global operators. In particular our segmentation pipeline makes use of *Gamma LuT*, *Mean*, *Thresholding* [20]. Also some simple set binary operators (e.g. union - OR, intersection - AND, etc.) are used.

## 3.2 Proposed pipeline

Figure 1 shows the data-flow processing starting from original microrarray images to the final segmented output, directly managed by data and quality measures extraction phases (Figure 5). Our technique processes each Microarray image to produce five semantic regions as shown in Figure 4: Background (*black*), Local Background (*blue*), Red Channel Foreground (*red*), Green Channel Foreground (*green*), Red and Green Channel Foreground (*yellow*). For each microarray spot the total Red Channel Foreground (Green Channel Foreground) can be obtained using the union operator on the pixels set identified by the regions *red* and *yellow* (*green* and *yellow*), while Total Background can be obtained using the union operator on the pixels set identified by *black* and *blue* regions. The pipeline can be ideally subdivided into two sequential blocks (Figure 1):

- Spot-Background separation (Figure 2)
- Foreground and Local Background identification (Figure 3)

### 3.2.1 Spot – Background separation

This block separates the spot pixels from the background. The first step is performed by using SRM for each microarray channel. For each region  $i$  obtained by SRM  $RegCh_{color,i}$  the average  $MeanRegCh_{color,i}$  is calculated. The value  $MeanRegCh_{color,i}$  is assigned to every pixel in region  $RegCh_{color,i}$ . The two steps above allow obtaining a more homogenous image: we can separate spot/background using regions mean intensity rather than pixels value intensity. The next step uses a *Gamma Lut* ( $\gamma=0,3$ ) to better distinguish the spot pixels from the background pixels. The images obtained are then processed using K-means algorithm ( $K=2$ ) in order to obtain two binary image channels (*RedBin*, *GreenBin*). The final step recovers spot pixels not captured previously (e.g. edges spot signal); the mean value (relative to the pixels classified as spot pixel in the previous step) is calculated for each channel. This threshold is used to establish which pixels can be recovered and considered as spot pixels. A binary mask for each channel is obtained using threshold: *Red RecBin* and *Green RecBin*. These masks are combined together using a union set operation with the masks *RedBin* and *GreenBin* respectively. The final output of this block are the masks *GBin* and *RBin* that contain pixels belonging to the spot and pixels belonging to the background (Figure 2).

### 3.2.2 Foreground and local background identification

This block produces a *Spot Guide Mask* and a *Foreground/Local Background Mask* (for each channel). *Spot Guide Mask (SGM)* is obtained using the union operator (*OR*) on the set that identifies pixels spot in *RBin* and *GBin*. *SGM* (pixels belonging to a spot in image obtained by microarray channels overlapping) characterizes which pixel in the combined channels image is a spot pixel (e.g. it contains a signal value).

Set relations between pixels in the sets induced by *RBin*, *GBin* and *SGM* are the following:

$$RGSpotSet = RSpotSet \cup GSpotSet \quad (8)$$

$$RSpotSet \subseteq RGSpotSet \quad (9)$$

$$GSpotSet \subseteq RGSpotSet \quad (10)$$

where

$$RSpotSet = \{p_{i,j} | RBin(i,j)=1 \quad \forall i = 1, \dots, rows \quad \forall j = 1, \dots, columns\} \quad (11)$$

$$GSpotSet = \{p_{i,j} | GBin(i,j)=1 \quad \forall i = 1, \dots, rows \quad \forall j = 1, \dots, columns\} \quad (12)$$

The *Spot\_Guide\_Mask* is defined as follows:

$$Spot\_Guide\_Mask(i,j) = (RBin(i,j)=1) \text{ OR } (GBin(i,j)=1) \quad \forall i = 1, \dots, rows \quad \forall j = 1, \dots, columns \quad (13)$$

*RGSpotSet* can be rewritten as function of *Spot\_Guide\_Mask*:

$$RGSpotSet = \{p_{i,j} | Spot\_Guide\_Mask(i,j)=1 \quad \forall i = 1, \dots, rows \quad \forall j = 1, \dots, columns\} \quad (14)$$

For each channel the set of pixels related to the foreground are identified by intersection of *SGM* with the binary masks *RBin*, *GBin*. Using sets defined above we can write:

$$RFSpotSet = RSpotSet \cap RGSpotSet \quad (15)$$

$$GFSpotSet = GSpotSet \cap RGSpotSet \quad (16)$$

*RFSpotSet* and *GFSpotSet* can also be identified using the *Red\_Mask\_Foreground* and *Green\_Mask\_Foreground* defined as follows:

$$Red\_Mask\_Foreground = (RMF(i,j) = 1) \text{ AND } (Spot\_Guide\_Mask(i,j)=1) \quad \forall i = 1, \dots, rows, \quad \forall j = 1, \dots, columns \quad (17)$$

$$Green\_Mask\_Foreground = (GMF(i,j) = 1) \text{ AND } (Spot\_Guide\_Mask(i,j)=1) \quad \forall i = 1, \dots, rows, \quad \forall j = 1, \dots, columns \quad (18)$$

*Red\_Mask\_Foreground (RMF)* and *Green\_Mask\_Foreground (GMF)* are equals to the *RBin*, *GBin* mask obtained in Spot-Background separation block. The masks are equals but the meaning is different; *RBin* and *GBin* are related to the separation of spot from the background, while *RMF* and *GMF* are related to the pixels spot signal map. We define local background set as the union between the set containing internal background pixels and the set of the pixels belonging to the minimum square that encloses the spot. In order to establish the mask characterizing local background set pixels we use other three masks: *Grid Guide Mask*, *InternalBackground Mask*, *RGBack Mask*. *InternalBackground Mask* is determined for each channel (*RInternalBackground*, *GInternalBackground*), while *Grid Guide Mask* and *RGBack Mask* are equals for both channels. *GGM* is obtained using *SGM* and considering (for each spot) all pixels belonging to the minimum square that is contained in the set induced by spots guide. The pixels present in the set induced by *GGM* and not present in the set induced by *SGM* are the pixels belonging to the set of pixels characterized by *RGBack Mask*. *Internal Background* is determined using the sets difference operator between *SGM* and *GMF* (in green channel case).

The local background pixels set (for each channel) is the union of the set induced by *RGBack* and the set induced by *Red or Green InternalBackground Mask*. Red local background Pixels set is characterized by *Red Mask Local Background (RMLB)*, while Green local background pixels set is characterized by *Green Mask Local Background (GMLB)* showed in Figure 3. *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 pixel belonging to *foreground/background/local background*, to calculate intensity and to extrapolate quality measures for each spot.

## 4. MISP SOFTWARE AND EXPERIMENTAL RESULTS

The software prototype architecture is shown in Figure 5. The software include visualization, segmentation, information and quality measures extraction. Segmentation step is performed using MISP pipeline presented in previous section.

### 4.1 Software description, parameters and heuristics

MISP shows Red and Green channel images, their overlapping and each image generated in each step of the pipeline. The GUI (Figure 8) allows setting of SRM and *Gamma-Lut* parameters. Information about Microarray and quality measures can be extracted and saved in two separated files. For sake of visualization, a copy of image data is compressed from 16 to 8 bits by a square root transformation and its brightness is adjusted by using a *Gamma-Lut* operator ( $\gamma=1.4$ ). For each Microarray image channel SRM is applied to obtain the segmented regions represented using a single colour value (the mean); the regions are brightened using a *Gamma-Lut* ( $\gamma=0.3$ ) to allow the K-means algorithm to extract more details. SRM parameters used in our experiments are  $Q = 1024$  and  $Nlevels = 65536$ .  $Q$  is related to the statistical complexity of the image and  $Nlevels$  is the number of gray levels in microarray images. The  $f$  function used to perform SRM merging order is  $f(p, p') = |p - p'|$  where  $p$  and  $p'$  are pixel channel values. To determine initials K-Means clusters centroids,  $min$  and  $max$  pixel values are found for each microarray spot. In the initial K-Means iteration, it clusters each pixel comparing its value with  $min$  and  $max$ , assigning respectively to the Foreground cluster or to the Background cluster depending on its relative distance. Successive centroids are obtained by calculating the mean values of each cluster of pixels. Next iterations, calculates, for every pixel values from Background cluster, the squared differences from each centroids; the pixel currently tested is assigned to the cluster with the closest centroid value. If the pixel has changed cluster, new centroids needs to be calculated. The same computation is done for foreground cluster pixels. This process is repeated until both clusters membership converge. It may happen that a pixel is alternatively assigned to the clusters: in this case k-Means stops its execution after  $NMAX = 100$  iterations. Foreground and Local Background identification are done as described in section 3.2.2.

### 4.2 Data extraction and quality measure

The data extraction phase makes use of the masks produced by the overall MISP process. We will discuss data and quality measures extraction for a single spot. The spot pixels highlighted by *SGM* are counted to derive the spot area, while spot pixels highlighted by *RMF* and *GMF* are counted to determinate  $FpixelR$  and  $FpixelG$ ; their values (coming from original image pair) are added to obtain  $FsumR$  and  $FsumG$ . Local Background pixels indicated by *RMLB* and *GMLB* are counted to obtain  $LBpixelR$ ,  $LBpixelG$ , and added to calculate  $LBsumR$  and  $LBsumG$ . Using such mentioned values,  $FmeanR$ ,  $FmeanG$ ,  $LBmeanR$  and  $LBmeanG$  are obtained in a obvious way. It is also calculated the *CorrectSignal* as difference among  $Fmean$  and  $LBmean$  for boths red and green channels. To determinate quality measure are used the formulas in section 2.3 using also *GGM*, *SGM* and the information previously calculated.

### 4.3 Experimental results

Preliminary results show how the proposed pipeline is able to capture in a more reliable way the underlying signal distribution of input data. In Figure 6 a sub-microarray having 5 x 5 spots [2], has been processed with MISP and for sake of comparison with ScanAlyze processing [1].

The results clearly show how the adaptive criterion is very accurate in the foreground/background separation. The ScanAlyze processing has been manually tuned, fixing for each spot the best parameters (circle radius and actual spot center). Such results are also confirmed by quality measures comparison shown, just for a 'critical' spot, in Figure 7.

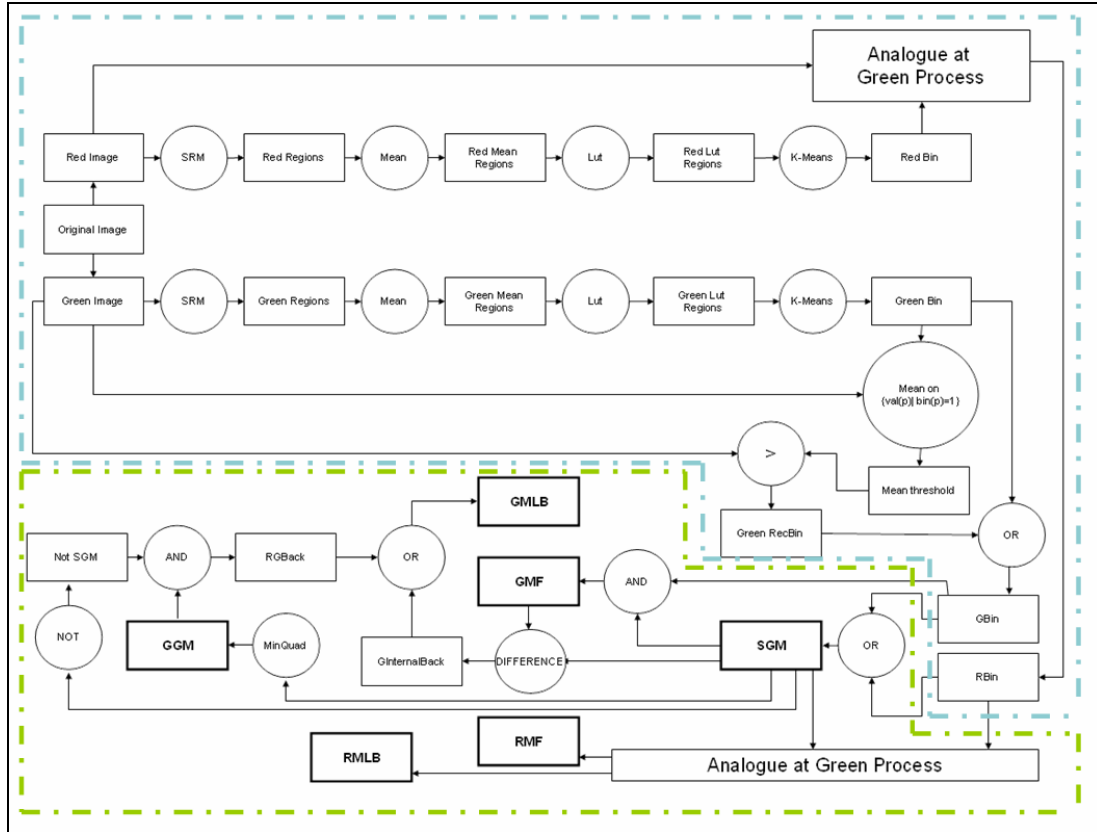
Preliminary results show also how 'q<sub>com</sub>' final value using adaptive pipeline should be slightly modified to take into account the valuable results obtained from such kind of segmentation. Further information and results can be found at the following web address: [www.dmi.unict.it/~jplab](http://www.dmi.unict.it/~jplab).

## 5. CONCLUSIONS AND FUTURE WORKS

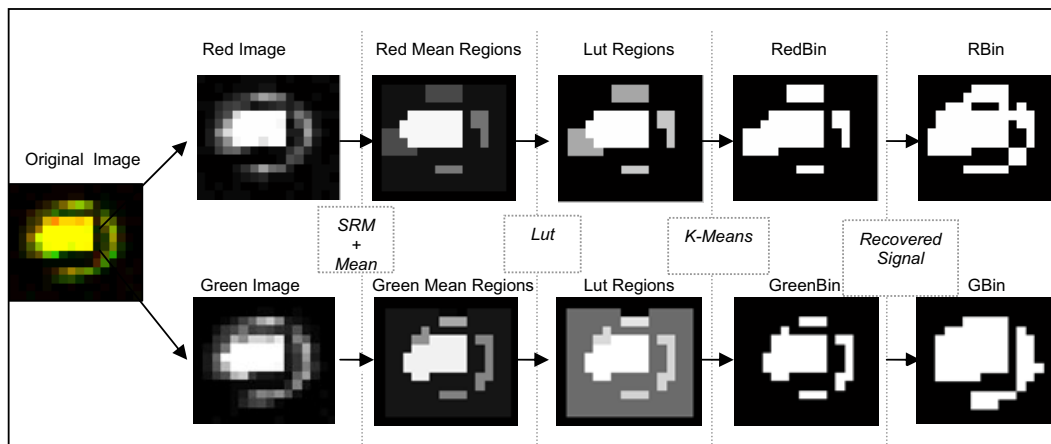
In this paper we propose a new advanced segmentation pipeline for Microarray Image Analysis. The pipeline uses a recent segmentation algorithm SRM based on statistical analysis coupled with K-Means algorithm. The overall MISPP process produces a set of binary masks used to derive accurate spots information and quality measures. A software prototype system has been developed; it includes visualization, segmentation, information and quality measure extraction. Experiments show the effectiveness of the proposed pipeline both in terms of visual accuracy and measured quality values. Future works will include the possibility to further image processing techniques (e.g. noise reduction), an extensive experimental phase devoted to better understand the improvements obtained in terms of biomedical data analysis and an accurate parameters estimation.

## REFERENCES

- [1] M. Eisen, *ScanAlyze*, <http://rana.lbl.gov/EisenSoftware.htm>, Software and Documentation, 1999
- [2] L. Heyer, *Magic Tool and Microarray Sample Data*, <http://www.bio.davidson.edu/projects/magic/magic.html>
- [3] L. Heyer, D. Z. Moskowitz, J. A. Abele, P. Karnik, D. Choi, A. M. Campbell, E. E. Oldham, B. K. Akin, *MAGIC Tool: integrated microarray data analysis*, Bioinformatic Application Note, Vol. 21, No. 9, 2005, pp. 2114–2115
- [4] NCBI, Microarray: chipping away at the mysteries of science and medicine, <http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html>, 2004
- [5] Y. H. Yang, M. J. Buckley, S. Dudoit, T. P. Speed, *Comparison of Methods for Image Analysis on cDNA Microarray Data*, Journal of Computational and Graphical Statistics, Vol. 11, No. 1, 2002, pp. 108-136
- [6] Y. H. Yang, M. J. Buckley, T. P. Speed, *Analysis of cDNA microarray images*, Briefings In Bioinformatics, Vol. 2, No. 4, December 2001, pp. 341-349
- [7] X. Wang, S. Ghosh, S. W. Guo, *Quantitative Quality control in microarray image processing and data acquisition*, Nucleic Acids Research, Vol. 29, No. 15e75, 2001
- [8] U. Sauer, C. Preininger, S. R. Hany, *Quick & Simple: Quality Control of Microarray Data*, Bioinformatics, Advance Access, December 2004
- [9] K. Groch, A. Kuklin, A. Petrov, S. Shams, *Image Segmentation and Quality Control Measures in Microarray Image Analysis*, J. Assoc. Lab. Automation, Vol. 6, No. 3, 2001, pp. 73-76
- [10] Angulo, Serra, *Automatic analysis of DNA microarray images using mathematical morphology*, Bioinformatics, Vol. 19, No. 5, 2003, pp. 553-562
- [11] Axon Instruments Inc, *GenePix Pro User's Guide*, <http://www.axon.com/>, Software and Documentation, 2001
- [12] R. Nock, F. Nielsen, *Statistical Region Merging*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No. 11, Nov. 2004, p.p. 1452-1458
- [13] S. Beucher, F. Meyer, *The Morphological Approach to Segmentation: The Watershed Transformation*, Mathematical Morphology in Image Processing, Vol. 34 of Optical Engineering, New York: Marcel Dekker, pp. 433-481
- [14] R. Adams, L. Bischof, *Seeded Region Growing*, IEEE Transactions on pattern analysis and machine intelligence, Vol. 16, No. 6, June 1994, pp. 641 -647
- [15] R. Nagarajan, *Intensity Based Segmentation of Microarray Images*, IEEE Transaction on Medical Imaging, Vol. 22, No. 7, July 2003, pp. 882-889
- [16] R. Nock and F. Nielsen, *Semi-supervised statistical region refinement for color image segmentation*, Pattern Recognition Vol. 38, Issue 6, June 2005, pp. 835-846
- [17] S. Wu, H. Yan, *Microarray Image Processing Based on Clustering and Morphological Analysis*, First Asia Pacific bioinformatics conference on Bioinformatics 2003 – Vol. 19, pp. 111 - 118, 2003
- [18] A. Jain, R. Dubes, *Algorithm For Clustering Data*, Prentice Hall, 1988
- [19] A. Jain, M.N. Murthy, P.J. Flynn, *Data Clustering: A Review*, ACM Computing Reviews, Nov 1999
- [20] R. C. Gonzales, R. E. Woods, *Digital Image Processing*, Second Edition, Prentice Hall, 2002



**Figure 1:** MISP: Microarray Image Segmentation Pipeline. Cyan is dashed line refers to Spot-Background Separation block, while Green refers to Foreground and Local Background identification.



**Figure 2:** Spot – Background separation.

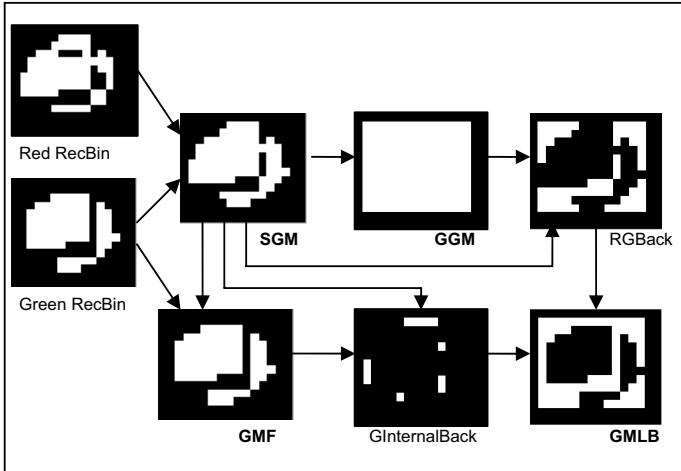


Figure 3: Green Channel Foreground and local background identification.

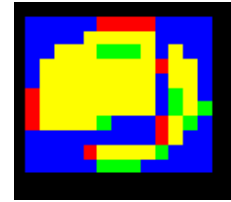


Figure 4: Microarray Image Semantic Color Region. Background (*black*), Local Background (*blue*), Red Channel Foreground (*red*), Green Channel Foreground (*green*), Red Channel and Green Channel Foreground (*yellow*).

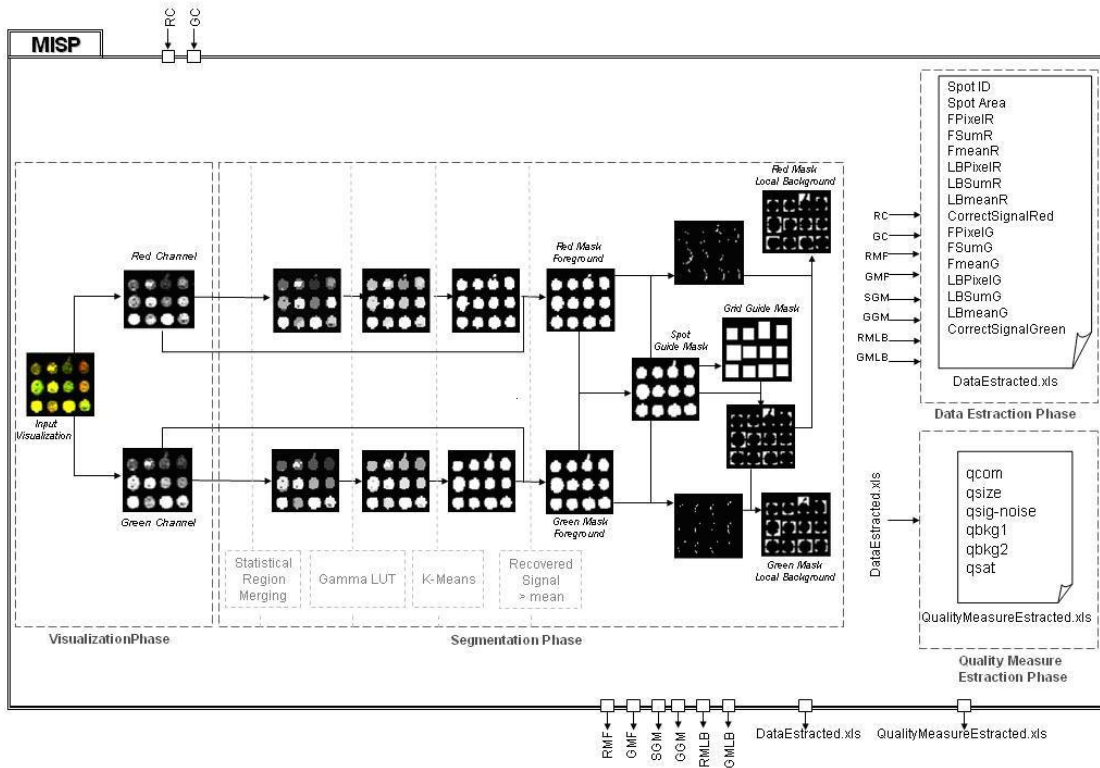


Figure 5: MISP software prototype architecture

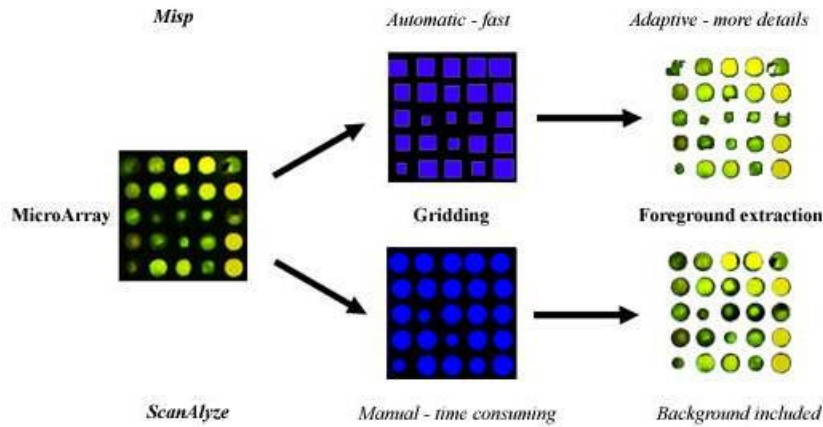


Figure 6: MISP vs. Scanalyze visual Comparison.

	Channel Red					Channel Green				
	qcom	qsize	qsig-noise	qbkg1	qbkg2	qcom	qsize	qsig-noise	qbkg1	qbkg2
MISP	0.723	0.897	0.812	0.621	0.628	0.725	0.779	0.821	0.636	0.681
Scanalyze	0.621	0.881	0.799	0.426	0.495	0.628	0.881	0.785	0.450	0.501

Figure 7: MISP vs. Scanalyze Analytical Comparison.

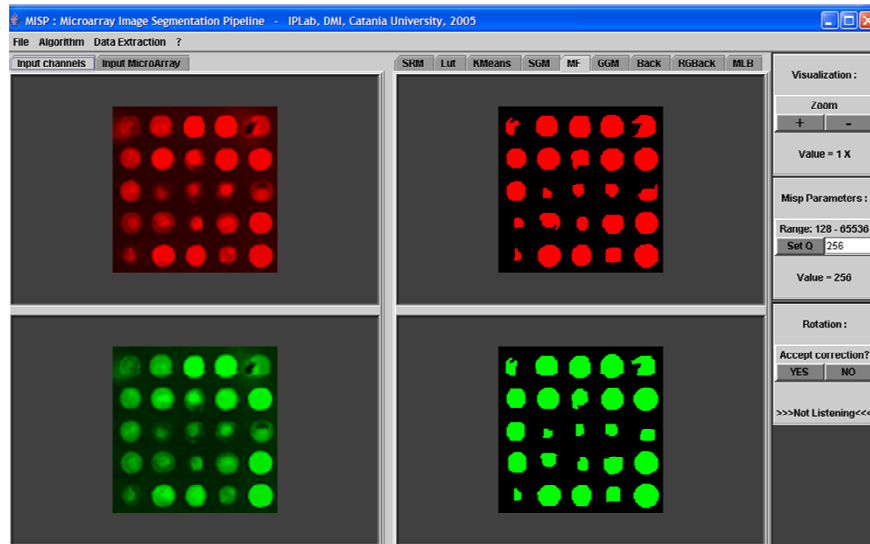


Figure 8: MISP GUI.