A Spot Modeling Evolutionary Algorithm for Segmenting Microarray Images

Eleni Zacharia and Dimitris Maroulis
Department of Informatics and Telecommunications, University of Athens
Greece

1. Introduction

cDNA microarrays is one of the most fundamental and powerful tools in biotechnology. Despite its relatively late discovery in 1995, it has since been utilized in many biomedical applications such as cancer research, infectious disease diagnosis and treatment, toxicology research, pharmacology research, and agricultural development. The reason for its broad use is that it enables scientists to analyze simultaneously the expression levels of thousands of genes over different samples (Leung et al., 2003).

More precisely, the process of a microarray experiment (Campbell et al., 2007) starts with the selection of a set of DNA probes that are of particular interest. A robot places the selected DNA probes on a glass slide, creating an invisible array of DNA dots. Two distinct populations of mRNAs (messenger RNAs) are then isolated from a control sample (i.e a cell developed under normal conditions) and a test sample (i.e. a cell developed under a specific treatment). The mRNA populations are reversely transcribed into cDNA (complementary DNA) populations which in turn are colored with separate fluorescent dyes of different wavelengths (i.e. Cy3 and Cy5). The dyed cDNA populations are mixed with purified water and the solution is placed on the glass slide in order for the cDNA populations to be hybridized with the slide's DNA dots. Finally, the hybridized glass slide is fluorescently scanned twice; one scan for each dye's wavelength. Hence, two digital images are produced, one for each population of mRNA. Each digital image contains a number of spots (corresponding to the DNA-cDNA dots) of various fluorescence intensities. Given that the intensity of each spot is proportional to the hybridization level of the cDNAs and the DNA dots, the gene expression information is obtained by analyzing the digital images.

As stated by Yang et al (Yang et. al, 2002), the process of analyzing a microarray image can be divided into three main phases, namely: "Gridding", "Spot-Segmentation" and "Spot-Intensity extraction". During the 1st phase, the microarray image is segmented into numerous compartments, each containing one individual spot and background. During the 2nd phase each compartment is individually segmented into a spot area and a background area, while during the 3rd phase the brightness of each spot is calculated. The expression-levels of the genes in these spots are a direct result of their individual brightness.

Amongst the stages of the microarray-image analysis, spot-segmentation remains the most challenging one. Ideally, the existing spots inside the microarray image are aligned in 2D array layouts. These 'ideal spots' also have a circular 2D shape with fixed diameters, while their intensity peaks at their central region and declines at regions further from their centre. In reality however, microarray images have poor quality due to the existence of noise and/or artifacts as well as due to uneven background (Wang et. al, 2003). Additionally, many spots are rather different to the ideal ones as they vary in size, shape and position due to imperfect sample-preparation and hybridization processes (Tu et. al, 2002). Last but not least, some spots are so poorly contrasted that are not clearly visible (Chen et al 2006).

As a result, a number of spot-segmentation techniques have been developed, some of which have been incorporated into commercial software programs. The fixed circle segmentation algorithm [implemented by the ScanAlyze software program (Eisen, 1999)] or the adaptive circle segmentation algorithm [implemented by the Dapple software program (Buhler et. al, 2000)] assumes that microarray spots are circular. However, this assumption is in fact invalid since a spot's morphology – as previously mentioned - is not always a circle. Moreover, both of these techniques require input parameters in order to define the spot's diameter. The adaptive shape-segmentation [implemented by the Spot software program(Buckley, 2000)] has been suggested in order to deal with the various shapes of the spots. The algorithm can segment regions of irregular shapes by implementing a watershed algorithm. However, a drawback in this method is that its performance is based on the appropriately specified number and locations of the starting points (seeds). Chen et al (Chen et al, 1997) suggested a thresholding method based on the statistical Mann-Whitney test. A disadvantage of this method is that its performance relies on the appropriate choice of background samples. Clustering algorithms, such as K-means, hybrid K-means and, fuzzy C-means (FCM) have been also applied in order to determine which pixels belong to the spot area and which ones to the background area (Bozinov et. al 2002), (Rahnenfuhrer et. al, 2003), (Nagarajan et. al 2003). Nevertheless, these methods become inaccurate, when the spots are poorly contrasted or when the spots are very close to each other. In the latter case, instead of segmenting the real spot, these methods may segment portions of neighboring spots. Another segmentation method, based on the clustering of pixels' values, is the model-based segmentation algorithm, proposed by Li et al (Li et. al, 2005). A disadvantage of this method is that it may over-segment the microarray spots since the number of clusters is determined automatically. Finally, there are segmentation methods based on active contours and multiple snakes (Ho et al, 2008),(Srinark et al 2004), (Srinark et al 2001). These methods give inaccurate results when the compartment is contaminated with noise and artifacts. The Markov Random Fields method (MRF) (Demirkaya et. al, 2005) utilizes the neighboring information, along with the intensity information, based on an MRF modeling of the compartment. However, one major drawback of this method is that it requires an initial classification of the pixels which in turn affects the final results. The segmentation method included in the Matarray toolbox of Matlab (Wang et al, 2001) combines both spatial and intensity information. A disadvantage of this method is that it requires input parameters in order to segment the spots.

All aforementioned techniques require human intervention in order to define input parameters or to correct the segmentation results. This apparent lack of automation can be disadvantageous during microarray image analysis. Indeed, human intervention may inevitably modify the actual results of the microarray experiment and lead to erroneous biological conclusions. Therefore, the necessity of an accurate and automatic spot-segmentation technique becomes obvious.

In this chapter, the spot-segmentation stage of the microarray-image analysis is expressed as an optimization problem which is subsequently solved by using genetic algorithms and fuzzy logic. In particular, a genetic algorithm (GA) represents the real-spots of the cDNA microarray image with spot-models, in a 3D space. The segmentation of the real-spots is conducted by drawing the contours of the spot-models. It should be noted that the spot-model presented in this chapter can be used for the representation of all types of real microarray spots such as peak-shaped, volcano-shaped and doughnut-shaped spots. Consequently, the proposed method can segment all possible types of microarray spots. Moreover, the genetic algorithm has been further developed in order to be noise-resistant and yield more accurate results. It adopts the Fuzzy Logic so as to take into account the uncertainties that exist in the pixels' intensities due to noise, artifacts and uneven background. Contrary to existing software systems, the proposed spot-segmentation method is fully automatic as it does not require any input parameters; it is also noise resistant and yields excellent results even under the following adverse conditions: i) the appearance of various spot-shapes, such as peak-shaped, volcano-shaped and doughnut-shaped spots, ii) the appearance of spots of diverse intensities, such as low-intensity spots or saturated spots and iii) the appearance of various spot-sizes. Last but not least, it outperforms other image analysis software programs as well as other well-known published techniques.

2. Genetic algorithms

Genetic Algorithms (GAs) are powerful, stochastic, non-linear optimization tools based on the principles of natural selection and evolution (Golderbg, 1989). Compared to traditional search and optimization tools (such as Blind Search Algorithms), GAs demonstrate superior performance, given that they are robust optimizers, suitable for solving problems for which there is little or no a priori knowledge of the underlying processes.

Given a specific optimization problem, a typical GA searches for the optimal solution as follows: Firstly, it creates a finite number of potential solutions encoded as alpha-numerical sequences called Chromosomes. These Chromosomes constitute an initial Population Pop_1. Subsequently, the GA produces a new Population Pop_2 according to the following: The Chromosomes constituting the Pop_1 are evaluated using a Fitness Function. Thereafter, the GA evolves the Population Pop_1 into a new Population Pop_2 using the three Genetic Operators: Reproduction, Crossover, and Mutation. This Evolutionary Cycle from one Population to the next (Pop_1 to Pop_2, Pop_2 to Pop_3 and so forth) continues until a specific termination criterion is satisfied. Subsequently, the essential elements of the GA are: Chromosome representation, Chromosome evaluation, the Evolutionary cycle, and the Termination criteria.

A Chromosome is often represented as a simple alpha-numerical sequence which encodes the values of variables defining a possible solution to the optimization problem at hand. Although a traditional GA uses a binary number in order to encode these variables, in the present application, a Real-Coded Genetic Algorithm (RCGA), which uses real values, is applied. The reason is that real-coded Chromosomes exhibit various advantages over binary-coded Chromosomes as they can use large or unknown domains for the variables they encode. On the other hand, assuming that the Chromosome has a fixed length, binary implementations cannot increase the domain without sacrificing precision (Herrera et al., 1998).

The evaluation of the Chromosome is based on a Fitness Function which assigns to the Chromosome a Fitness Value measuring the quality of the solution that the Chromosome represents. Naturally, the Fitness Function depends on the particular optimization problem at hand and on the Chromosome representation.

Reproduction, Crossover and Mutation are the three Genetic Operators used for the creation of new Chromosomes (Herrera et al., 1998). All of them have been implemented in several, distinct fashions depending on the Chromosome representation.

Common terminating criteria are: (i) A solution that satisfies the defined minimum standards, (ii) The attainment of a maximum number of Populations, (iii) The attainment of a fixed number of Populations for which the Fitness Value of the best Chromosome remains the same, and (v) Combinations of the above (Hayes, 2006).

3. Microarray spots

The following three types of microarray spots can be identified in a microarray image (Kim et al. 2007):

1. Peak-shaped spot (Fig. 1a); this type of spot has an intensity that peaks at its central region and declines at regions further from the centre. In the case when the peak is thin, the spot resembles to a 2D-Gaussian function. In the case when the peak is wide, the spot resembles to a plateau.
2. Volcano-shaped spot (Fig. 1b); this type of spot is defined as the peak-shaped spot having a small hole in the area of its peak. It therefore resembles to a volcano.
3. Doughnut-shaped spot (Fig. 1c); this type of spot has a thin rim of high intensity and a large hole of very low intensity at its central region.

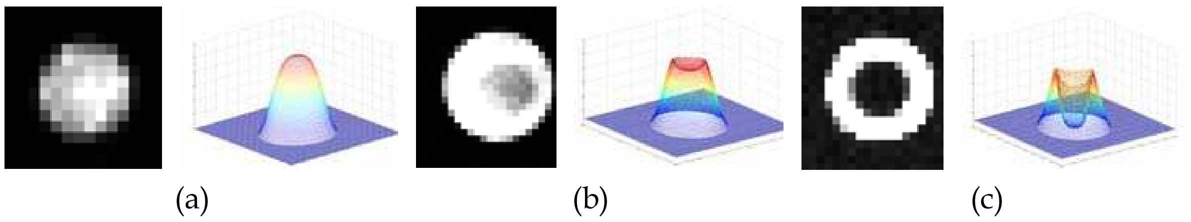


Fig. 1. Three types of real microarray spots in 2D and 3D dimensions:(a) a peak-shaped spot, (b) a volcano-shaped spot, and (c) a doughnut-shaped spot.

4. Proposed spot-segmentation method

Given that I_{REAL} is one of the compartments of a real microarray image containing one individual spot S_{REAL} and background B_{REAL}, the segmentation procedure aims to the delineation of the boundaries of the spot S_{REAL}. The segmentation procedure is divided into two stages:

1st Stage: The compartment I_{REAL} of the microarray image is optimally represented by a 3D compartment-model I_{MODEL}.

2nd Stage: The boundaries of the microarray spot S_{REAL} are depicted by drawing the contour of the spot-model S_{MODEL}.

4.1 Morphological models for a microarray spot and its compartment

Due to the aforementioned common spots' characteristics, a microarray compartment can be represented by a 3D compartment-model, in which the third dimension represents the intensity. More precisely, a microarray spot can be represented using: i) a 3D-curve representing the main-body S_{MB} of the spot-model, and ii) a 3D-curve representing the inner-dip S_{ID} of the spot-model.

4.1.1 The spot-model and its components

The main-body and the inner-dip 3D curves have opposite orientation and they resemble the 3D Gaussian or plateau curve (Fig.2). More precisely, the main body of the spot-model $S_{MB}(x,y)$ is defined by the following equation:

$$S_{MB}(x,y) = h_{MB} \cdot \left[erf\left(a_{MB} + r_{MB}(x,y)\right) + erf\left(a_{MB} - r_{MB}(x,y)\right) \right] , \tag{1}$$

Where

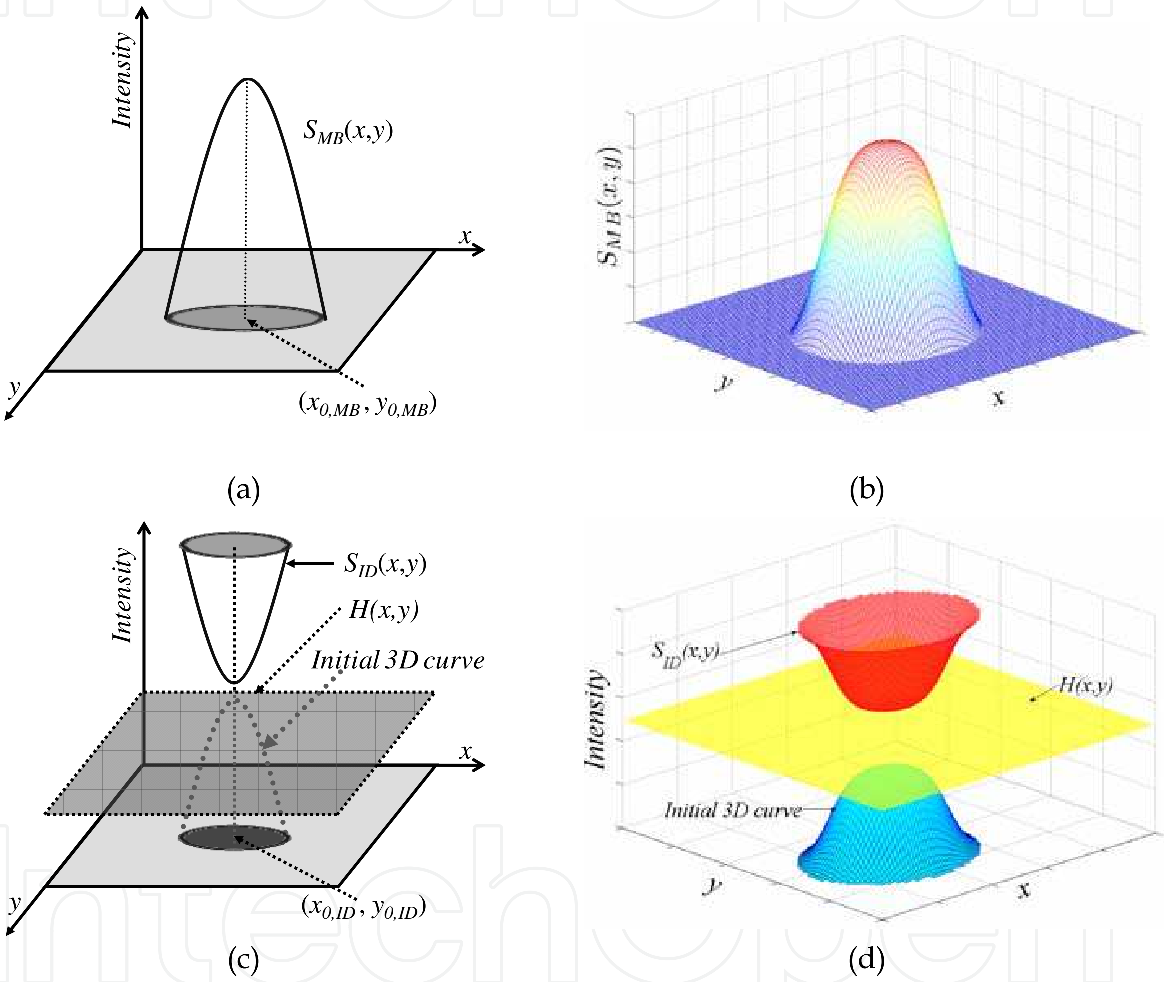


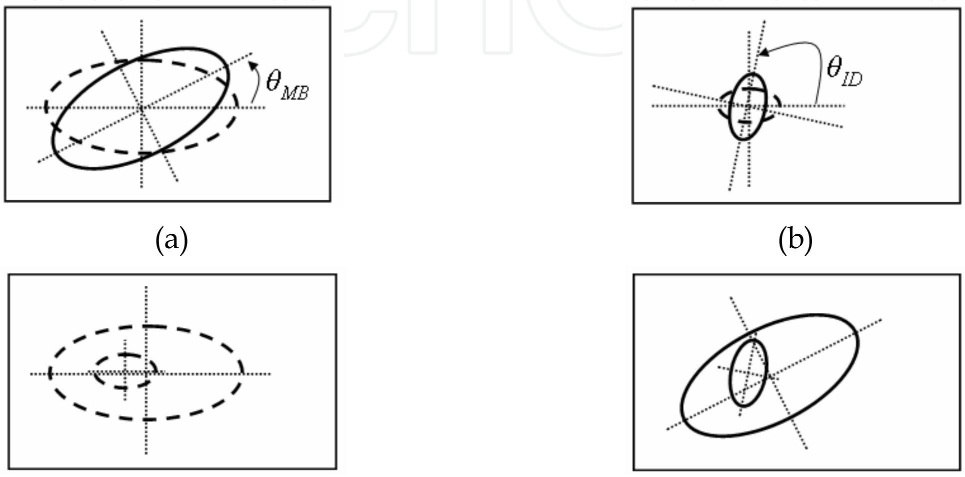
Fig. 2. Components of the spot-model: (a,b) The main-body $S_{MB}(x,y)$ of the spot-model,(c,d) 3D representation of the inner-dip $S_{ID}(x,y)$ 3D-curve, the initial 3D-curve and the horizontal surface *H(x,y)*.

$$r_{MB}(x,y) = \sqrt{\frac{(x - x_{0,MB})^2}{D_{x,MB}} + \frac{(y - y_{0,MB})^2}{D_{y,MB}}} . \tag{2}$$

h_{MB} controls the height of the main body of the spot-model. $erf(z)$ denotes the error function encountered in integrating the normal distribution. $(x_{0,MB}, y_{0,MB})$ are the coordinates of the center of the main body of the spot-model on the 2D plane. $D_{x,MB}$ and

$D_{y,MB}$ control the slope of the 3D curve at two main directions (x and y) of the 2D plane, while a_{MB} controls the shape of the 3D curve. For $a_{MB} \to 0$, $S_{MB}(x,y)$ resembles a two-dimensional Gaussian function, while for $a_{MB} \to \infty$, $S_{MB}(x,y)$ resembles a plateau or saturated spot. A more detailed illustration of $S_{MB}(x,y)$ is depicted in Fig. 3. It is worth pointing out that $S_{MB}(x,y) \in [0, S_{MB}(x_{0,MB}, y_{0,MB})]$.

The inner dip of the spot-model $S_{ID}(x,y)$ is defined as a symmetrical 3D curve - in respect to an horizontal surface $H(x,y)$ - to an 'initial 3D curve' which derives from eq. (1), and whose maximum value appears in its center $(x_{0,ID}, y_{0,ID})$.



(a) (b)

(c) (d)

Fig. 3. Illustration of the rotations: (a) The dashed curve represents the contour of the main body of the spot-model before the rotation, while the solid curve represents the contour of the main body of the spot-model after the rotation, (b) The dashed curve represents the contour of the inner dip of the spot-model before the rotation, while the solid curve represents the contour of the inner dip of the spot-model after the rotation, (c) The contour of the total spot-model without applying the rotations, and (d) The contour of the total spot-model after applying the rotations.

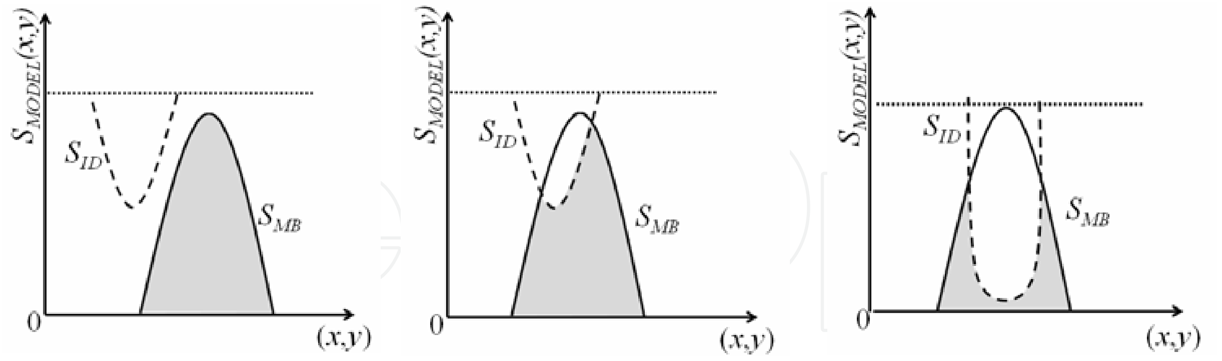
4.1.2 Total spot-model

The total spot-model $S_{Model}(x,y)$ as a function of x, y is defined by the following mathematical equation:

$$S_{MODEL}(x,y) = Min\left[S_{MB}(x_{\theta,\mathrm{MB}}, y_{\theta,\mathrm{MB}}), S_{ID}(x_{\theta,ID}, y_{\theta,ID})\right] \tag{3}$$

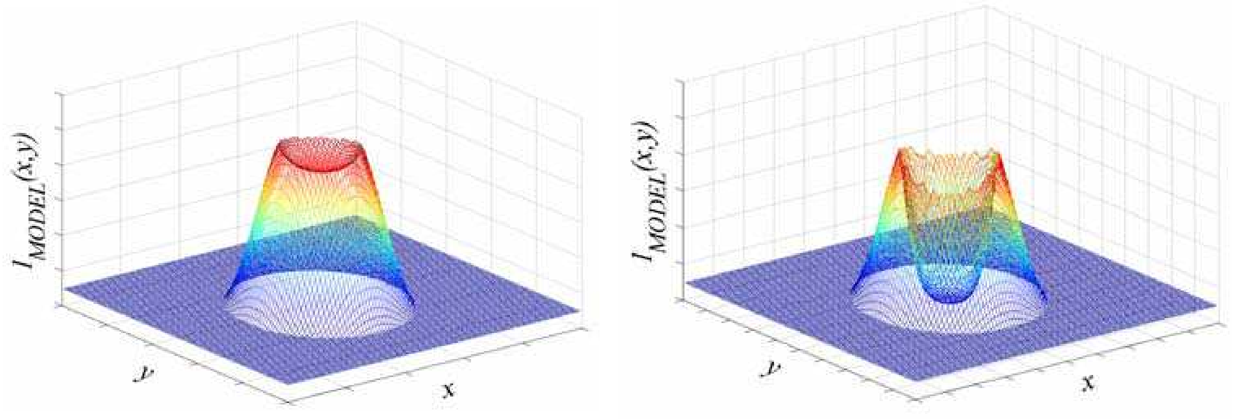
where $(x_{\theta,\mathrm{MB}}, y_{\theta,\mathrm{MB}})$ are the rotated coordinates of the (x,y) by an angle θ_{MB} around the 3D curve's center $(x_{0,MB}, y_{0,MB})$ of the main body of the spot-model. Likewise, $(x_{\theta,ID}, y_{\theta,ID})$ are the rotated coordinates of the (x,y) by an angle θ_{ID} around the 3D curve's center $(x_{0,ID}, y_{0,ID})$ of the inner dip of the spot-model.

Rotating the two compartments of the spot-model through the angles θ_{MB} and θ_{ID}, permits both the $S_{MB}(x,y)$ and $S_{ID}(x,y)$ 3D curves to have any possible direction on the 2D plane. An example is shown in Fig. 3. A graphical explanation of eq. (3) is depicted in Fig. 4. The resulting total-models are the areas colored with grey. Depending of the distance between the S_{MB} and S_{ID} centers and the height of the S_{ID} 3D curve, the resulting total-model can resemble a peak-shaped spot (Fig.4a), a volcano-shaped spot (Fig.4b), or a doughnut-shaped spot (Fig.4c).



(a) (b) (c)

Fig. 4. S_{MB} and S_{ID} components of the morphological models of: (a) a peak-shaped spot, (b) a volcano-shaped spot, and (c) a doughnut-shaped spot. The total morphological models are the grey areas.



(a) (b)

Fig. 5. Examples of the 3D compartment-models containing: (a) a volcano-shaped spot, and (b) a doughnut-shaped spot.

4.1.3 The compartment-model

The compartment-model $I_{MODEL}(x,y)$ as a function of x,y is defined by the following mathematical equation:

$$I_{MODEL}(x,y) = Max\left[B_{AV}, S_{MODEL}(x,y)\right] \tag{4}$$

where B_{AV} denotes the average background intensity of the compartment-model I_{MODEL}. B_{AV} corresponds to a threshold of the lowest values of the $S_{MODEL}(x,y)$. Pixels whose values are lower than B_{AV} belong to background, and their values are set equal to B_{AV}. Thus: $I_{MODEL}(x,y) \in [B_{AV}, S_{MB}(x_{0,MB}\ \ y_{0,MB})]$.

Two examples of compartment–models $I_{MODEL}(x,y)$ are depicted in Fig. 5. The first compartment-model contains a volcano-shaped spot-model while the second compartment-model contains a doughnut-shaped spot-model.

4.2 Optimum 3D representation and definition of real-spot contour

The first stage of the segmentation procedure is regarded as an optimization problem of modeling a microarray compartment and it is tackled by using the proposed genetic algorithm. A genetic algorithm determines the compartment-model which optimally represents the real-one. More precisely, it determines the values of the variables of the compartment-model (eq. 4) so that the resulting compartment-model represents optimally the real-one.

4.2.1 Chromosome representation

A chromosome *m* represents a specific compartment-model I_{MODEL}^{m} in a three-dimensional space, where *m* stands for a specific chromosome. It is therefore a simple numerical sequence which encodes the values of the variables defining the specific compartment-model. It consists of 3 segments: The first segment encodes the value of the average background intensity of the compartment-model B_{AV}^{m}. The second segment encodes the values of the variables of the main-body S_{MB}^{m} of the spot-model S_{MODEL}^{m}, while the third segment encodes the values of the variables of the inner-dip S_{ID}^{m} of the spot-model S_{MODEL}^{m}.

4.2.2 Chromosome evaluation

The aim of the genetic algorithm is the maximization of the resemblance between the compartment-model I_{MODEL}^{m} and the real-one I_{REAL}. In other words, the higher the resemblance of the compartment-model I_{MODEL}^{m} (represented by the chromosome *m*) to the real-compartment I_{REAL} is, the higher the value of the fitness function of a chromosome *m* becomes. Based on the aforementioned remark, the chromosome evaluation contains the following three main objectives:

1. Maximization of the degree of overlap between the area containing the real microarray spot S_{REAL} and the area containing the spot-model S_{MODEL}^{m} (represented by chromosome *m*),
2. Maximization of the resemblance between the real microarray spot S_{REAL} and the main body of the spot-model S_{MB}^{m} (represented by chromosome *m*). In this case, let I_{MB} be a model-compartment which contains only the main body of the spot-model (instead of the total-spot model). In correspondence with eq. (4), I_{MB} is defined by the following equation:

   $$I_{MB}(x,y) = Max\left[B_{AV}, S_{MB}(x,y)\right] \tag{7}$$

   Subsequently, the aforementioned maximization is equivalent to the maximization of the resemblance between the real-compartment I_{REAL} and the model-compartment I_{MB}^{m} (represented by chromosome *m*).
3. Maximization of the resemblance between the real-compartment I_{REAL} and the model-compartment I_{MODEL}^{m} containing the total spot-model S_{MODEL}^{m}.

It should be noted, however, that since the real-compartment is contaminated with noise and artifacts, its intensity values are noticeably fluctuated – even between two consecutive pixels – resulting in a scabrous 3D-curve that contains many peaks. As a result, pixels belonging to the spot area S_{REAL} may have lower intensity values than the pixels belonging to the background area B_{REAL}. Correspondingly, pixels belonging to the background area B_{REAL} may have higher intensity values than the pixels belonging to the spot area S_{REAL}.

Contrary to the scabrous 3D-curve of the real-compartment, the compartment-model has a smooth 3D-curve. Consequently, some of the points of the 3D-curve of the compartment-model are identical to the points of the real-one while some others interpolate the points of the real-one. The identical points should belong mostly to the region near the spot's contour while the interpolated points should belong mostly to spot areas or background areas.

To deal with the ambiguity and vagueness of the intensity values of pixels – due to noise, artifacts and uneven background – the genetic algorithm adopts the Fuzzy Logic. We set the '*membership degree*' of a pixel *p* to belong to the background area or to the spot area according to the following two rules of fuzzy logic theory:

1. The smaller the intensity's value $I_{REAL}(p)$ is, the greater the '*membership degree*' that p_r belongs to the background area becomes, and
2. The higher the intensity's value $I_{REAL}(p)$ is, the greater the '*membership degree*' that p_r belongs to the spot area becomes.

Based on the two aforementioned rules, the membership function $\mu_B(p)$ of a pixel p in order to belong to the background area and the membership function $\mu_S(p)$ of a pixel p in order to belong to the spot area are defined by the following equations:

$$\mu_B(p) = \begin{cases} 1, & if \;\; I_{REAL}(p) \le I_B \\ \dfrac{I_F - I_{REAL}(p)}{I_F - I_B}, & if \;\; I_B < I_{REAL}(p) < I_F \\ 0, & if \;\; I_{REAL}(p) \ge I_F \end{cases} . \tag{5}$$

and,

$$\mu_S(p) = \begin{cases} 0, & if \;\; I_{REAL}(p) \le I_B \\ \dfrac{I_{REAL}(p) - I_B}{I_F - I_B}, & if \;\; I_B < I_{REAL}(p) < I_F \\ 1, & if \;\; I_{REAL}(p) \ge I_F \end{cases} \tag{6}$$

where I_B and I_F are two intensity values. More precisely, let I_o be the intensity corresponding to the minimum between the maxima of the two normal distributions which represent the distributions of background pixels and spot pixels (Fig. 6). I_{min} and I_{max} are the minimum and maximum intensity values that appear in the I_{REAL}. Let N_1 be the number of pixels whose intensities' values are less than I_o and, N_2 be the number of pixels whose intensities' values are higher or equal to I_o. I_B is chosen so that $k \cdot N_1$ number of pixels have intensity lower or equal to I_B, where k is a constant ($0 \le k \le 1$). I_F is chosen so that $k \cdot N_2$ number of pixels have intensity higher or equal to I_F.

Fig. 7 represents the membership functions $\mu_B(p)$ and $\mu_S(p)$. It becomes obvious that pixels with intensity lower or equal to I_B belong to the background area ($\mu_B(p) = 1$ and $\mu_S(p) = 0$), while pixels with intensity higher or equal to I_F belong to the spot area ($\mu_B(p) = 0$ and $\mu_S(p) = 1$). Pixels, with intensity between I_B and I_F, have a 'membership degree' $\mu_B(p)$ to belong to the background area and a 'membership degree' $\mu_S(p)$ to belong to spot area ($\mu_B(p) \ne 0$ and $\mu_S(p) \ne 0$).

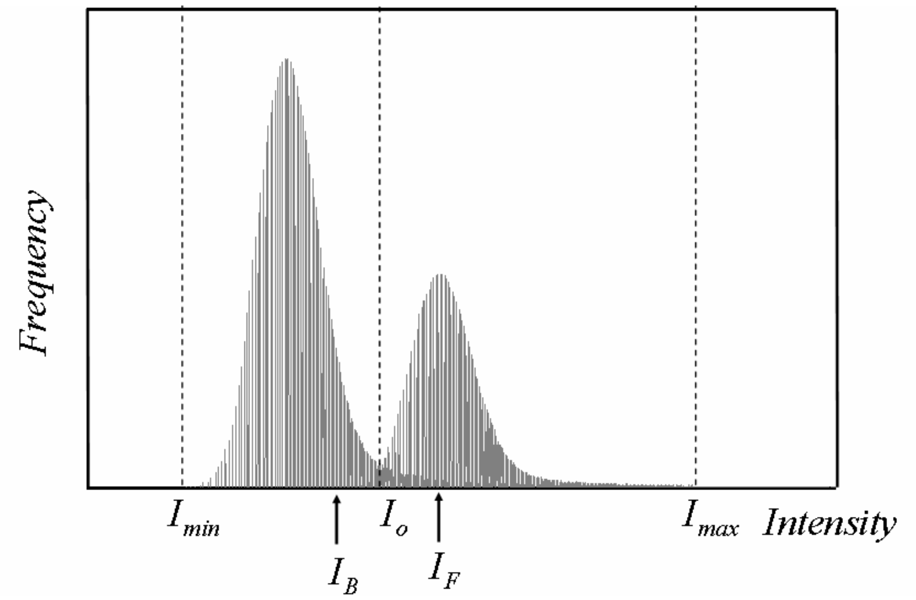


Fig. 6. A typical histogram of a real microarray compartment. The left curve corresponds to background pixels while the right curve corresponds to spot pixels.

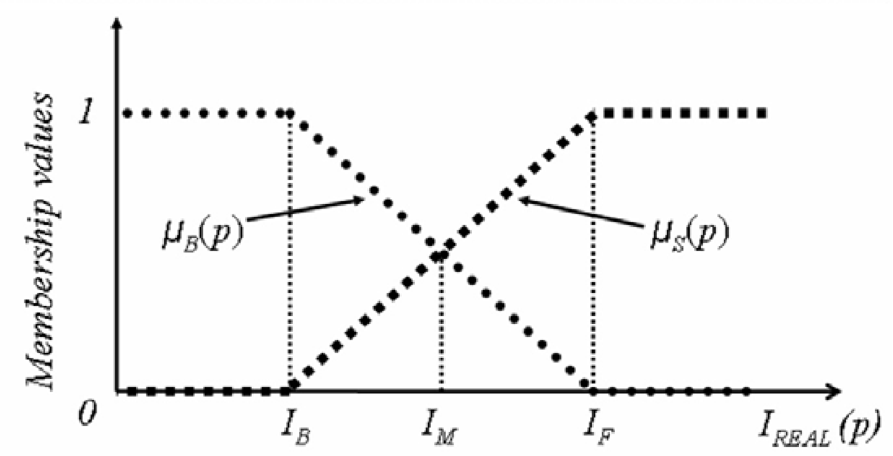


Fig. 7. The membership functions $\mu_B(p)$ and $\mu_S(p)$.

In the next subsections, the three aforementioned objectives for the chromosome evaluation, as well as the way they are combined in order to form the fitness function are apposed in detail.

4.2.2.1 Overlap of the area containing the real microarray apot S_{REAL} and the area containing the spot-model S^m_{MODEL}

Let $\hat{S}_{REAL}$ be the set of pixels whose intensity value is higher or equal to I_M and, $\hat{B}_{REAL}$ be the set of pixels whose intensity value is lower than I_M (Fig. 7). Ideally, $\hat{S}_{REAL}$ contains the pixels belonging to the spot area S_{REAL} while $\hat{B}_{REAL}$ contains the pixels belonging to the background area B_{REAL}. By overlapping the I_{REAL} and the I^m_{MODEL} (Fig. 8) four different regions can be identified: 1) S_A is the set of pixels whose members are the pixels which are located in the area of $\hat{S}_{REAL}$ and in the area of S^m_{MODEL}. 2) S_B is the set of pixels whose members are the pixels which are located in the area of $\hat{S}_{REAL}$ and in the area of B^m_{AV}. 3) S_C is the set of pixels whose members are the pixels which are located in the area of $\hat{B}_{REAL}$ and in the area of S^{m}_{MODEL}. 4) S_D is the set of pixels whose members are the pixels which are located in the area of $\hat{B}_{REAL}$ and in the area of B^{m}_{AV}

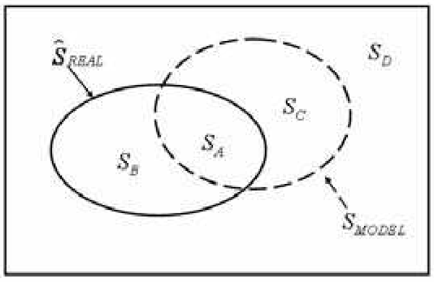


Fig. 8. Overlapping of the I_{REAL} compartment and I^{m}_{MODEL} compartment. The solid curve represents the area of $\hat{S}_{REAL}$ while the dashed curve represents the area of S^{m}_{MODEL}.

Using the aforementioned regions, the true positive rate *TP*(*m*) and the true negative rate *TN*(*m*) can be calculated for the S^{m}_{MODEL}. Due to the uncertainties existing in the pixel's intensities, the pixels contributing to the calculations are weighted; In the calculation of *TP*(*m*), the weight coefficient of a pixel *p* equals to its corresponding 'membership degree' $\mu_S(p)$, while in the calculation of *TN*(*m*) the weight coefficient of a pixel *p* equals to its corresponding 'membership degree' $\mu_B(p)$.

The higher the *TP*(*m*) and the *TN*(*m*) are, the higher the overlapping of the S^{m}_{MODEL} with the $\hat{S}_{REAL}$ is. As a result, the overlap $F_{Overlap}(m)$ of the area containing the real microarray spot S_{REAL} and the area containing the spot-model S^{m}_{MODEL} is defined by the following equation:

$$F_{Overlap}(m) = TP(m) \cdot TN(m) \tag{8}$$

4.2.2.2 Measure for calculating the error of a model-compartment at a pixel p

Let I^{m}_{C} be either the model-compartment I^{m}_{MODEL} or the model-compartment I^{m}_{MB}. If the surface of the real-compartment I_{REAL} was smooth, the error of the model-compartment I^{m}_{C} at a pixel *p* should be defined as:

$$E^{m}_{REAL}(p) = \frac{|I^{m}_{C}(p) - I_{REAL}(p)|}{I_{REAL}(p)} \tag{9}$$

However, the surface is not smooth since the real microarray compartment is contaminated with noise. As a result, the error of the model-compartment I^{m}_{C} at a pixel *p* is defined by the following equation:

$$E^{m}(p) = Min\left[E^{m}_{REAL}(p), E^{m}_{MR}(p)\right] \tag{10}$$

where,

$$E^{m}_{MR}(p) = \frac{|I^{m}_{C}(p) - I_{MR}(p)|}{I_{MR}(p)} \tag{11}$$

and,

$$I_{MR}(p) = \begin{cases} \underset{k \in K}{Median}\left[I_{REAL}(k)\right], \ if\ Max\left[\mu_S(p), \mu_B(p)\right] \geq \lambda_m \\ I_{REAL}(p), otherwise \end{cases} \tag{12}$$

$$K = \left\{k \mid (\mid k - p \mid \leq 1) \ \wedge \mid \mu_S(k) - \mu_S(p) \mid \leq \lambda_d \right\}. \tag{13}$$

$I_{MR}(p)$ equals either to $I_{REAL}(p)$ or to the median intensity value of a set of pixels {*K*} located in a neighborhood near the pixel *p*, according to the values of $\mu_S(p)$ or $\mu_B(p)$. λ_m and, λ_d denote two constants ($0 \leq \lambda_m, \lambda_d \leq 1$) which control the $I_{MR}(p)$ value of a pixel *p*.

4.2.2.3 Resemblance between the real-compartment I_{REAL} and the model-compartment I_{MB}^m

The resemblance $R_{MB}(m)$ between the real-compartment I_{REAL} and the model-compartment I_{MB}^m is defined by the following equation:

$$R_{MB}(m) = f_1(m) \cdot f_2(m), \tag{14}$$

where

$$f_1(m) = \frac{\#\{p \mid p \in S_1\}}{\#\left\{p \mid p \in \hat{S}_{REAL}\right\}}, \tag{15}$$

$$f_2(m) = \frac{\#\{p \mid p \in S_2\}}{\#\left\{p \mid p \in \hat{B}_{REAL}\right\}} \tag{16}$$

and,

$$S_1 = \left\{p \mid E^m(p) \leq E_{MAX} \wedge I_{MB}^m(p) > B_{AV}\right\}, \tag{17}$$

$$S_2 = \left\{p \mid E^m(p) \leq E_{MAX} \wedge I_{MB}^m(p) \leq B_{AV}\right\}. \tag{18}$$

The symbol # denotes the number of the elements of the set that is defined by the brackets {}. E_{MAX} is a positive constant which expresses the maximum acceptable error of the model-compartment at a pixel.

S_1 denotes a set of pixels whose members are the pixels *p* of the compartment-model I_{MB}^m which are located in the area of the main body of the spot-model (S_{MB}) and efficiently represent the corresponding ones of the real compartment I_{REAL} ($E^m(p) \leq E_{MAX}$). Likewise, S_2 denotes a set of pixels whose members are the pixels *p* of the compartment-model I_{MB}^m which are located in the area of the background (B_{AV}) and efficiently represent the corresponding ones of the real compartment I_{REAL}.

$f_1(m)$ denotes the percentage of the $\hat{S}_{REAL}$ pixels which have been efficiently represented by the compartment-model I_{MB}^m. Likewise, $f_2(m)$ denotes the percentage of the $\hat{B}_{REAL}$ pixels which have been efficiently represented by the compartment-model I_{MB}^m. From eq. (14), the further pixels have been efficiently represented by the I_{MB}^m compartment-model, the higher the value of $R_{MB}(m)$ becomes.

4.2.2.4 Resemblance between the real-compartment I$_{REAL}$ and the model-compartment I^{m}_{MODEL}

The resemblance between the real-compartment I_{REAL} and the model-compartment I^{m}_{MODEL} is defined by the following equation:

$$R_{MODEL}(m) = f_3(m) \cdot f_4(m) \tag{19}$$

where

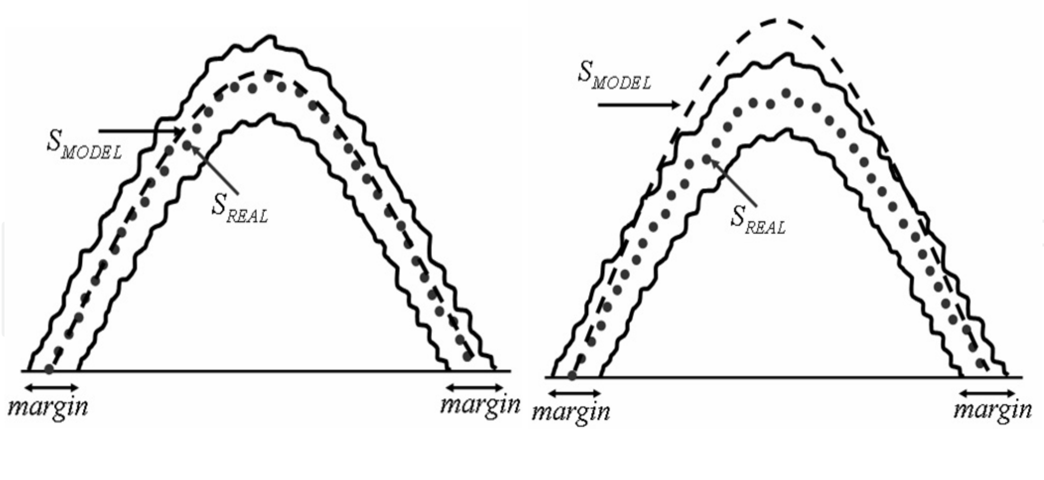
$$f_3(m) = \sum_{p \in I_{REAL}} \mu_S(p) \cdot (1 - w \cdot E^m(p)), \tag{20}$$

$$f_4(m) = \sum_{p \in I_{REAL}} \mu_B(p) \cdot (1 - w \cdot E^m(p)) \tag{21}$$

and,

$$w = \begin{cases} 0.1, & if\ E^{m}_{REAL}(p) \le E_{MAX} \\ 1, & otherwise \end{cases}, \tag{22}$$

If the value of $E^{m}_{REAL}(p)$ of a pixel p is less or equal to E_{MAX} (eq. 22), the error between the model-compartment I^{m}_{MODEL} and the real-compartment I_{REAL} at that pixel is considered negligibly small and thus insignificant. Consequently, the $E^m(p)$ error ((eq. 20) and (eq. 21)) is multiplied by a factor of 0.1 (w=0.1).



(a) (b)

Fig. 9. The dotted curve represents the intensity of a real microarray spot while the dashed curve represents a spot-model. Spot-models whose values fall within the margin, defined by the solid curves, efficiently represent the real microarray spot. (a) An efficient spot-model. (b) An inefficient spot-model.

In fact, the constant E_{MAX} controls an acceptable margin of the error existing between the intensities' values of the compartment-model and the intensities' values of the real compartment. As an example Fig. 9 depicts the margin in the area of the real microarray spot. Spot-models whose values fall within the margin efficiently represent the real microarray spot.

As the value of $f_3(m)$ increases, so does the number of those pixels belonging to $\hat{S}_{REAL}$ which are efficiently represented by the S_{MODEL}. Likewise, as the value of $f_4(m)$ increases, so does the number of those pixels belonging to $\hat{B}_{REAL}$ which are efficiently represented by the B_{AV}. As a result, the greater the number of pixels that are efficiently represented by the I_{MODEL}^{m} compartment-model is, the higher the value of $R_{MODEL}(m)$ becomes.

4.2.2.5 Fitness function

Each chromosome *m* in every population is evaluated using a fitness function, *F*(*m*), which assigns to it a degree of how appropriate a solution to the optimization problem it is. The higher the value of the fitness function, the more appropriate the chromosome is. The Fitness Function, *F*(*m*), of a Chromosome *m* that encodes a possible solution to the particular optimization problem is defined by the following equation:

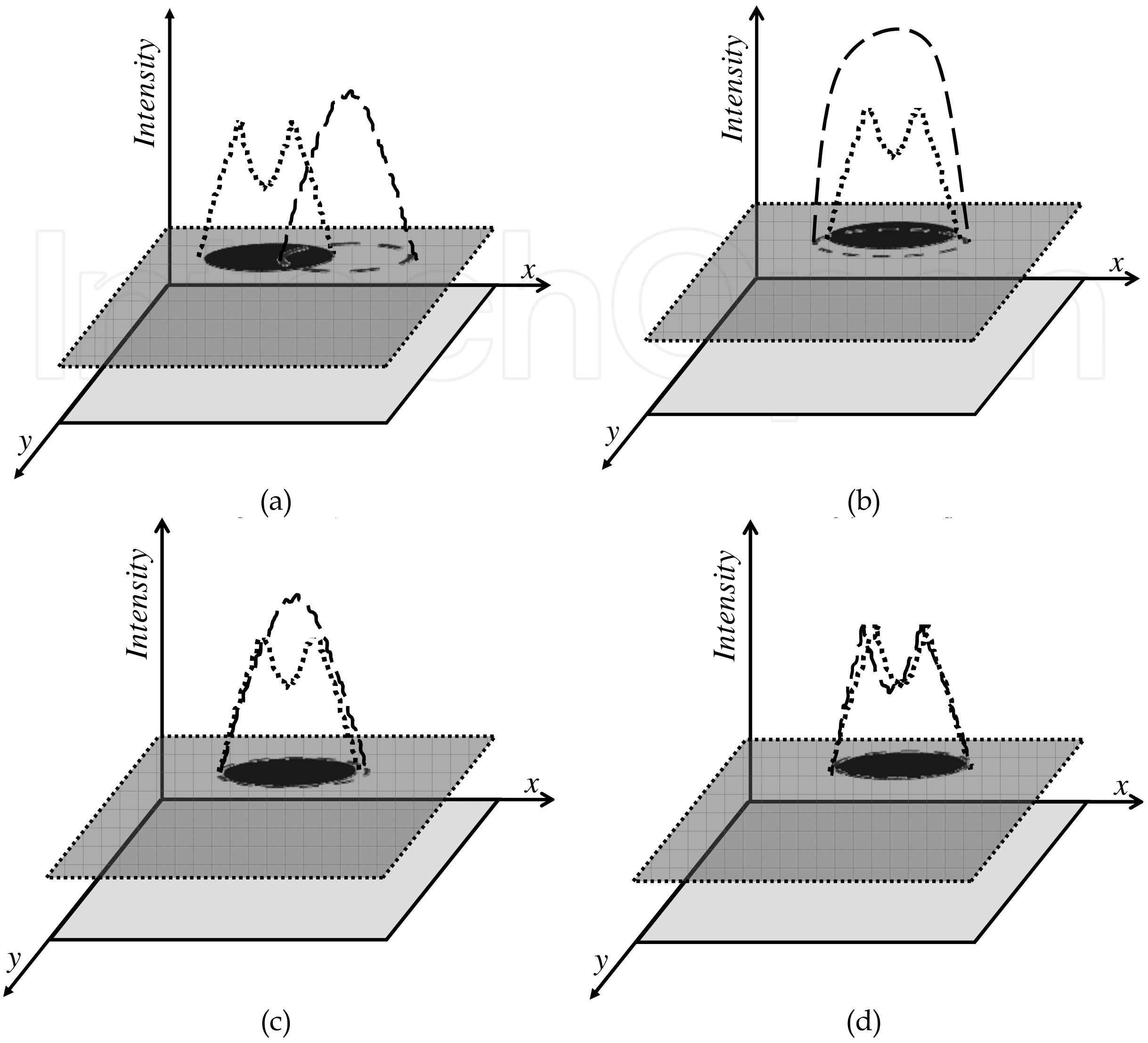
$$F(m) = \begin{cases} R_{MB}(m),\ if\ Min\left[f_1(m), f_2(m), R_{MB}(m)\right] \leq Th_R \\ 1 + R_{MB}(m),\ else if\ f_3(m) \leq 0 \wedge f_4(m) \leq 0 \\ R_{MODEL}(m) \cdot F_{Overlap}(m),\ otherwise \end{cases} \tag{23}$$

The Fitness Function *F*(*m*) of a Chromosome *m* equals to $R_{MB}(m)$ (1st case), to $1 + R_{MB}(m)$ (2nd case) or to $R_{MODEL}(m) \cdot F_{Overlap}(m)$ (3rd case), according to the values of $f_1(m)$, $f_2(m)$, $R_{MB}(m)$, $f_3(m)$ and, $f_4(m)$.

If one of the value of $f_1(m)$, $f_2(m)$ and, $R_{MB}(m)$ is less or equal to a threshold Th_R, it means that the model-compartment I_{MB}^{m} does not resemble at all to the real-compartment I_{REAL} (1st case). Th_R is a threshold which controls the minimum acceptable resemblance of the model-compartment I_{MB}^{m} with the real-compartment I_{REAL}.

If the values of $f_1(m)$, $f_2(m)$ and, $R_{MB}(m)$ are higher than the threshold Th_R, it means that the model compartment I_{MB}^{m} resembles to an extent to the real-compartment I_{REAL}. In this case, the fitness function checks the value of $f_3(m)$ and $f_4(m)$. If their values are less than zero, the model compartment I_{MODEL}^{m} does not resemble at all to the real-compartment I_{REAL}, thus the model-compartment is not an acceptable one (2st case). On the other hand, if their values are higher than zero, it means that the model compartment I_{MODEL}^{m}, represented by the chromosome *m*, represents to a degree the real compartment (3rd case). Of course, the higher the value of $R_{MODEL}(m) \cdot F_{Overlap}(m)$ is, the higher the resemblance between the real compartment with the model compartment I_{MODEL}^{m} becomes.

Using the fitness function *F*(*m*), the higher the resemblance of the model-compartment I_{MODEL}^{m} with the real one is, the higher the value of the fitness function *F*(*m*) becomes. This is because the genetic algorithm can assign to the chromosome *m* of the 3rd case a higher fitness value than to the one of the 2nd case and to the one of the 1st case. For example, the genetic algorithm can progressively assign – from upper left to lower right – a higher fitness value to the chromosomes representing the compartment-models in Fig. 10.



(a) (b) (c) (d)

Fig. 10. Overlapping of the real-compartment I_{REAL} and compartment-model I^{m}_{MODEL} . The dotted curve represents the real spot S_{REAL}, while the dashed curve represents the spot-model S^{m}_{MODEL} . The chromosome m representing the I^{m}_{MODEL} in (d) should have progressively higher fitness value than that of (c), (b) and (a).

4.3.3 Evolutionary circle-termination criteria

Let Pop_n be a population of chromosomes, where n stands for the consecutive number of populations. The population consists of N_{pop} chromosomes. A new population Pop_{n+1} of an equal number of chromosomes (N_{pop}) is created through the following stages: (i) Reproduction stage: P_r% of the best chromosomes of the current population Pop_n are carried over to the new population Pop_{n+1}. (ii) Crossover-Mutation stage: The chromosomes needed to complete the new population Pop_{n+1} are produced through iterations of the following: Four chromosomes of the population Pop_n are selected using the tournament selection method (Miller et al., 1995); These chromosomes are subsequently subjected in groups to a crossover operator (according to a P_c% probability) and then to a mutation operator (according to a P_m% probability). The best two of the four resulting chromosomes (those two with the best fitness value) proceed to the new population Pop_{n+1}. It should be noted that the mutation operator applied is the wavelet-mutation as it exhibits a fine-tune ability as opposed to other mutation operator (Ling et al., 2007). Moreover, the crossover operator applied is the joint application of the BLX-a and the dynamic heuristic crossover as it is the most promising crossover application (Herrera et al., 2005).

New populations are thus produced until at least one of the following two criteria is satisfied: (i) the genetic algorithm is executed up to a maximum number of populations G_{Max}; (ii) the genetic algorithm is executed up to a maximum number of populations G_{Fit} for which the best fitness value has remained unchanged.

5. Results

Several experiments were executed so as to evaluate the performance of the proposed method for spot-segmentation on both synthetic and real cDNA microarray images. Most of the parameters were experimentally adjusted once, and thus they remained unchanged during all the experiments. The constant k of 0.6 was adopted as the most appropriate in order to distinguish: i) the pixels which belong to the background area, ii) the pixels which belong to the spot area, and iii) the pixels which have a 'membership degree' μ_B in order to belong to the background area and a 'membership degree' μ_S in order to belong to the spot area. The constant parameters λ_m of 0.7 and λ_d of 0.2 were adopted as the most appropriate so as to control the $I_{MR}(p)$ value of each pixel p of the real microarray compartment ((eq. 12) and (eq. 13)). A constant E_{MAX} of 0.2 was adopted so as to control the maximum acceptable error of a model-compartment I_C^m at a specific pixel. A threshold Th_R of 0.15 was adopted as the most appropriate one in order to define the minimum acceptable resemblance between a model compartment I_{MB}^m and the real one.

The population size of the genetic algorithm N_{pop} was set to 100. This size is high enough to reduce the possibility of the genetic algorithm prematurely converging to a local solution that would not be an efficient one. Meanwhile, it does not increase the time required for the population to converge to an efficient solution (Achiche et al., 2004). The percentage of each population which was reproduced was relatively small (P_r=10%) as the reproduction was used only for the best chromosomes of the population to be preserved in the next population. In accordance with Miller et al (Miller et al., 2003) the high crossover probability of 80% was chosen (P_c=80%). The mutation probability was experimentally adjusted to 30% (P_m=30%). The termination criterion was satisfied when the genetic algorithm was executed for 500 populations (G_{Max}=1000) or when the best fitness value remained unchanged for 250 populations (G_{Fit}=250).

5.1 Evaluation of the performance using synthetic microarray images

In order to compare the proposed method with preexisting ones objectively, we used an existing dataset of synthetic microarray images for which the ground truth is known. The dataset contains 50 good quality images and 50 low quality images. Each image has been produced by the microarray simulator of Nykter et al (Nykter et al., 2006). It is digitized at 330 x 750 pixels and it contains 1000 spots. Nykter's simulator has been designed to produce synthetic microarray images with realistic characteristics. Consequently, the good quality images have low variability in spot sizes and shapes, while the noise level is reasonable low. On the contrary, the low quality images contain spots whose shape and size vary significantly. In addition, noise level is significantly higher for the low quality images. It should be noted that this dataset has already been used for the evaluation of other well-known segmentation techniques (see table I), as it is described in (Lehmussola et al., 2006).

During these experiments, the efficiency of the proposed method was analyzed by means of a statistical analysis. The statistical analysis is the one described in (Lehmussola et al., 2006). More precisely, the segmentation accuracy was measured on a pixel level. Firstly, the probability of error *PE* and the discrepancy distance *D* for each synthetic spot were calculated. Then, the median probability of error and the median discrepancy distance for both good and low quality images were calculated.

The probability of error PE measures the mis-segmented pixels. It is defined as:

$$PE = P(F)\cdot P(B\,|\,F) + P(B)\cdot P(F\,|\,B) \tag{24}$$

where $P(F)$ and $P(B)$ are the a priori probabilities of foreground and background. $P(F\,|\,B)$ denotes the probability of error in classifying background as foreground, while $P(B\,|\,F)$ denotes the probability of error in classifying foreground as background.

The discrepancy distance *D* gives different weights for mis-segmented pixels based on how spatially far they are located from the nearest correct segmentation result.

$$D = \frac{\sqrt{\sum_{i=1}^{N} d^2(i)}}{A} \tag{25}$$

where *N* is the number of mis-segmented pixels, *d*(*i*) is the Euclidian distance from the *i*-th mis-segmented pixels to the nearest pixel that actually belongs to the mis-segmented class. *A* is the number of pixels in the image.

Algorithm	Median Value of Probability of error		Median Value of Discrepancy distance	
	Good	Low	Good	Low
Fixed Circle	0.049	0.049	0.027	0.027
Adaptive Circle	0.019	0.192	0.017	0.074
Seeded region growing	0.099	0.114	0.037	0.048
Mann-Whitney	0.165	0.162	0.066	0.074
Hybrid k-means	0.017	0.020	0.016	0.029
Markov random field	0.154	0.053	0.063	0.039
Matarray	0.004	0.031	0.008	0.068
Model-based segmentation	0.094	0.101	0.052	0.067
Proposed method	**0.000**	**0.012**	**0.000**	**0.018**

Table 1. Performance of commonly used and well-known segmentation algorithms as well as of the proposed method

The evaluation results of the proposed method are shown in table I (last row). It becomes obvious that the proposed method can accurately segment the spots of good quality images while it can segment the spots of low quality images quite efficiently. In the same table we have apposed the results of commonly used and well-known segmentation techniques (first eight rows), as they are reported by Lehmussola et al. Comparing the results of the proposed method with the results of the other software programs, it is obvious that the results of the proposed method are significantly more successful than the ones of the other software programs, indicating the high performance of the proposed method. The significant number of spots which are contained in the used dataset supports these arguments additionally. Indeed, the evaluation of all the methods has been statistically calculated in 50000 artificial microarray spots for which the ground truth is given, meaning that the correct segmentation result is known.

Fig. 11 presents a segmentation result on two blocks of a good quality and a low quality synthetic microarray image. As it is obvious the proposed algorithm has very efficiently segmented the microarray spots.

5.2 Evaluation of the performance using real microarray images

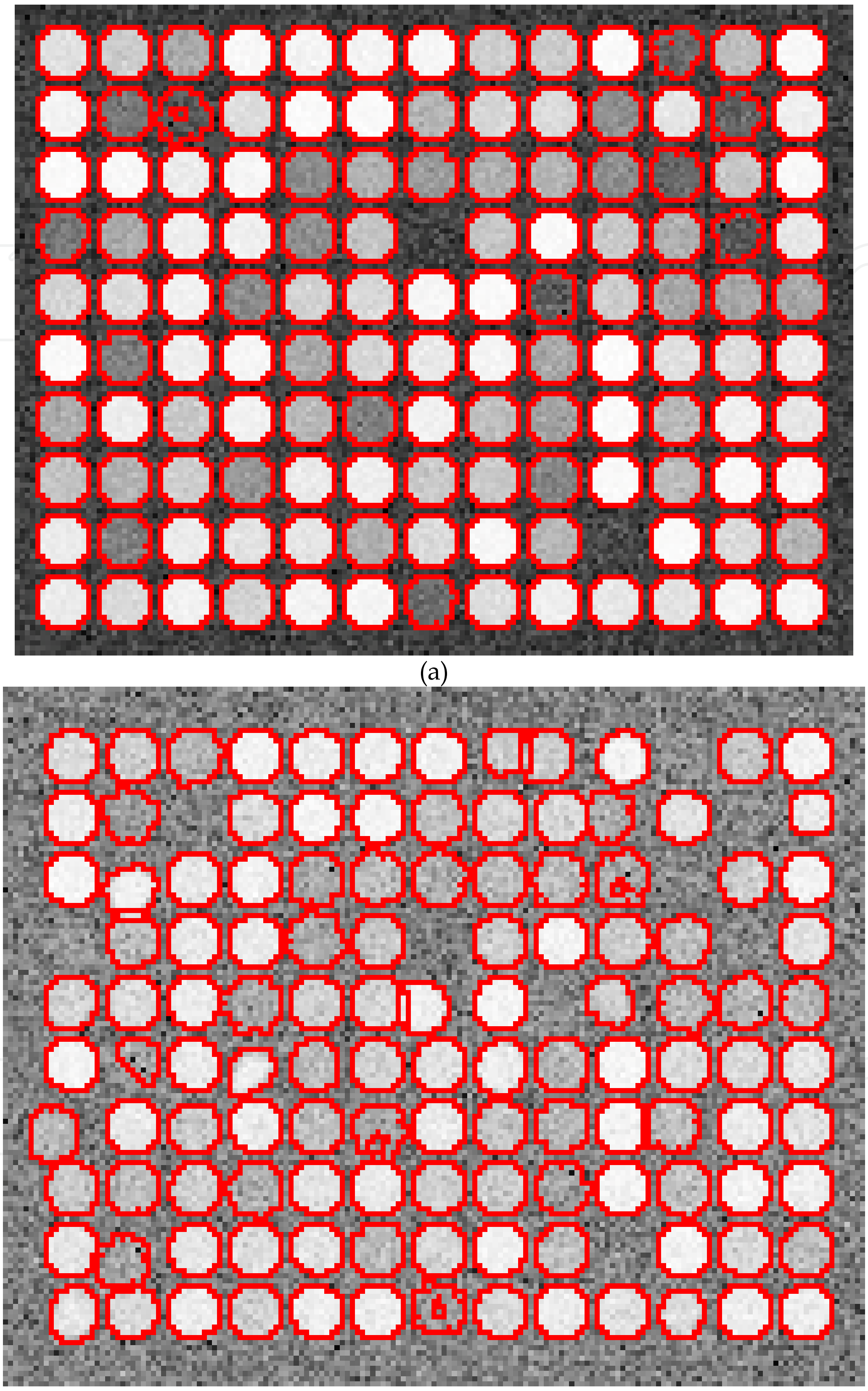
The second dataset contains ten microarray blocks, which have been arbitrarily selected from ten microarray images obtained from the Stanford Microarray Database (SMD) (Standford Microarray Database), which is publicly available. The blocks are digitized at ~ 450 x 450 pixels at 16-bit grey level depth and they are stored in tiff format. Each one of them contains 864 spots. Thus, the blocks contain 8640 spots in total. The microarray images have been produced by comprehensively analyzing the gene expression profiles in 54 specimens of acute lymphoblastic leukemia, 37 positive and 17 negative to BCR-ABL. BCR-ABL is a fusion gene product resulting from translocation between the 9th and the 22nd chromosomes.

Fig. 12 shows the segmentation results of a real-microarray sub-image which is contaminated with noise and contains the three types of microarray spots (peaked-shaped spots, volcano and doughnut-shaped spots). As it is obvious, the proposed method has very efficiently segmented the spots. Moreover, the proposed method has correctly detected the absence of the first spot.

6. Conclusions

Spot-segmentation in microarray images comprises one of the most challenging stages of the microarray image analysis sequence. In this chapter, the segmentation procedure is a result of an optimization problem which is tackled by using a genetic algorithm, which represents, in a three dimensional space, the real-spots of the microarray image with spot-models. In vue of this, fuzzy logic is adopted in order to take into account the uncertainties existing in the pixels' intensities, and which have been caused by noise, artifacts, and uneven background. The segmentation of the real-spots is conducted by drawing the contours of the spot-models.

The proposed approach is noise-resistant and it is efficient under the following adverse conditions: i) the appearance of various spot-shapes, such as peak-shaped spots, volcano-shaped spots and doughnut-shaped spots, ii) the appearance of spots of diverse intensities, such as low intensity spots which are not clearly visible or saturated spots and iii) the appearance of various spot-sizes. Last but not least, it is fully-automatic since it does not require any input parameter or human intervention in order to determine the contours of microarray spots properly. The experimental results over synthetic and real images demonstrate that it is very efficient and effective. Furthermore, it outperforms various existing well-known and broadly used segmentation techniques.



(a)

(b)

Fig. 11. Spot-segmentation result of 2 blocks: a) in a good quality artificial microarray image and, b) in a low quality artificial microarray image.

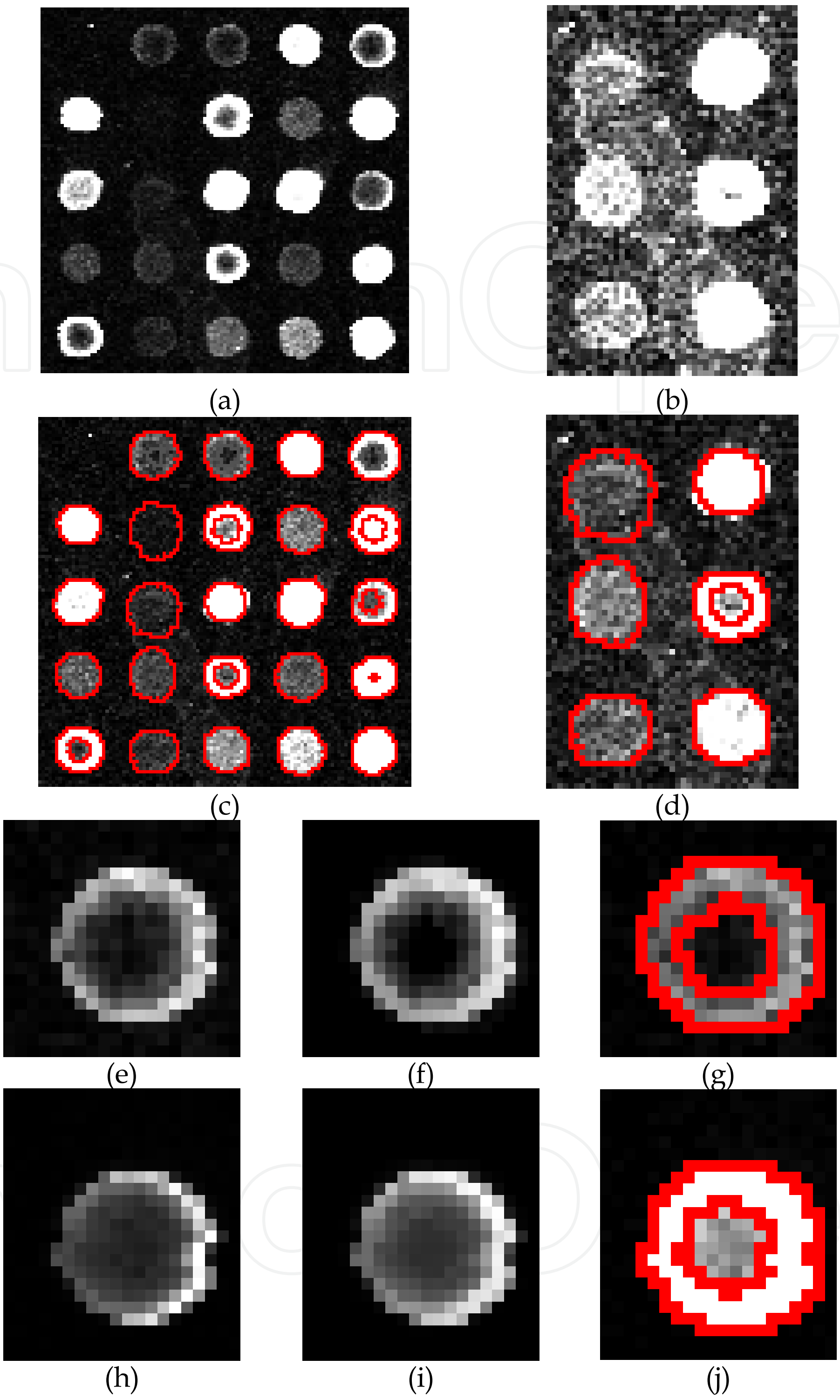


Fig. 12. Spot-segmentation results: (a) A region of a real microarray image containing the tree types of microarray spots in the presence of noise, (b) enlargement of a part of the microarray image which is contaminated with noise, (c,d) the spot-segmentation results of (a,b), (e) enlargement of a doughnut-shaped spot (f) the spot-model representing the doughnut-shaped spot, (g) the segmentation result of the doughnut-shaped spot, (h) enlargement of a volcano-shaped spot (i) the spot-model representing the volcano-shaped spot, and (j) the segmentation result of the volcano-shaped spot.

7. References

Achiche, S. Baron, L. & Balazinski, M. (2004). "Real/binary-like coded versus binary coded genetic algorithms to automatically generate fuzzy knowledge bases: a comparative study," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 4, pp. 313-325, 2004.

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

Buckley M.J. (2000). The Spot User's Guide, CSIRO mathematical and information sciences. http://www.cmis.csiro.au/IAP/Spot/ spotmanual.htm

Buhler, J. ; Ideker, T. ; & Haynor, D. "Dapple: improved techniques for finding spots on DNA microarrays," *UW CSE Technical Report UWTR 2000-08-05*, pp. 1-12, Aug. 2000.

Campbell, A.M. & Heyer L.J. (2007). " Discovering Genomics, Proteomics & Bioinformatics," 2nd ed. , Pearson Benjamin Cummings, 2007, pp. 233-238.

Chen, Y.; Dougherty, E.R. &. Bittner, M.L. (1997). "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal Biomedical Optics*, vol. 2 no. 4, pp. 364-374, 1997.

Chen, W.B.; Zhang, C. & Liu, W.L. (2006). "An Automated Gridding and Segmentation Method for cDNA Microarray Image Analysis," in *Proc. 19th IEEE Symp. Computer-Based Medical Systems*, Salt Lake City, 2006, pp. 893-898.

Eisen. M.B. (1999). ScanAlyze. [Online]. Available: http://rana.lbl.gov/EisenSoftware.htm

Demirkaya, O. ; Asyali, M.H. & Shoukri, M.M. (2005). "Segmentation of cDNA microarray spots using markov random field modeling," *Bioinformatics*, vol. 21, no. 13, pp. 2994–3000, Apr. 2005.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization & Machine Learning, Boston: Addison-Wesley, Reading, 1989, ch. 1.

Hayes, C.S.M. (2006). "Generic Properties of the Infinite Population Genetic Algorithm," Ph.D. dissertation, Dept. Mathematics, Montana State Univ., Bozeman, Montana, USA, 2006.

Herrera, F.; Lozano, M.; & Verdegay, J.L. (1998). "Tackling Real Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, Nov. 1998.

Herrera, F.; Lozano, M. & Sanchez, A.M. (2005). "Hybrid crossover operators for real-coded genetic algorithms: An experimental study," *Soft Computing*, vol. 9, no. 4, pp. 280-298, Apr. 2005.

Ho, J. & Hwang, W.L. (2008). "Automatic Microarray Spot Segmentation using a snake-fisher model," IEEE Trans. on Medical Imaging, vol. 27, no. 6, pp. 847-857, Jun. 2008.

Kim H.Y. et al. (2007). "Characterization and simulation of cDNA microarray spots using a novel mathematical model," *BMC Bioinformatics*, vol. 8, pp. 485-496, March 2007.

Lehmussola, A.; Ruusuvuori, P.; & Yli-Harja, O. (2006). "Evaluating the performance of microarray segmentation algorithms," Bioinformatics, vol. 22, no. 23, pp. 2910-2917, Oct. 2006.

Leung, Y.F. & Cavalieri, D. (2003). "Fundamentals of cDNA microarray data analysis," *Trends in Genetics*, vol. 19, no. 11, pp. 649-659, Nov. 2003.

Ling, S.H. & Leung, F.H.F. (2007). "An improved genetic algorithm with average-bound crossover and wavelet mutation operations," *Soft Computing*, vol. 11, no. 1, pp. 7-31, 2007.

Li, Q. ; Fraley, C. ; Bumgarner, R.E. ; Yeung, K.Y. & Raftery, A.E. (2005). "Donuts, scratches and blanks: robust model-based segmentation of microarray images," *Bioinformatics*, vol. 21, no. 12, pp. 2875-2882, Apr. 2005.

Miller, B.L. & Goldberg, D.E. (1995). "Genetic Algorithms, Tournament selection, and the Effects of Noise," *Complex Systems*, vol. 9, no. 3, pp. 193-212, 1995.

Miller, M.T. Jerebko, A.K. Malley, J.D. & Summers, R.M. (2003). "Feature selection for computer-aided polyp detection using genetic algorithms," in *Proc. of SPIE*, Santa Clara, 2003, pp. 102-110.

Nagarajan, R. (2003). "Intensity-based segmentation of microarray images," *IEEE Trans. on Medical Imaging*, vol. 22, no. 7, pp. 882–889, Jul. 2003.

Rahnenfuhrer, J. & Bozinov, D. (2003). "Hybrid clustering for microarray image analysis combining intensity and shape features," *BMC Bioinformatics*, vol. 5, no. 5, p. 47-58, Apr. 2004.

Nykter, M. et al. (2006). "Simulation of microarray data with realistic characteristics," BMC *Bioinformatics*, vol. 7, pp. 349-366, Jul. 2006.

Srinark, T. & Kambhamettu, C. (2004). "A Microarray Image Analysis System Based on Multiple Snakes," *Journal of Biological Systems*, vol. 12, no. 25, Jun. 2004.

Srinark, T. & Kambhamettu, C. (2001). "A framework for Multiple Snakes," in *Proc. Comp. Vision and Pattern Recogn.*, vol. 2, pp. 202-209, 2001.

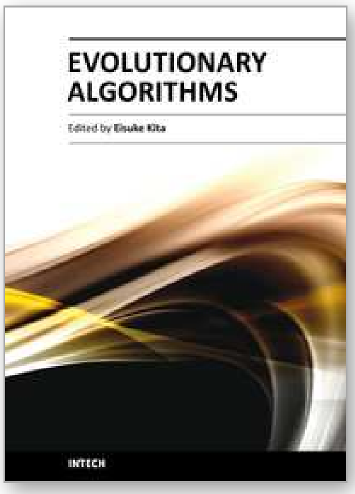
Standford Microarray Database. [Online]. Available: http://genome-www5.stanford.edu/

Tu, Y.; Stolovltzky, G.; & Kleln, U. (2002). "Quantitative noise analysis for gene expression microarray experiments," in *Proc. of National Academy of Sciences of the United States of America (PNAS)*, vol. 99, no.22, pp. 14031–14036, Oct. 2002.

Wang, X.H. & Istepanian, R.S.H. (2003). "Microarray image enhancement by denoising using stationary wavelet transform," *IEEE Trans. on Nanobioscience*, vol. 2, no. 4, pp 184-189, Dec. 2003.

Wang, X.; Ghosh, S.; & Guo, S-W. (2001). "Quantitative quality control in microarray image processing and data acquisition," *Nucleid Acids Research*, vol. 29, no. 15, 2001.

Yang, Y. H.; Buckley, M. J.; Dudoit, S. & Speed, T. (2002) "Comparison of methods for image analysis on cDNA microarray data," *J. of Comp. & Graphical Statistics*, vol. 11, no. 1, pp. 108–136, 2002.



Evolutionary Algorithms
Edited by Prof. Eisuke Kita

ISBN 978-953-307-171-8
Hard cover, 584 pages
Publisher InTech
Published online 26, April, 2011
Published in print edition April, 2011

Evolutionary algorithms are successively applied to wide optimization problems in the engineering, marketing, operations research, and social science, such as include scheduling, genetics, material selection, structural design and so on. Apart from mathematical optimization problems, evolutionary algorithms have also been used as an experimental framework within biological evolution and natural selection in the field of artificial life.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821