CHAPTER 4.6

ADVANCED EVOLUTIONARY ALGORITHMS FOR INTELLIGENT MICROARRAY IMAGE ANALYSIS

Eleni Zacharia and Dimitris Maroulis

Dept. of Informatics and Telecommunications,
University of Athens GR 15784, Ilisia, Greece
E-mail: eezacharia@gmail.com, dmaroulis@di.uoa.gr

cDNA microarrays, one of the most fundamental and powerful biotechnology tools, is being utilized in a variety of biomedical applications as it enables scientists to simultaneously analyze the expression levels of thousands of genes over different samples. One of the most essential processes of cDNA microarray experiments is the image analysis one, which is divided into three phases, namely, gridding, spot-segmentation and intensity extraction. Although its two former phases appear relatively straightforward, they are in fact rather challenging procedures due to the nature of microarray images. For their implementation, the currently available software programs require human intervention, which significantly affects the biological conclusions reached during microarray experiments. In this chapter, the basic process of analyzing a microarray image is described and advanced evolutionary algorithms implementing the automatic gridding and segmentation processes are presented. In reality, both of these algorithms are based on optimization problems which are solved by using evolutionary genetic algorithms. Contrary to existing software systems, the proposed methods are fully automatic as they do not require any human intervention; they are also noise resistant and yield excellent results even under adverse conditions. Last but not least, they outperform other software programs as well as established techniques.

1. Introduction

In the last decade, microarray technology is being increasingly applied in numerous fields of biomedical research such as cancer research, pharmacology research, toxicology research, infectious disease diagnosis and treatment, and agricultural development.[1] The reason for its broad use and success can be

attributed to its main revolutionary feature: the ability to simultaneously analyze the expression levels of thousands of genes over different samples.

The process of a microarray experiment[2] 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 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.

Ideally, the existing spots inside the microarray image have a circular 2D shape with fixed diameters, while their intensity peaks at their central region and declines at regions further from their centre. Moreover, they are aligned in 2D array layouts, namely blocks. Likewise, all blocks inside the microarray image are arranged in a 2D array layout too and each contains an equal number of spots. As a result, adjacent blocks and adjacent spots are clearly separated inside the microarray image. However, the edge-to-edge distance between two adjacent blocks is larger than the distance between two adjacent spots within the same block (Fig. 1).

As stated by Yang et al,[3] the process of analyzing a microarray image can be divided into three main phases namely: "Gridding", "Spot-Segmentation" and

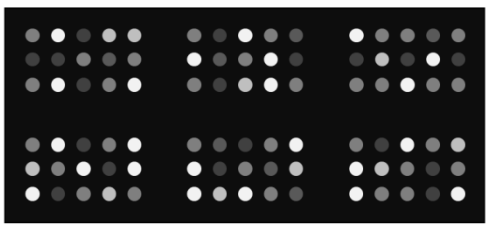


Fig. 1. An illustration of a synthetic microarray image: This image is composed of 6 blocks in a 3x2 layout. Each block contains 15 spots which are located in a 5x3 layout.

"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 result from their individual brightness.

There is a considerable number of software systems and techniques that have been developed and proposed to date in order to analyze the microarray images. However, gridding microarray images and segmenting microarray spots remain two rather challenging and complicated procedures, and therefore they still require human intervention. This is attributed to the nature of microarray images. More precisely, the quality of images is often degraded due to the existence of noise and/or artifacts as well as due to uneven background.[4] Furthermore, microarray images may contain low-intensity spots that are so poorly contrasted that are not clearly visible.[5] Furthermore, there may be rotations, misalignments and local deformations of the ideal rectangular grid[6] as spots are not ideally aligned in a 2D array layout. In addition, 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.

Examples of software systems which are frequently used for analyzing microarray images include the ScanAlyze,[7] Dapple,[8] ImageGene,[9] and Spot.[10] Most of these software programs perform the gridding procedure by enabling manual template matching. In particular, the user defines the 'optimal' template – by specifying a number of parameters – with the prospect of maximizing the inclusion of spot pixels inside their corresponding compartments of the template. Moreover, most of them perform the segmentation procedure by using either the circle segmentation algorithm (fixed or adaptive), or the adaptive shape-segmentation algorithm. The former algorithm assumes that microarray spots have a circular shape while the latter algorithm can segment regions of irregular shapes by implementing a watershed algorithm.

Apart from these software systems, a considerable number of techniques have been proposed to date in order to analyze microarray images. Some well-known approaches to gridding microarray images are based on axis projections,[11] or on morphological filtering.[12,13,14] For example, Liew et al[12] has estimated the grid from some guide-spots, which are detected by using adaptive thresholding and morphological processing steps. Other gridding methods use graph models[15] in which spots are represented as ε-graphs, with up, down, left, and right edges. There are also many gridding methods trying to solve rotation and misalignment problems,[16-20] which improve the accuracy results of the previous ones. Several

methods have been also developed to segment microarrays spots. For instance, Chen et al.[21] suggested a threshold method based on the statistical Mann-Whitney test which relies on the appropriate choice of background samples. Clustering algorithms, such as K-means, hybrid K-means and, fuzzy C-means (FCM) have been additionally applied in order to determine which pixels belong to the spot area and which ones to the background area.[22-25] Another segmentation method, based on the clustering of pixels' values, is the model-based segmentation algorithm, proposed by Li et al.[26] This method uses a removal technique of the components that are spatially connected in order to exclude small disconnected clusters which are assumed to be artifacts.

All aforementioned techniques require human intervention in order to define input parameters or to correct the gridding or 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. As a result, the automation of the gridding procedure as well as the automation of the spot-segmentation procedure remains an important issue to resolve.

In this chapter, accurate algorithms implementing the automatic gridding and segmentation processes are presented. Both these algorithms are based on the optimization technique of evolutionary algorithms and particularly of genetic algorithms. Consequently, prior to describing the two methods, a brief introduction of the evolutionary genetic algorithms' technique is apposed. The remainder of this chapter is organized in five sections: In section 2 a brief description of the evolutionary genetic algorithms' optimization technique is provided. In Sections 3 and 4, advanced evolutionary genetic algorithms are presented for the gridding and segmentation phase respectively. Section 5 illustrates experiments, while in section 6 conclusions are apposed.

2. Evolutionary Genetic Algorithms' Optimization Technique

Genetic algorithm is a powerful optimization search methodology based on the principles of natural selection and evolution.[27] Compared to conventional heuristic methods, it is a promising alternative as it exhibits significant advantages: it can optimize a large number of variables of extremely complex cost surfaces and solve problems for which there is little or no *a priori* knowledge of the underlying processes.

A conventional genetic algorithm[28] begins its search by constructing a finite number of potential solutions encoded as alpha-numerical sequences called chromosomes. These chromosomes, which constitute an initial population Pop_1,

are evaluated using a fitness function. Subsequently, the population Pop_1 evolves into a new population Pop_2 using the following three genetic operators: reproduction, crossover, and mutation. This evolutionary cycle of a current population Pop_n to its next Pop_{n+1} (where n stands for the consecutive number of populations), continues until a specific termination criterion is satisfied. In conclusion, the following four basic elements are deemed necessary for the determination of a particular problem: chromosome representation, chromosome evaluation, evolutionary cycle, and termination criteria.

In the following sections, the evolutionary genetic algorithms used in gridding and spot-segmenting phases enclose the same evolutionary cycle and termination criteria. More precisely, a new population Pop_{n+1} is created from the current Pop_n by applying 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} ,and (ii) Crossover-Mutation stage: The chromosomes needed to complete the new population Pop_{n+1} are produced through iterations; Four chromosomes of the population Pop_n are selected using the tournament selection method.[29] These chromosomes are subsequently subjected in turn to the joint application of the BLX-a,[30] and the dynamic heuristic crossover operator,[30] according to a P_c% probability and then to the wavelet mutation operator,[31] according to a P_m% probability. The best two of the four resulting chromosomes proceed to the new population Pop_{n+1}. Moreover, the genetic algorithms are executed up to a maximum number of populations G_{Max}, or up to a maximum number of populations G_{Fit} for which the best fitness value has remained unchanged.

3. Gridding Microarray Images

Normally, gridding a microarray image is a two-stage procedure: Firstly, the microarray image is cut into distinct, quadrilateral in shape, blocks. Subsequently, each of these blocks is cut into numerous, quadrilateral in shape, single-spot compartments (Fig. 2). All of the above procedures are accomplished by the determination of the line-segments constituting the borders of blocks (1^{st} stage) or spots (2^{nd} stage).

Based on the aforementioned remark, our gridding method utilizes two similar stages; its 1^{st} stage determines the borders of each block, while its 2^{nd} one determines the borders of each spot inside the block. Let G be a quadrilateral which represents a microarray image or block. Each of the aforementioned stages include the determination of the line-segments whose members are defined by the two horizontal sides of G (Fig. 3a, "*BC*" and "*AD*" sides), and the

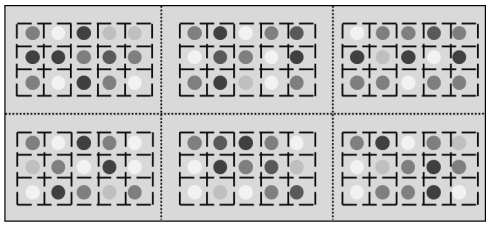


Fig. 2. An illustration of a gridding result of a synthetic microarray image.

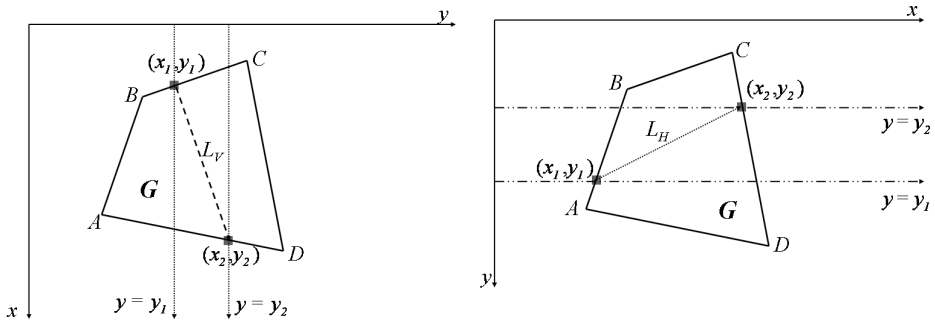


Fig. 3. Illustrations of the quadrilateral in shape *G*: Cartesian systems for the representation of the line-segments belonging to the (a) S_V set and (b) S_H set.

determination of the line-segments whose members are defined by the two vertical sides of *G* (Fig. 3b, "*BA*" and "*CD*" sides). Let S_V be the former set of line-segments, while S_H be the latter set of line-segments. The determination of each of the S_V or S_H sets of line-segments is expressed as an optimization problem which is tackled by using an evolutionary genetic algorithm.

3.1. *Attributes of the line-segments contained to the S_V or S_H sets*

Any line-segment on a 2D Cartesian plane can be defined by its end-points using the following equation:

$$y = y_1 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_1), \tag{1}$$

where (x_1, y_1), (x_2, y_2) denote the coordinates of the two end-points of the line-segment, and $x \in [x_1, x_2]$ is the independent variable of the equation.

In the case of $x_1 = x_2$, the aforementioned equation (Eq. 1) is not valid. Consequently, in order for any line-segment L_i belonging to the S_V or S_H set to be

able to be algebraically described by Eq. 1, the 2D Cartesian plane of Fig. 3a or Fig. 3b respectively is utilized.

Moreover, the line-segments L_i belonging to the S_V or S_H sets intersect with the two sides of the quadrilateral in shape G. Thereby, the x-coordinates (x_1 and x_2) of their end-points can be computed from the y-coordinates (y_1 and y_2). For instance, the x-coordinates of the line-segment "L_V" belonging to the S_V set (Fig. 3a) can be computed given that: (i) the intersection point of the line $y = y_1$ and the line-segment ("*BC*") is the point (x_1, y_1), and (ii) the intersection of the line $y = y_2$ and the line-segment ("*AD*") is the point (x_2, y_2). Likewise, the x-coordinates of the line-segment "L_H" (Fig. 3b) belonging to the S_H set can be also computed in the same manner.

3.2. *Genetic Algorithm for Gridding Microarray Images*

3.2.1. Chromosome Representation

The chromosome m represents all line-segments L_i, i=1,…, $N(m)$ belonging either to the S_V set or the S_H set. Due to the nature of the alignment of blocks inside the microarray image and the arrangement of spots inside the blocks, the chromosome encodes a line-segment (belonging to the S_V set or the S_H set) and the distance d between adjacent line-segments of the corresponding set, instead of encoding each line-segment. More precisely, the chromosome m has been encoded as a string of three real values (Fig. 4); the two y-coordinates (y_1 and y_2) of the end-points of a line-segment and the distance d between two adjacent line-segments. The y-coordinates (y_1 and y_2) are located on the horizontal sides ("*BC*" and "*AD*") or the vertical sides ("*BA*" and "*CD*") of the quadrilateral in shape G.

Fig. 5 depicts a quadrilateral in shape G as well as the line-segments constituting the borders between adjacent blocks or spots. In the case when the genetic algorithm searches for the exact values of the variables of the optimal line-segments belonging to the S_V set, its chromosome will encode the y-coordinates of the end-points of L_1 and the distance d_V. Respectively, for the S_H set, its chromosome will encode the y-coordinates of the end-points of L_4 and the distance d_H .

Variables of a line-segment (y-coordinates)		Distance between adjacent line-segments
y_1	y_2	d

Fig. 4. A schematic illustration of chromosome's encoding.

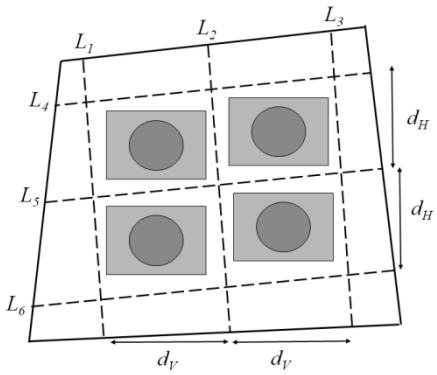


Fig. 5. Illustration of the line-segments constituting the grid structure of a quadrilateral in shape *G*.

3.2.2. Chromosome Evaluation

It is self-explanatory that a line-segment which is part of the grid is located in the background area existing between adjacent blocks or spots. The pixels of this area have intensities which are generally lower than the intensities of the pixels constituting spots. As a result, the probability $P(L_i)$ of a line-segment L_i to be part of the grid is defined by the following equation:

$$P(L_i) = f_B^{Ri}(L_i) - f_S^{Ri}(L_i)\,, \tag{2}$$

where R_i denotes the region of G which contains those pixels whose distance from the line-segment L_i is less than a margin w (Fig. 6). The real-valued function $f_B^{Ri}(L_i)$ expresses the percentage of pixels of the region R_i whose intensity is lower than a value I_B (background pixels), while the real-valued function $f_S^{Ri}(L_i)$ expresses the percentage of pixels of the region R_i whose intensity is higher than the value I_B (spot pixels). I_B is an intensity value which is

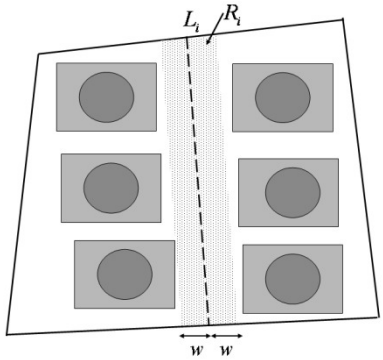


Fig. 6. Illustration of the region R_i (dotted gray area) near the dashed line-segment L_i.

defined as the value which is present in most pixels of G. Any pixel with intensity value lower than I_B is considered to belong to the background.

The Fitness Function $F_G(m)$ of a chromosome m that encodes a possible solution to the particular optimization problem is defined by the following equation:

$$F_G(m) = \begin{cases} S_p(m) \cdot N(m), \; if \; m \equiv efficient \\ S_p(m), otherwise \end{cases} \tag{3}$$

where the real-valued function $S_p(m)$ denotes a total sum of the probabilities $P(L_i)$ of the line-segments L_i, i=1,…,$N(m)$, that are represented by the chromosome m, and have a higher than a threshold P_{MAX} probability $P(L_i)$ to be part of the grid. P_{MAX} is a threshold which expresses the minimum acceptable probability of a line-segment to be part of the grid. Therefore, it controls which of the line-segments L_i participate in the sum $S_p(m)$. More precisely, $S_p(m)$ is defined as:

$$S_p(m) = \sum_{i=1}^{N(m)} P(L_i) \cdot q_i \; , \tag{4}$$

where

$$q_i = \begin{cases} 1, \; if \; P(L_i) > P_{MAX} \\ 0, \; otherwise \end{cases} \tag{5}$$

A chromosome m is efficient if most of the line-segments that it represents are well-defined, by being located in background areas. Therefore, we define the percentage $f_{LS}(m)$ of the line-segments L_i, i=1,…,$N(m)$, that are represented by the chromosome m, and have a lower than or equal to a threshold P_{Low} probability $P(L_i)$ to be part of the grid:

$$f_{LS}(m) = \frac{\sum_{i=1}^{N(m)} k_i}{N(m)}, \tag{6}$$

where

$$k_i = \begin{cases} 1, \; if \; P(L_i) \le P_{Low} \\ 0, \; otherwise \end{cases} \tag{7}$$

If the $f_{LS}(m)$ percentage of the chromosome m is lower or equal to a threshold f_{Max}, it means that the line-segments, represented by the chromosome m, are well-defined because they are located in background areas, some of which may be

contaminated with noise. Therefore the chromosome m is efficient. Otherwise, the chromosome m is inefficient.

Using the Fitness Function $F_G(m)$, the genetic algorithm can assign to an efficient chromosome m a higher fitness value than to an inefficient one. Moreover, the multiplication $S_p(m)\cdot N(m)$ in Eq. 3 prevents the genetic algorithm from converging to a local solution that would be not an optimal one (this would lead to termination without determining all the line-segments belonging to the S_H or the S_V sets). Indeed, the higher the fitness value of the chromosome is, the greater number $N(m)$ of the line-segments L_i which have a high probability $P(L_i)$ to be part of the grid it represents.

3.3. *The Refinement Procedure*

The genetic algorithm is based on the observation that the line-segments - having the same direction and constituting the borders of blocks (or spots) - are ideally equidistant. Although this observation is true for an ideal microarray image, in reality due to rotations, misalignments and local deformations of the ideal rectangular grid, it may not be valid. As a result, the determined line-segments may slightly vary from the optimal ones.

In order to tackle this problem, each line-segment L_i belonging to the S_V or S_H sets is replaced by a new one, L_i', so long as the following are valid:

- the line-segment L_i' is located inside the region R_i of G ($L_i' \in R_i$)
- the probability $P(L_i')$, of the line-segment L_i', to be part of the grid, is higher than the equivalent probability $P(L_i)$ of L_i, by more than a threshold T_p ($P(L_i') - P(L_i) > T_p$).

An example of the refinement procedure is depicted in Fig. 7.

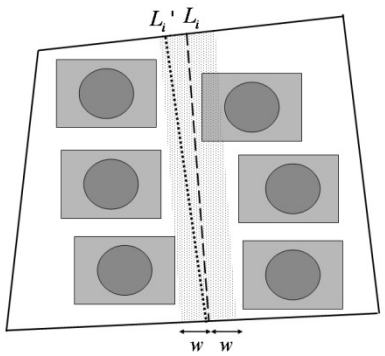


Fig. 7. Illustration of the refinement procedure: The line-segment L_i is replaced by the line-segment L_i'.

4. Spot-Segmenting Microarray Images

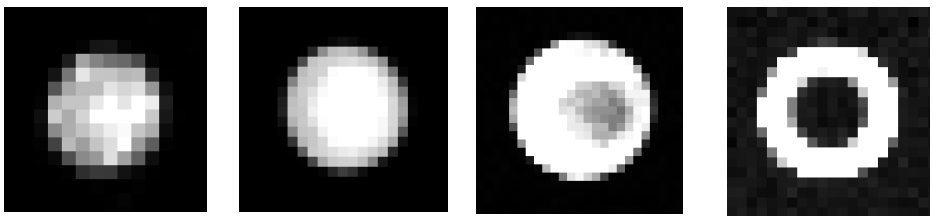
Based on empirical observations, microarray spots have some common characteristics. One of the most significant is that their shape is approximately elliptical and it can be discriminated into the following four categories[32]: peak, plateau, volcano and doughnut (Fig. 8).

Spot-segmentation is the second essential phase of microarray image analysis, which follows the gridding one. Due to the aforementioned common spots' characteristic, it may be more efficient to deal with the segmentation of a microarray spot thanks to its 3D modeling, using intensity as the third dimension. This is due to the fact that the contour of a microarray spot can be depicted in the image plane by drawing the contour of its spot-model. In the proposed method, the determination of the 3D spot-model of a specific real-spot is expressed as an optimization problem which is solved by using an evolutionary genetic algorithm.

4.1. *3D mathematical model for a microarray compartment*

4.1.1. Spot-model and its components

Let I_{REAL} be a compartment of a real microarray image containing one individual spot S_{REAL} and its background. The real-spot S_{REAL} can be represented by a spot-model S_{MODEL} which is expressed as a 3D-curve constituting of two components: The first component S_{MB} represents the 3D-curve of the main-body of the spot-model, while the second one S_{ID} represents the 3D-curve of the inner-dip of the spot-model.



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

Fig. 8. Examples of real microarray spots: (a) a peaked-shaped spot, (b) a plateau-shaped spot, (c) a volcano-shaped spot, and (d) a doughnut-shaped spot.

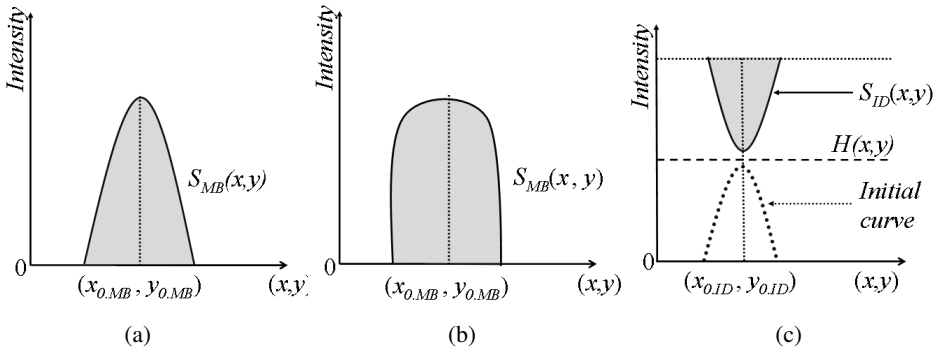


Fig. 9. Cross-sections of the 3D curves of the components of the spot-model: (a) a Gaussian $S_{MB}(x,y)$ 3D curve, (b) a plateau $S_{MB}(x,y)$ 3D curve and (c) a $S_{ID}(x,y)$ 3D curve.

The main-body $S_{MB}(x,y)$ of the spot-model can be expressed as the diffusion function $C(x,y)$ proposed by Bettens et al.[33] This 3D-curve resembles the 3D Gaussian or plateau curve (Fig. 9a,b). Likewise, the inner-dip of the spot-model $S_{ID}(x,y)$ can be defined as a symmetrical 3D-curve to an 'initial $C(x,y)$ 3D-curve', in respect to an horizontal surface $H(x,y)$ (Fig. 9c). The spot-model $S_{Model}(x,y)$ is constructed by combining the $S_{MB}(x,y)$ and $S_{ID}(x,y)$ 3D-curves as the following equation indicates:

$$S_{MODEL}(x, y) = Min\left[S_{MB}(x, y), S_{ID}(x, y) \right] \tag{8}$$

A graphical explanation of Eq. 8 is depicted in Fig. 10. The resulting total-models depend on the 3D curves of their corresponding S_{MB} and S_{ID} components and especially on the relative position of their centers and heights. More precisely, in the case of the distance between the S_{MB} and S_{ID} centers being large, the resulting total-model resembles a peak-shaped spot (Fig. 10a and Fig. 10b). In the case of the distance between the S_{MB} and S_{ID} centers being small, the resulting total-model resembles a volcano-shaped spot (Fig. 10c) or a doughnut-shaped spot (Fig. 10d), according to the height of the S_{ID} 3D curve.

4.1.2. Compartment-model and its components

Likewise to the spot-model S_{MODEL}, let I_{MODEL} be a compartment-model which represents, in a 3D space, the real one I_{REAL} by modeling the latter's intensities' values. The I_{MODEL} can be expressed as a 3D-curve constituting of two components: i) a surface representing the average background intensity B_{AV} of the compartment-model and, ii) a 3D-curve representing the spot-model S_{MODEL}.

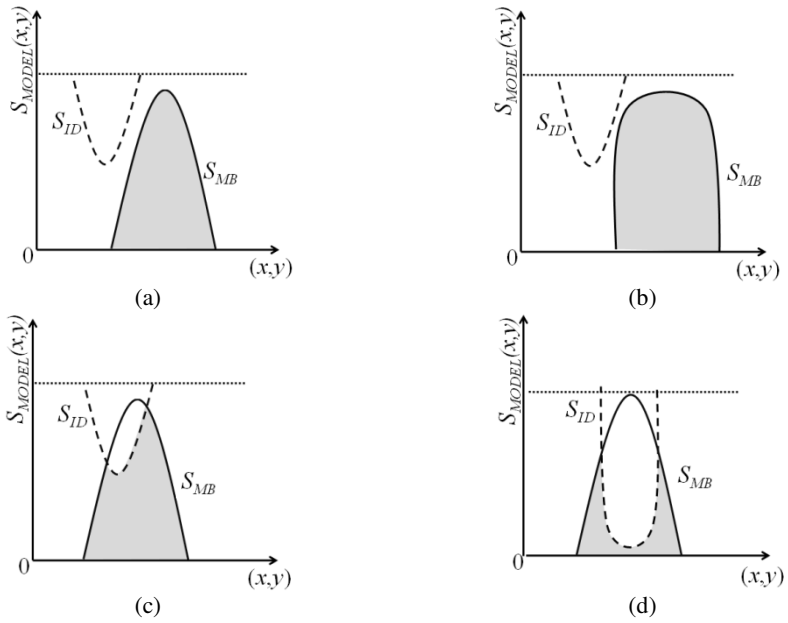


Fig. 10. 2D illustrations of four spot-models by portraying the cross-sections of their $S_{MB}(x,y)$ and $S_{ID}(x,y)$ 3D-curves. The resulting spot-models $S_{MODEL}(x,y)$ which are the surfaces coloured in grey are the following: (a) a peak-shaped spot, (b) a plateau-shaped spot, (c) a volcano-shaped spot, (d) a doughnut-shaped spot.

The compartment-model $I_{MODEL}(x,y)$ corresponds to a 3D-curve whose equation is defined as:

$$I_{MODEL}(x,y) = Max\left[B_{AV}, S_{MODEL}(x,y)\right], \tag{9}$$

where B_{AV} denotes the average background intensity of the compartment-model and it corresponds to a threshold of the lowest values of the $S_{MODEL}(x,y)$. Pixels whose values are lower than B_{AV} belong to the background and their values are set equal to B_{AV}. A graphical explanation of Eq. 9 is depicted in Fig. 11.

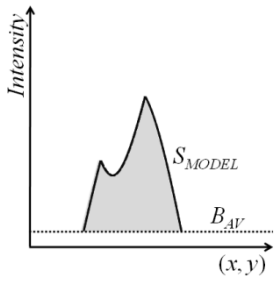


Fig. 11. Cross-section of the 3D curve of the compartment model.

4.2. *Evolutionary genetic algorithm for segmenting microarray images*

Given a specific compartment I_{REAL}, the genetic algorithm determines the compartment-model which represents optimally the real-one I_{REAL}. Thus, the contour of the real-spot is depicted by drawing the contour of the spot-model inside its compartment.

4.2.1. Chromosome Representation and Characterization

A chromosome *m* represents in a 3D space a specific compartment-model I^m_{MODEL}. It has been encoded as a numerical sequence consisting of three segments (Fig. 12). The first segment encodes the value of the average background intensity B^m_{AV} of the compartment-model. The second segment encodes the values of the variables of the main-body S^m_{MB}, while the third segment encodes the values of the variables of the inner-dip S^m_{ID} of the spot-model S^m_{MODEL}.

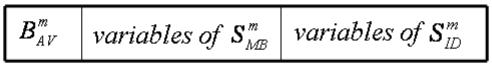


Fig. 12. Illustration of a chromosome *m*.

Each chromosome *m* is characterized as '*inefficient*' or '*efficient*'. The former denotes a chromosome which represents a compartment-model I^m_{MODEL} that hardly resembles the real-one I_{REAL}. In this case, the S^m_{MB} component of its spot-model S^m_{MODEL} approximates deficiently the real-spot S_{REAL}. The latter denotes a chromosome which represents a compartment-model I^m_{MODEL} that resembles – to a degree – the real-one I_{REAL}. In this case, the S^m_{MB} component of the spot-model S^m_{MODEL} approximates – to a degree – the real-spot S_{REAL}. Fig. 13 illustrates examples of '*inefficient*' and '*efficient*' chromosomes.

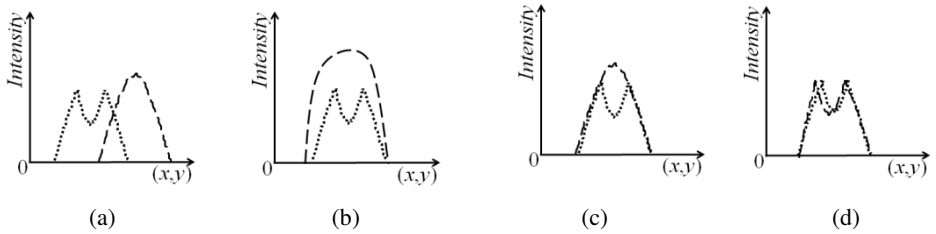


Fig. 13. 2D illustrations of four chromosomes: (a,b) two inefficient chromosomes, (c,d) two efficient chromosomes.

4.2.2. Chromosome Evaluation

The key concept of chromosome evaluation is that the higher the resemblance of a compartment-model I^m_{MODEL} to the real-one I_{REAL} is, the higher the value of the fitness function of the chromosome m becomes. Therefore, the relative intensity error $E(x,y)$ between the intensity of the model-compartment's pixel (x,y) and the corresponding one of the real-compartment's pixel is calculated as follows:

$$E(x,y) = \frac{|\, I^m_{MODEL}(x,y) - I_{REAL}(x,y)\,|}{I_{REAL}(x,y)} \tag{10}$$

The fitness function $F_S(m)$ of a chromosome m is defined by the following equation:

$$F_S(m) = - \iint\limits_{x,y \in Compartment} E(x,y)\ dxdy \tag{11}$$

Using the Fitness Function $F_S(m)$, the genetic algorithm can progressively assign – from left to right – a higher fitness value to the chromosomes representing the compartment-models in Fig. 13.

5. Results

Several experiments on various real and synthetic cDNA microarray images were conducted so as to evaluate the performance of our methods in gridding and spot-segmentation.

5.1. *Gridding Results*

The microarray images used for the evaluation were obtained from the Stanford Microarray Database (SMD),[31] which is publicly available and broadly used. These 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.[35] BCR-ABL is a fusion gene product resulting from translocation between the 9th and the 22nd chromosomes. In all the conducted experiments for gridding evaluation, the parameters have been experimentally adjusted once, and thus they remained constant during all experiments.[36] Thus, the whole experimental procedure on real cDNA microarray images took place without any human intervention.

The efficiency of the proposed method was evaluated by means of the statistical analysis described by Blekas et al.[36] More precisely, each microarray spot was classified in one of the following three categories: '*perfectly*',

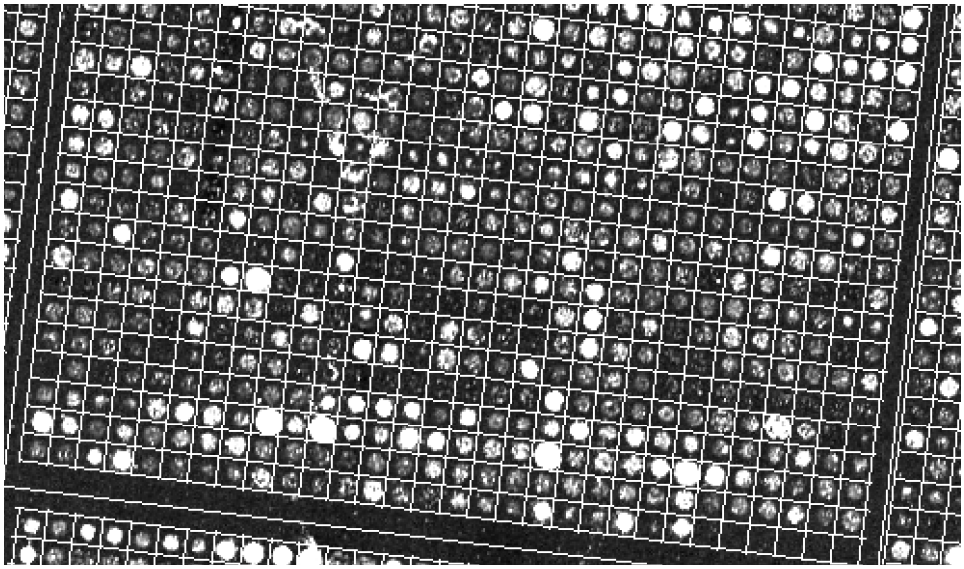


Fig. 14. Example of a gridding result.

'*marginally*' and '*incorrectly*' gridded. A spot was '*perfectly*' gridded if the entire spot area was contained inside the equivalent compartment of the grid. A spot was '*marginally*' or '*incorrectly*' gridded - respectively - if more or less than 80% of the entire spot area was contained inside the equivalent compartment of the grid. It should be noted that the ground truth is given in the SMD.

Using the gridding method, 95.1% of spots were perfectly placed inside the compartment, 4.3% were very nearly gridded, while only 0.6% were gridded incorrectly. It should be pointed out that the gridding method outperforms established techniques,[36] such as the one proposed by Blekas et al,[36] as well as popular software programs such as ScanAlyze and SpotFinder. Fig. 14 depicts the gridding result of a noisy and rotated microarray sub-image containing several spots of various intensities and sizes. This example indicates that the effectiveness of the proposed method is not influenced by spot intensities and sizes neither by rotations and misalignments of the ideal rectangular grid.

5.2. *Spot-Segmentation Results*

Since ground truth for the segmentation's results on real microarray images does not exist – even for the SMD database –, synthetic microarray images were used for the evaluation of the proposed segmentation approach. More precisely, the microarray images used for the evaluation were obtained on the Internet.[38] They

have been produced by the microarray simulator of Nykter which generates synthetic microarray images with realistic characteristics[39] and varied quality. Half of the images (good quality images) have low variability in spot sizes and shapes, and their noise level is reasonably low. The rest of them (low quality images) contain spots whose shape and size vary substantially. In addition, the noise level is sufficiently higher in the low quality images. Finally, each image is digitized at 330 x 750 pixels and contains 1000 spots.

Using the presented spot-segmentation approach, 91.5% of spots were very efficiently segmented, and no spurious spot has been detected. Fig. 15 illustrates the segmentation result of a microarray block taken from a good-quality synthetic image, while Fig. 16 illustrates the segmentation result of a microarray block taken from a low-quality synthetic image. On these segmentation results, one can observe that the proposed method has optimally segmented all the microarray spots of Fig. 15, and nearly all the microarray spots of Fig. 16.

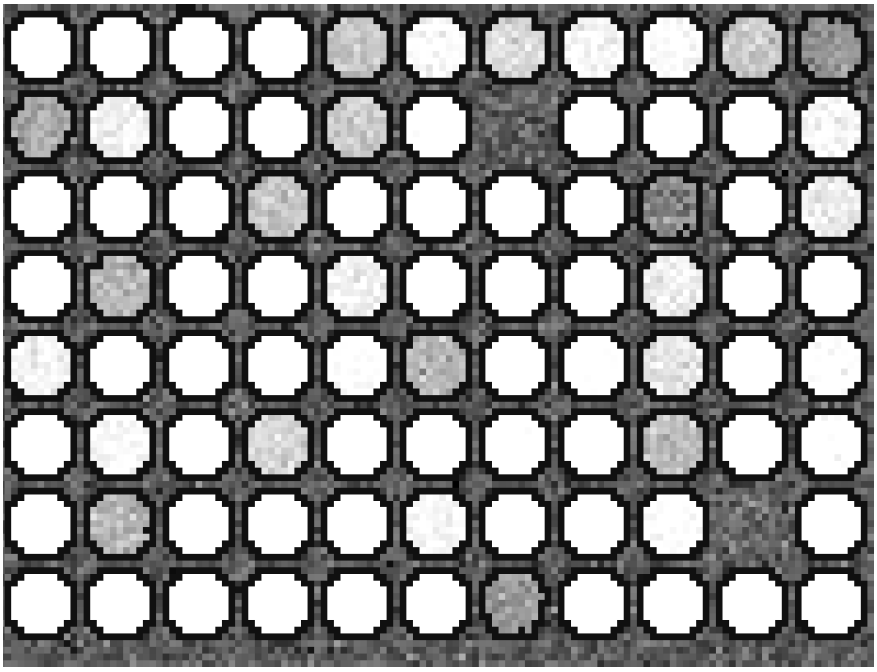


Fig. 15. Examples of spot-segmentation results on a good quality sub-image.

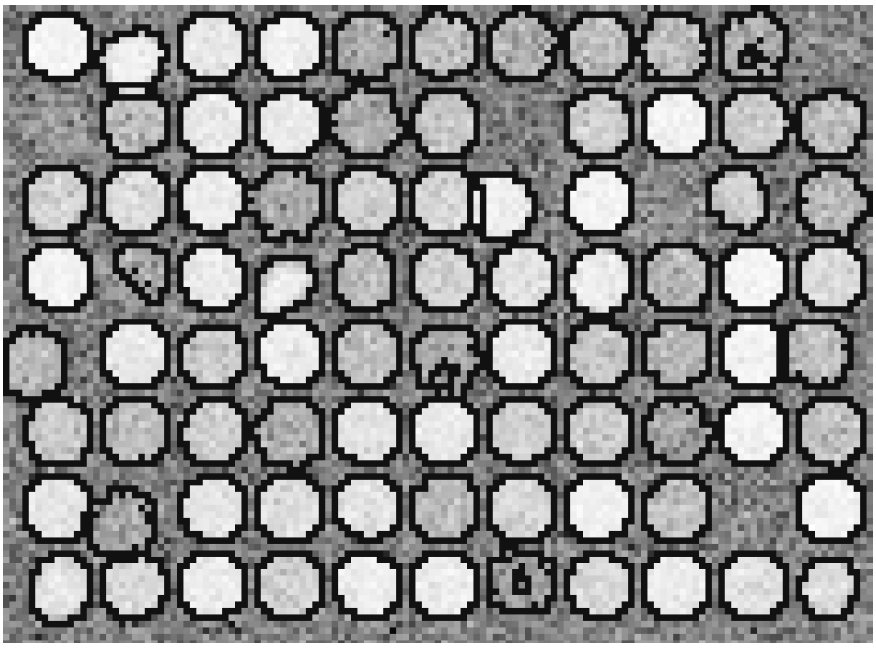


Fig. 16. Examples of spot-segmentation results on a low quality sub-image.

6. Conclusions

In the process of microarray image analysis, gridding and spot-segmenting remain undoubtedly the most challenging stages. Although much progress has been made, they still require human intervention which can affect significantly the biological conclusions reached during microarray experiments.

In this chapter a method for gridding and spot-segmenting microarray images is presented which is based on the concept of evolution. The proposed method is fully automatic since all needed parameters have been adjusted once and they have been kept fixed during all the experiments. Consequently, there is no requirement of any input parameter or human intervention in order to determine properly the gridding and the contours of microarray spots. The experimental results over synthetic and real images firmly confirm the validity of our method, as well as its robustness and effectiveness.

References

1. E. K. Lobenhofer, P. R. Bushel, C. A. Afshari and H. K. Hamadeh, "Progress in the Application of DNA Microarrays," *Enviromental Health Perspectives*, vol. 109, no. 9, pp. 881–891, Sept. 2001.
2. A. M. Campbell and L. J. Heyer, *Discovering Genomics, Proteomics & Bioinformatics*, 2nd ed., Pearson Benjamin Cummings, 2007, pp. 233-238.
3. Y. H. Yang, M. J. Buckley, S. Dudoit, and T. Speed, "Comparison of methods for image analysis on cDNA microarray data," *Journal of Computational & Graphical Statistics*, vol. 11, no. 1, pp. 108–136, Mar. 2002
4. X. H. Wang, and R. S. H. Istepanian, "Microarray image enhancement by denoising using stationary wavelet transform," *IEEE Trans. of Nanobioscience*, vol. 2, no. 4, pp 184-189, Dec. 2003.
5. W. B. Chen, C. Zhang, and W. L. Liu, "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.
6. Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," in *Proc. National Academy of sciences*, USA, 2002, pp.14031-14036.
7. M. B. Eisen. (1999). ScanAlyze. [Online]. Available: http://rana.lbl.gov/EisenSoftware.htm
8. J. Buhler, T. Ideker, and D. Haynor, "Dapple: improved techniques for finding spots on DNA microarrays," UW CSE Technical Report UWTR 2000-08-05, pp. 1-12, Aug. 2000.
9. Biodiscovery Inc. (2005). ImaGene. [Online]. Available: http://www.biodiscovery.com/imagene.as
10. Buckley MJ (2000) The Spot User's Guide, CSIRO mathematical and information sciences. http://www.cmis.csiro.au/IAP/Spot/ spotmanual.htm
11. N. Deng and H. Duan, "The Automatic Gridding Algorithm based on projection for Microarray Image", in *Proc. Int. Conf. Intelligent Mechatronics and Automation*, Chendu, 2004, pp. 254-257.
12. A. W. C. Liew, H. Yan, and M. Yang, "Robust adaptive spot segmentation of DNA microarray images," *Pattern Recognition*, vol. 36, no. 5, pp. 1251–1254, May 2003.
13. J. Angulo and J. Serra, "Automatic analysis of DNA microarray images using mathematical morphology," *Bioinformatics*, vol. 19, no. 5, pp. 553–562, 2003.
14. R.Hirata, J. Barrera, R. F. Hashimoto, and D. O. Dantas, "Microarray gridding by mathematical morphology," in *Proceedings of 14th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '01)*, pp. 112–119, Florianopolis, Brazil, October 2001.
15. H.-J. Jin, B.-K. Chun, and H.G. Cho, "Extended epsilon regular sequence for automated analysis of microarray images," in The IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05), the Workshop on Computer Vision Methods for Bioinformatics (CVMB), San Diego, Calif, USA, June 2005.
16. P. Bajcsy, "Gridline: automatic grid alignment in DNA microarray scans," IEEE Trans. Image Processing, vol. 13, no. 1, pp. 15–25, Jan. 2004.
17. M. Steinfath, W. Wruck, H. Seidel, H. Lehrach, U. Radelof, and J. O'Brien, "Automated image analysis for array hybridization experiments," *Bioinformatics*, vol. 17, no.7, pp. 634–641, Jul. 2001.

18. N. Brandle, H. Bischof, and H. Lapp, "Robust DNA Microarray image analysis," *Machine Vision and Applications*, vol. 15, no.1, pp. 11–28, Oct. 2003.
19. J. Ho, W. L. Hwang, H. H. S. Lu, D. T. Lee, "Gridding Spot Centers of smoothly distorted microarray images," *IEEE Trans. Image Processing*, vol. 15, no. 2, pp. 342-353, Feb. 2006.
20. G. Antoniol and M. Ceccarelli, "Microarray image gridding with stochastic search based approaches," *Image and Vison Computing*, vol. 25, no. 2, pp. 155-163, Feb.2007.
21. Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal Biomedical Optics*, vol. 2 no. 4, pp. 364-374, 1997.
22. D. Bozinov, and J. Rahnenfuhrer, "Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering," *Bioinformatics*, vol. 18, no. 5, pp. 747–756, 2002.
23. J. Rahnenfuhrer, and D. Bozinov, "Hybrid clustering for microarray image analysis combining intensity and shape features," *BMC Bioinformatics*, vol. 5, no. 5, p. 47-58, Apr. 2004.
24. E. Ergut, Y. Yardimci, E. Mumcuoglu, and O. Konu, "Analysis of microarray images using FCM and K-Means clustering algorithms," *in Proc. Int. Conf. Signal Processing*, 2003, pp. 116–121.
25. R. Nagarajan, "Intensity-based segmentation of microarray images," I*EEE Trans. Medical Imaging*, vol. 22, no. 7, pp. 882–889, Jul. 2003.
26. Q. Li, C. Fraley, R. E. Bumgarner, K. Y. Yeung, and A. E. Raftery, "Donuts, scratches and blanks: robust model-based segmentation of microarray images," *Bioinformatics*, vol. 21, no. 12, pp. 2875-2882, Apr. 2005.
27. M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, 1999.
28. D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Boston: Addison-Wesley, Reading, 1989, ch. 1.
29. B. L. Miller, and D. E. Goldberg, "Genetic Algorithms, Tournament selection, and the Effects of Noise," Complex Systems, vol. 9, no. 3, pp. 193-212, 1995.
30. F. Herrera, M. Lozano, and A. M. Sanchez, "Hybrid crossover operators for real-coded genetic algorithms: An experimental study," *Soft Computing*, vol. 9, no. 4, pp. 280-298, Apr. 2005.
31. S. H. Ling, and F. H. F. Leung, "An improved genetic algorithm with average-bound crossover and wavelet mutation operations," *Soft Computing*, vol. 11, no. 1, pp. 7-31, 2007.
32. H. Y. Kim *et al*., "Characterization and simulation of cDNA microarray spots using a novel mathematical model," *BMC Bioinformatics*, vol. 8, pp. 485-496, March 2007.
33. E. Bettens, P. Scheunders, D. van Dyck, L. Moens, and P. van Osta, "Computer analysis of two-dimensional electrophoresis gels: a new segmentation and modeling algorithm," *Electrophoresis*, vol. 18, pp. 792-798 1997.
34. Standford Microarray Database. [Online]. Available: http://genome-www5.stanford.edu/
35. D. Juric *et al.*, "Differential Gene Expression Patterns and Interaction Networks in BCR-ABL-Positive and -Negative Adult Acute Lymphoblastic Leukemias," *Journal of Clinical Oncology*, vol. 25, no. 11, pp. 1341-1349, April 2007.
36. E. Zacharia, D. Maroulis, "Microarray Image Analysis based on an Evolutionary Approach", *in Proc of the 19th International Conference on Pattern recognition* (*ICPR*), Tampa, Florida, USA, 2008.

37. K. Blekas, N. P. Galatsanos, A. Likas, and I. E. Lagaris, "Mixture Model Analysis of DNA Microarrray Images," *IEEE Trans. Medical Imaging*, vol. 24, no. 7, pp. 901-909, July 2005.
38. Microarray Database. [Online]. http://www.cs.tut.fi/sgn/csb/spotseg/
39. M. Nykter, *et al*., "Simulation of microarray data with realistic characteristics," *BMC Bioinformatics*, vol. 7, pp. 349-366, Jul. 2006.