Neural Comput & Applic (2019) 31:2311–2327
DOI 10.1007/s00521-017-3190-6

ORIGINAL ARTICLE

Neural networks ensemble for automatic DNA microarray spot classification

Juan Carlos Rojas-Thomas[1] · Marco Mora[2] · Matilde Santos[1]

Received: 1 December 2016 / Accepted: 14 August 2017 / Published online: 30 August 2017

Abstract In this work, a new step for the DNA microarray image analysis pipeline is proposed using neural computing techniques. We perform the classification of the spots into morphology-derived classes in order to assist the segmentation procedure that is traditionally performed after the gridding process. Our method consists of extracting multiple features from each individual spot area (or cell—derived from the gridding process) that are then reduced to a presumably optimal subset using a feature selection process, the sequential forward selection algorithm. Classification is then realized by means of a neural network ensemble with a tree-like structure, made up of seven multi-layer perceptron networks. The architecture of each neural network has been obtained through an exhaustive automatic searching process that optimizes the size of the network as a function of the classification error rate. The neural ensemble classifier is tested on two subgrids extracted from real microarray DNA images and is shown to achieve high accuracy rates over the seven different classes of spot. In addition, a dataset with more than 1000 samples of classes of spot has been generated and made freely available.

Keywords DNA microarray images · Spot classification · Neural networks ensemble · Optimization · Sequential forward selection · Image processing

1 Introduction

Microarray technology has become one of the indispensable tools that many biologists use to monitor genome wide expression levels of genes in a given organism. The gene expression level indicates the synthesis of different messenger ribonucleic acid (mRNA) molecule in a cell. Using this gene expression level, it is possible to diagnose diseases, identify tumours, select the best treatment to resist illness and detect mutations [1]. Thus, it is important to develop computational techniques that provide automatic classification of genes for the diagnosis of particular diseases.

In the literature of DNA microarray image processing, different classes of spot have been defined with various objectives, such as to assess the performance of the segmentation algorithms or the simulation of microarray images based on statistical models. Its classification is traditionally performed by an expert in a visual way. The automatic classification of real spot images as part of the microarray image processing pipeline is not well developed due to its complexity, the high level of degradation of these images and the high values of intensities they present.

In this work, a new step for the DNA microarray image analysis pipeline is proposed. Indeed, the main contribution of this work is to propose a new methodology for microarrays DNA images processing. The novelty consists of the fact of using the whole cell to classify the type of spot. As it has been proved, this approach helps the segmentation and subsequent identification of the spots. Even more, it can be used to develop an adaptive segmentation algorithm using the class of spot as input information. The segmentation is then more accurate and the quantization step could be also enhanced and made lighter.

✉ Matilde Santos
msantos@ucm.es

[1] Facultad de Informática, Universidad Complutense de Madrid, C/Profesor García Santesmases 9, 28040 Madrid, Spain

[2] Department of Computer Science, Universidad Católica del Maule, Avenida San Miguel 3605, Talca, Chile

Another advantage of the proposed method is that it works with many descriptors instead of a few of them. The more relevant are selected by the well-known sequential forward selection (SFS) algorithm [2] that reduces them to a supposedly optimal subset. As a result, the developed classifier is adapted to the specific characteristics of the problem.

Classification is then performed by a neural ensemble with a tree structure, made up of seven multi-layer perceptron (MLP) neural networks. The configuration of each neural network is estimated automatically by an exhaustive evolutive searching process that optimizes the size of the network as a function of the classification error rate. The neural classifier is tested on sub-grids extracted from real microarray DNA images and is shown to achieve high accuracy rates. Considering the complexity of the problem, these results confirm the efficiency of this approach.

Another contribution of this work is the generation of a database that is created from the experiments considering the six classes of spot defined in [3] and a seventh class called empty spot or absent spot [4]. The database contains 725 samples for training and 336 for testing. It has been made available for all researches with free access.

The paper is organized as follows. The rest of this section establishes the framework of the research and the background. Section 2 presents the definition of the spot classes and details the generation of a new database of microarray images that will be used as benchmark. Section 3 explains the selection of features process. Section 4 describes the general architecture of the ensemble of classifiers and the methodology used to configure and train each one of its neural networks. Section 5 shows and discusses the results obtained with real images. Finally, Sect. 6 summarizes the conclusions and future works.

1.1 Research framework and background

Learning the control of gene expression is critical for our understanding of the relationship between genotype and phenotype. The need for reliable assessment of transcript abundance in biological samples has driven scientists to develop technologies such as DNA microarray and more recently RNA-Seq to meet this demand [5].

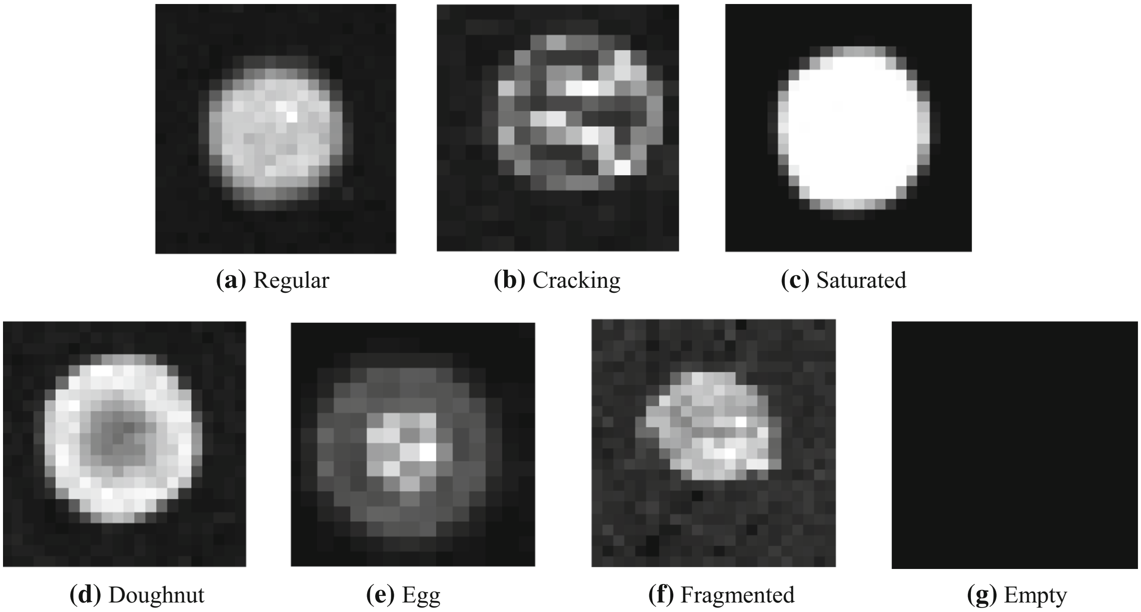
Microarray analysis has become a great source of information for biologists to understand the workings of DNA which is one of the most complex codes in nature. The DNA microarrays are a substrate, with a matrix shape, over which genetic material is deposited, generally following a regular pattern [6]. When the DNA of the samples interacts with the reference genes of the microarray, a hybridization process occurs. In the specialized literature, the specific region where the hybridization process of a particular gene occurs is called “spot”. After the hybridization process, two images of the whole microarray are generated. Then they are combined in a final image of RGB format. The final colour of a spot is a function of the ratio between the intensities of the two dyes (red and green) and, as a result, it indicates the relative abundance of the corresponding gene in the samples [1]. The digital processing of these images aims at obtaining measures of the quantity of the material hybridized of each sample.

Ideally all spots are round and have the same diameter, but in fact they vary in size and shape, and present artefacts which distort the image [7] and, even sometimes, the intensity of the spot is lower than the background [8]. Previous researches have tried to categorize this spot variability through the definition of classes of generic models, described by a set of parameters, with different objectives. In [9], the authors identify four classes of spots. In [10, 11], the problem of processing saturated spots is presented, which corresponds to spots that register a brightness higher than the detection capacity of the scanner.

An interesting work is presented in [3], where the spot is classified using the information of the whole cell. First, the image of the cell is transformed to polar coordinates, the radial/angular projections are obtained, the granulometric curves are calculated and, finally, statistics are extracted from those projections for categorizing the spots. The authors in [12] also propose the idea of clustering over a full image area in order to accomplish the segmentation of cDNA microarray images.

The task of spot segmentation falls within the category of classification, that is, assigning pixels into spot and non-spot classes. In the case of a segmentation based on classifiers, the class of spot predicts the morphology of it. Therefore, a pixel can have a higher or lower probability of belonging to a spot according to the correlation between the spatial position, its intensity level and the intensity of its neighbours. In [13], a classification-based segmentation approach for cDNA microarray images is proposed. Pixels are classified into spot, background and noise, a process that directly leads to the final segmentation. Other similar works are shown in [14–16]. The paper by Biju and Mythili [17] presents a fuzzy clustering algorithm for cDNA microarray image spots segmentation. In our case, we have used up to seven classes of spot.

Regarding the use of neural networks (NN), they are a well-established tool for classification problems. Some of the examples found in the literature where this computing technique has been applied to microarray classification are the following, among others. Wang et al. [8] propose a method of segmenting microarray images using a series of artificial neural networks, which are based on multi-layer perceptron (MLP) and Kohonen networks. In [18], authors apply extreme learning machine (ELM)-based microarray



(a) Regular **(b)** Cracking **(c)** Saturated

(d) Doughnut **(e)** Egg **(f)** Fragmented **(g)** Empty

Fig. 1 Spot classes

data classification. But the goal in this case is not just to predict the class labels but to make clear what lead to the results, i.e. the genes involving with a specific disease. Therefore, they are mainly focused on the sequence feature selection problem.

A paper by Nanni et al. [19] develops a spot quality control strategy using a random sub-space ensemble of neural networks and a feature selection algorithm. They combine the random sub-space ensemble of Levenberg–Marquardt neural net classifiers and the SVM trained using the features selected by the Pudil's method. They aim at microarray spot quality classification, and thus, they work with only two categories: good and bad. In [20], the authors introduce a new approach for classifying DNA microarray data based on artificial neural networks and a dimensional reduction technique, the artificial bee colony (ABC) algorithm. They use this evolutive algorithm as an optimization technique for selecting the set of genes, from a DNA microarray, that best described a particular disease. After that, this information is used to train three types of ANN (multi-layer perceptron (MLP), radial basis function (RBF) and support vector machine (SVM)) for classifying the DNA microarrays associated with this disease. This is quite different from our work where we first calculate as many features as possible to then apply the SFS algorithm to reduce the number of them.

To summarize, microarray image segmentation is an important and still challenging problem. Although many microarray image segmentation (clustering) methods have been proposed in the literature, there has been little progress on developing efficient algorithms to segment a microarray image and it is still an open problem [21].

2 Materials

This work uses the definition of classes presented in [3]. It represents the majority of the cases observed in the databases of microarray images. It also includes the "absent spot" class, which consists, in general terms, in the cells that do not contain any spot, and whose intensity corresponds to the microarray background [4]. Examples of these classes are shown in Fig. 1.

The formal definition of these classes begins with the following function [3]:

$$f : E \rightarrow T = \{t_{\min}, t_{\min} + 1, \ldots, t_{\max}\} \tag{1}$$

where f corresponds to the grey intensity function of the whole microarray image, E is a discrete space ($E \subset \mathrm{Z}^2$) and T is a sorted set of discrete grey values. For an image of 16 bits, $t_{\min} = 0$ and $t_{\max} = 2^{16}–1 = 65{,}535$. The function $f(x)$ is the value of intensity of the image at the point $x = (x, y)$.

$Z_i \subset E$ is the cell that contains the spot i, defined as the area whose pixels are closest to this spot centre than any other. Based on this, f_i is defined as:

$$f_i : Z_i \rightarrow T \tag{2}$$

which corresponds to the intensity function of the pixel x, where $f_i(x) = f(x)$. That is, f_i is a restrictive form of $f(x)$ for the region defined by Z_i.

The generic model of intensity distribution for any spot i is given by the equation:

$$f_i(x) = a_i s_i\left(x - x_i^c\right) + n_i(x) \tag{3}$$

where $s_i(y)$ corresponds to the morphological form of the distribution of the spot I considering a cylindrical model, in which a_i represents the height of the cylinder associated with the spot i, x_c^i corresponds to the coordinates of the spot centre and $n_i(x)$ is the function that describes the noise presented in the image. The function that represents the noise has two components that identify two different sources of noise:

$$n_i(x) = n^g(x) + n_i^l(x) \tag{4}$$

where $n^g(x)$ represents the background signal at x, described by a Gaussian function, and $n_i^l(x)$ represents the noise signal of the local background associated with local aspects such as an inhomogeneous lighting and the presence of artefacts.

The morphological function s_i is built as follows:

$$s_i(y) = r_i(\theta)t_i(y) \tag{5}$$

where $r_i(\theta)$ corresponds to a function in polar coordinates that represents the contour of the spot. This defines a closed border:

$$s_i(y) = \begin{cases} t_i(y) : \left\|x - x_i^c\right\| \leq r_i(\theta) \\ 0 : \left\|x - x_i^c\right\| > r_i(\theta) \end{cases} \tag{6}$$

in which $t_i(y)$ is a spatial function of the spot intensities (texture).

Finally, according to the particular distribution of the functions $r_i(\theta)$ and $t_i(y)$, seven main classes (topologies) of spots are defined (Fig. 1):

- *Regular spot* This type of spot has a circular shape and a homogeneous distribution of intensities. Both the function of the radius, $r_i(\theta)$, and the global variation of intensities, $a_i t_i(y)$, are modelled by normal distributions. For the whole microarray, it is considered that the average value of the radius varies uniformly within a small interval. The range of values of the coefficient a_i is also represented by a uniform distribution in the range $[t_{\min}, t_{\max}]$.
- *Cracking spot* These spots have a cracked appearance, presenting dark regions or lines on its surface. The function of the radius $r_i(\theta)$ has the same normal distribution as a regular spot. The distribution of intensities is expressed as $t_i(y) = \tilde{t}_i(y) - \chi_i(y)$, where $\tilde{t}_i(y)$ follows the same model of a regular spot and $\chi_i(y)$ corresponds to a cracked function whose value is greater than zero if y belongs to the cracked region. The distribution of $\chi_i(y)$, the morphology of the lines (number, length, etc.) and the spatial position are difficult to model, but typically the thickness of the lines is lower than the spot radius.
- *Saturated spot* These spots represent a uniform level of intensity equals to the maximum values allowed, being $a_i = t_{\max}$. The texture function does not present variations ($t_i(y) = 1$) and the contour function of the spot $r_i(\theta)$ presents the same normal distribution as a regular spot.
- *Doughnut spot* These spots have a circular hole in their centre. The distribution of the intensities is a combination of two normal functions: one for the central region, $t_i^{\text{low}}(y)$, with a mean value of 0, and another for the peripheral region, $t_i^{\text{high}}(y)$, with a mean value of 1. The contour is defined by two functions that have normal distribution similar to the regular spot: one for the contour of the central region, $r_i^{\text{in}}(\theta)$ and another for the peripheral region, $r_i^{\text{out}}(\theta)$.
- *Egg spot* These spots have the reverse situation than the doughnut spot. The function that represents the intensities of the central region, $t_i^{\text{high}}(y)$, has an average intensity higher than the function that represents the intensities of the peripheral region, $t_i^{\text{low}}(y)$.
- *Fragmented spot* These spots present degenerated or irregular borders, with a significant standard deviation δ_r in relation to the mean. This type of spot presents, in addition, a smaller area than a typical spot. The function of intensities $t_i(y)$ is modelled as a normal distribution.
- *Empty spot* The cell does not have any spot, so that the intensity function $f_i(x)$ corresponds to the function of intensities of the microarray background. Following the Angulo model, in this case $r_i(\theta)$ should be equal to 0 for all θ.

2.1 Creation of a database of microarray images

The size and structure of the database can be crucial in order to get good classification results [22]. In this work, the microarray images database consists in a set of images of cells in a greyscale extracted from the original microarray images. They are saved with 16-bit tiff format. Therefore, the image resolution is 2^{16} and the average size of cells is 21 × 21 pixels.

The sources of the DNA microarray images are two databases widely known and with free access: the Princeton University Microarray Database (PUMAdb)[1] and the Stanford Microarray Database.[2] The experiments from which the images were extracted are: *Mus musculus* (id experiment: 58012, 57133, 57129), *Acyrthosiphon pisum* (id experiment: 101767, 101769, 102673, 102675, 102380), *Mycobacterium tuberculosis H37rv* (id experiment: 83716), *Francisella tularensis* (id experiment: 59225), *Chlamydomonas reinhardtii* (id experiment: 45603) and *Arabidopsis thaliana* (id experiment: 16673).

[1] Available at https://puma.princeton.edu/.

[2] Available at http://smd.princeton.edu/.

The two databases of cell images generated have 725 microarray images for training and 336 images for testing. That is, a total number of 1061 images that cover the whole spectrum of spot classes are now available for the scientific community interested in DNA microarray images processing.[3]

The technique used for the gridding is based on the statistical analysis of the one-dimensional projection of the image. This type of algorithm obtains the sum of all intensities over a set of adjacent lines (rows or columns), each result called the projection vector. Then the local extremes (maximum intensities for the signals and minimum for the background) are detected inside the projection vector. These local extremes represent an approximation to the centre of the spots. From these estimations, horizontal and vertical lines are generated, whose intersections indicate the positions where the spots are located in the microarray. The specific implementation used in this paper is based on [23].

When the training database was created, one of our objectives was to balance the distribution of the classes. The ground truth for the database was created by an expert who classified the images comparing them with the classes defined by Angulo. The final percentage of images of each class in the database is the following: regular 24%, cracking 18%, saturated 2%, doughnut 16%, egg 17%, fragmented 15% and empty 8%. The low proportion of saturated spots is due to the relative scarcity of this type of spot in the microarray images. However, as this type of spot is the easiest one to be classified, this fact does not affect the performance of the classifier. The total number of cells images chosen for the training database was 725. This dataset has been proved to be sufficient for the study.

Another goal was the generation of a free access repository of images for the research community[3]. The creation of a database is a laborious and tedious task; therefore, the availability of this database will allow the researchers to save a lot of time for their research, as well as to boost this research line and to facilitate the uniformity of criteria.

3 Feature selection process

Feature selection is a major problem in microarray spot quality classification methods [19]. The process of feature selection involves extracting a set of descriptors from the cell images, based on their intensities, and then selects those which optimize the separability of the classes. In our case, this process is repeated for each class independently. However, because of the nature of the problem, a preprocessing of the images is usually required before the extraction of the descriptors [24]. In this work, the preprocessing has been carried out as follows.

3.1 Pre-processing of cell images

In our work, the pre-processing consists of scaling the relative intensity of certain classes of spots. Due to the wide range of the microarray images intensities, a great number of spots remain invisible when the images are visualized. In order to visualize all the spots, each cell must be transformed to a greyscale (0–256) at local level (it means a transformation from 2^{16} to 2^{8} bits). The algorithm is applied on each colour channel separately once they have been converted to greyscale with a bit depth of 16 bits.

Only then the spot morphology is revealed, which is a critical point in order to assign the corresponding class during the creation of the database. Therefore, to keep the consistency between the database generation and the automatic classification process, it was decided, for certain classes of spots, to carry out a transformation to a greyscale, previously to the process of feature extraction. This applies to regular, cracking, doughnut, egg and fragmented spot classes. The other classes of spot are better categorized in the original space of intensities, with values between 0 and 65,535, and therefore, no transformation is required to extract the features. This applies to saturated and empty spot classes.

3.2 Set of features

The problem of optimal feature selection is still open, each method making a specific approximation to solve it [25]. In our case, we have worked with a wide and general framework. A total number of 363 features of intensities were computed for each spot image of the database. These features are grouped into the following categories [26]:

- *Basic features* Simple intensity information related to the mean intensity in the region; standard deviation, kurtosis and skewness of the intensity in the region; in the image, mean first derivative in the boundary of the region (gradient) and second derivative (Laplacian) in the region. Additionally, five contrast measurements can be extracted in order to analyse the difference of intensity between object and background. There are 11 basic intensity features that have been taken into account.
- *Statistical textures* Texture information extracted from the distribution of the intensity values based on the Haralick approach [27]. They are computed using co-occurrence matrices that represent second-order texture information (the joint probability distribution of

[3] Available at http://new.litrp.cl/index.php/data-repository.

intensity pairs of neighbouring pixels in the image), where mean and range—for five different pixel distances in eight directions—of the following variables were measured: (1) angular second moment, (2) contrast, (3) correlation, (4) sum of squares, (5) inverse difference moment, (6) sum average, (7) sum entropy, (8) sum variance, (9) entropy, (10) difference variance, (11) difference entropy, (12, 13) information measures of correlation and (14) maximal correlation coefficient. A total of 2 × 14 × 5 = 140 statistical features have been considered.

- *Local binary patterns* Texture information extracted from occurrence histogram of local binary patterns (LBP) computed from the relationship between each pixel intensity value with its eight neighbours. The features are the frequencies of each one of the histogram bins. LBP are very robust in terms of greyscale and rotation variations [28]. Other LBP features such as semantic LBP can be used in order to bring together similar bins. We use 59 uniform LBP features and 31 semantic LBP features, giving a total number of 90 features for this category.
- *Filter banks* Texture information extracted from image transformations such as discrete Fourier transform—magnitude and phase—discrete cosine transform (DCT) [29] and Gabor features based on 2D Gabor functions, i.e. Gaussian-shaped bandpass filters, with dyadic treatment of the radial spatial frequency range and multiple orientations. They represent an appropriate choice for tasks requiring simultaneous measurement in both space and frequency domains (usually eight scales and eight orientations). Additionally, the maximum, the minimum and the difference between both are computed. We use 16 DCT features, 16 Fourier features and 8 × 8 + 3 Gabor features, i.e. 16 + 2 × 16 + 67 = 115 features were extracted using filter banks.
- *Invariant moments* Information of shape and intensities based on the Hu moments [30], with a total of 7 features for this category.

3.3 Features selection

Data pre-processing and feature selection enhance the performance of the classifiers [18]. That is why, after computing the features previously described over each of the images of the database, on the whole cell, the more relevant subsets of features were selected for each class of spot using the well-known sequential forward selection (SFS) algorithm. This technique carries out a "bottom-up" search strategy that, starting from an empty feature subset and adding one feature at a time, achieves the best feature subset that can be obtained with the desired cardinality. It should be noted that due to the large number of initial characteristics (363), the application of an exhaustive search to find the set of optimal characteristics is not possible as it would lead to analyse 2^363 possible combinations.

In particular, the SFS adds to, or removes from, one feature at a time that most/less contributes to the correct classification. It is based on an error function that minimizes the amount of attributes while optimizing the classification. Finally the smallest possible set of features that optimizes the classification process is obtained [2].

Specifically, the classification criteria used in the SFS toolbox we have used are called SP100. With this method, the decision line is set so that the sensitivity is 100%, that is, you favour the class that interests you most, instead of placing the line decision in the middle of the overlapping region between classes as it is traditionally done.

The SFS is based on an error function that minimizes the number of attributes while optimizing the classification process. This error function maximizes the ratio: Sp = TN/(FP + TN) where TN = true negative and FP = false positive, that is, minimizes the number of false positives.

The feature selection process was applied to the training database (725 spot images). It is worth remarking that the testing was performed on a different dataset (336 images). Indeed, they are two different databases with different origin. The repository of 725 spots was generated from different microarray images in order to have enough samples of all the classes for the training of the networks. The repository of the test dataset (336 spots) is generated from the two real images shown in Figs. 6 and 8. Each real image has 168 spots.

4 Artificial neural networks ensemble

One of the goals of microarray data analysis is to cluster genes or samples with similar expression profiles together, to make meaningful biological inference about the set of genes or samples. Since traditional classifiers have not reached sufficient sensitivity and specificity, another possible way is combining the classifiers in ensembles. In this paper, we take advantage of neural networks, which have been proved efficient for microarray image processing [20], combining multi-layer perceptron (MLP) as a multi-class classifier.

The MLP has been selected for several reasons. The main reason is that a neural network is equivalent to a universal function approximator [31], with the property of being able to separate initially non-linearly separable data. In particular, a MLP with a single hidden layer allows to reduce the training error as much as desired by increasing

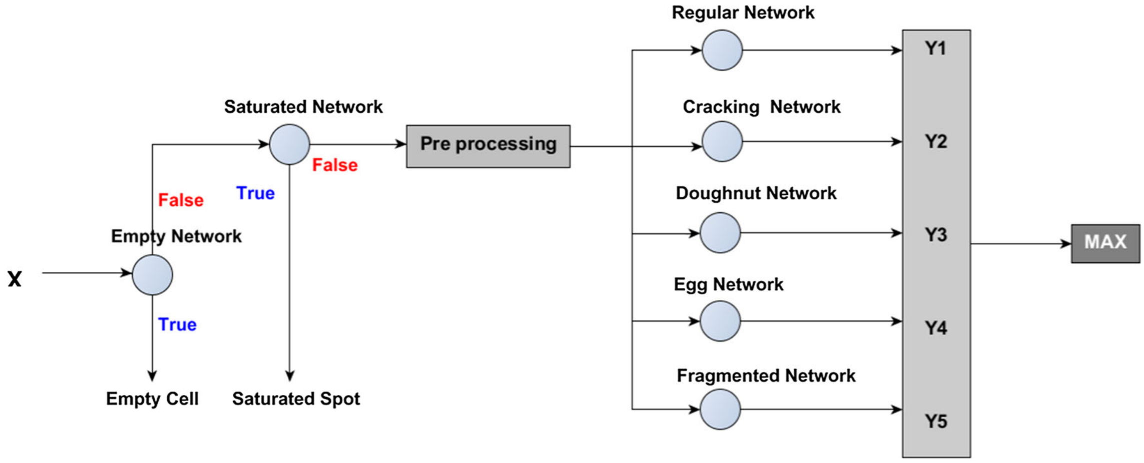


Fig. 2 Architecture of the neural classifier

the number of neurons in the hidden layer. In addition, the MLP training algorithm is well defined and is equivalent to a nonlinear optimization problem without constraints. Therefore, considering the neural network as an optimization problem, we can define cost functions that allow the automatic estimation of network parameters [32], with fast convergence searching algorithms [33]. Therefore, both the estimation of the configuration parameters and the training of the network can be carried out automatically.

4.1 Structure of the classifier

To address the problem of DNA microarray image classification, a hierarchical classifier is proposed. It is made up of 7 sub-classifiers, each one being specialized on detecting a specific class of spot. The tree-like structure of the classifier is justified by the fact that it facilitates the work of the sub-classifiers, having now to discriminate a more specific set of spot classes as it goes deep through the different levels of the classifier, increasing the percentage of hits. This structure is reinforced by the pre-processing applied to the images of some spot classes, as explained in the previous section.

The sequence of sub-classifiers was selected based on the knowledge obtained from experts and from some experiments. During some classification experiments, it was observed that certain kinds of spot are better discriminated on the 16 bits original image resolution (saturated and empty spot). This is because what most characterizes these two classes from the rest is its intensity level in the original scale of 2^{16} bits, which, besides being the classes with the highest rate of success, suggests classification these two classes first. For other classes, the best results are obtained using the greyscale of 8 bits. This way the consistency with the visual classification performed by the human expert on the training set is maintained. In fact, the expert performs the same processing before the classification to visualize the morphology of the spots that would, otherwise, remain largely invisible.

Therefore, based on the analysis of the problem and the obtained results, the classifier has been structured in three levels. In the first level, a sub-classifier determines whether the spot belongs to the empty class. If this is true, the classification process ends; otherwise, the image is passed to the second level. At this level, the image is processed by a sub-classifier that determines whether the spot belongs to the saturated class. If this condition is met, the classification process ends; otherwise, the image goes to the third level. At this last level, the image is processed by five sub-classifiers in a parallel way. Each one of them determines the level of membership of the spot to a specific class. In a competitive decision framework, the sub-classifier which brings the highest score assigns the class to the spot. The five sub-classifiers of this level correspond to regular, cracking, doughnut, egg and fragmented classes. Figure 2 illustrates the architecture of the classifier.

Each sub-classifier is implemented by a multi-layer perceptron (MLP) neural network, with linear activation function in the output layer and sigmoid function in the hidden layer, as this architecture corresponds to a universal function approximator [31]. This structure has been widely applied to classification in different fields [32, 35]. Theoretically, it is possible to reduce the classification error as much as you want by increasing the number of neurons in the hidden layer.

Every neural network of the classifier has as many inputs as features have been selected by the SFS algorithm. In addition, the networks have only a single output that will be close to 1 if the input belongs to the corresponding class or close to 0 otherwise. If the level of membership for each class is very close to each other, the algorithm still selects the highest. The fact that a spot had similar membership value to different classes would mean that it presents mixed characteristics, and that could lead to a definition of new classes of spots.

4.2 Neural networks optimization algorithm

With the purpose of obtaining neural networks with good generalization ability, the training process is performed controlling the over fitting, using the smallest possible number of neurons in the hidden layer. In order to determine this number of neurons of each neural network that gives an accurate classification, and at the same time keeping the network as simple as possible, an iterative searching procedure is adopted. The details of this procedure are the following:

- It works with a set of N_i spot samples by class, being i an integer number between 1 and 7 that represents the class.
- The iterative process starts with an initial configuration of one neuron in the hidden layer and gradually increases the size of this layer by adding one neuron each iteration. A maximum number of neurons, M, is set as the limit for this process. In this paper, M was set to 25.
- Each iteration generates a predefined number of networks, K, with the same number of neurons in the hidden layer, but with different weights values assigned randomly to each of them. In this paper, K was set to 1000.
- Each one of these networks is trained independently. To train and test the network performance, the samples are randomly chosen. For the training, validation and testing sets, the 70, 15 and 15% of the total samples were selected, respectively.
- The "repeated random sub-sampling validation" or random cross-validation strategy is used to end the training. Over the epochs, the classification error of the training set decreases gradually. The training stops when the classification error of the validation set starts to increase. This strategy also contributes to avoid the overfitting of the network.
- In case the training does not stop by the previous criterion, a maximum of training epochs, E, is used as a limit. For this paper E was set to 1000.
- After an iteration ends, from all of the K networks generated, the one with the lowest error rate is selected. This error rate is defined as the average of the quadratics errors of the network in the three sets (training, validation and testing), calculated after finishing its training process.
- After the M iterations, there are M selected networks that represent the best one of each iteration. From all of these, the one with the lowest error rate is chosen as the final classifier.

4.3 Training

The maximum values of the parameters used for training the neural network have been selected empirically, by trial and error, using the previous experience of the authors and information found in the literature. They were deliberately enlarged to cover the largest possible number of cases.

Each net is independently trained as a binary classifier with the Bayesian regularization backpropagation algorithm [33]. If the training procedure is understood as an optimization process with nonlinear restrictions, then this algorithm has the following characteristics:

- Cost function $F = \beta E_D + \alpha E_W$, where E_D is the sum of the squared errors, E_W is the sum of the squares of the network weights, and α and β are the parameters of the objective function. The parameters α and β are computed automatically according to a procedure described in [33].
- The searching method corresponds to the Levenberg–Marquardt algorithm [34], which allows a fast classification error convergence.

5 Results and discussion

First of all, regarding the feature selection process, a total number of 363 intensity features in the greyscale were computed for each cell image of the training database (725). In particular, the Balu toolbox was used to compute the descriptors.[4] For the regular class, 57 features were selected; for the cracking class, 30 features; for the saturated class, one feature; for the doughnut class, 30 features; for the egg class, 16 features; for the fragmented class, 31 features and, finally, for the empty class, one feature. After applying the SFS algorithm, the description of the features selected for each class and the corresponding success rate (Sp value, last row) are presented in Table 1. The classification accuracy has been obtained at the end of the selection process for the best set of features selected by SFS algorithm. It is given by the value of the Sp function (between 0 and 1), as defined in Sect. 3.3. The best value corresponds to Sp = 1 (FP = 0) and the worst one occurs when there are many false positives.

From these results (Table 1), it can be deduced that the Hu and Haralick features are not selected for any class, and the features that have been selected to classify the regular,

[4] Available at http://dmery.ing.puc.cl/index.php/balu.

Table 1 Description of the types of features selected for each class and the success rate given by the SFS algorithm

Descriptor	Class						
	Regular	Cracking	Saturated	Doughnut	Egg	Fragmented	Empty
Basic	4	1	1	2	1	4	1
Haralick	0	0	0	0	0	0	0
LBP	29	13	0	14	7	10	0
DCT	8	3	0	3	1	5	0
Fourier	7	5	0	4	6	6	0
Gabor	9	8	0	7	4	6	0
Hu	0	0	0	0	0	0	0
Total	57	30	1	30	16	31	1
Sp value	0.876	0.973	1	0.980	0.997	0.950	1

cracking, doughnut, egg and fragmented classes are mainly the local binary patterns. In detail, the percentage of LBP-based characteristics selected for each of the classes is as follows: regular, 51%; cracking, 43%; doughnut, 47%; egg, 44% and fragmented, 32%.

One of the peculiarities of this type of descriptors that makes them especially robust is that they compare relative intensities between pixels, giving consistent results under different grey levels. This is especially useful for the above-mentioned spots, which are principally defined by their morphology, but at the same time they present a great variety of intensity level for the same class.

In the cases of the saturated and empty spots, they are at the opposite ends of the range of intensities, which explains why the use of a single feature is enough to differentiate them.

In particular, for the empty spot the selected feature is the standard deviation of the image intensities. For the saturated spot, the selected feature is the standard deviation of the x and y axis profiles regarding the centre of gravity of the image.

These results may mean that the discarded features do not provide significant information for the classification of the spots. Perhaps they are redundant, and on the contrary, the selected features captured the nature of the problem in a more effective way.

Once the relevant features have been considered, the iterative algorithm that finds the best network (minimizing the number of neurons in the hidden layer) for each spot class is applied. Figure 3 shows the evolution of the number of neurons for each neural classifier during this optimization process. Tests were performed covering a range between 1 and 25 neurons for the hidden layer, which corresponds to the number of bars in the figures. Each bar represents the error rate given by the best network selected for each configuration. Then, from these 25 networks selected, the one with the lowest error rate was chosen as the classifier for each spot class.

Table 2 details the best number of neurons of the hidden layer of these classifiers. Figure 4 shows their configuration, with the number of neurons of the input, hidden and output layers. As it was expected, the results show a direct relation between the complexity of the class and the number of neurons in the hidden layer.

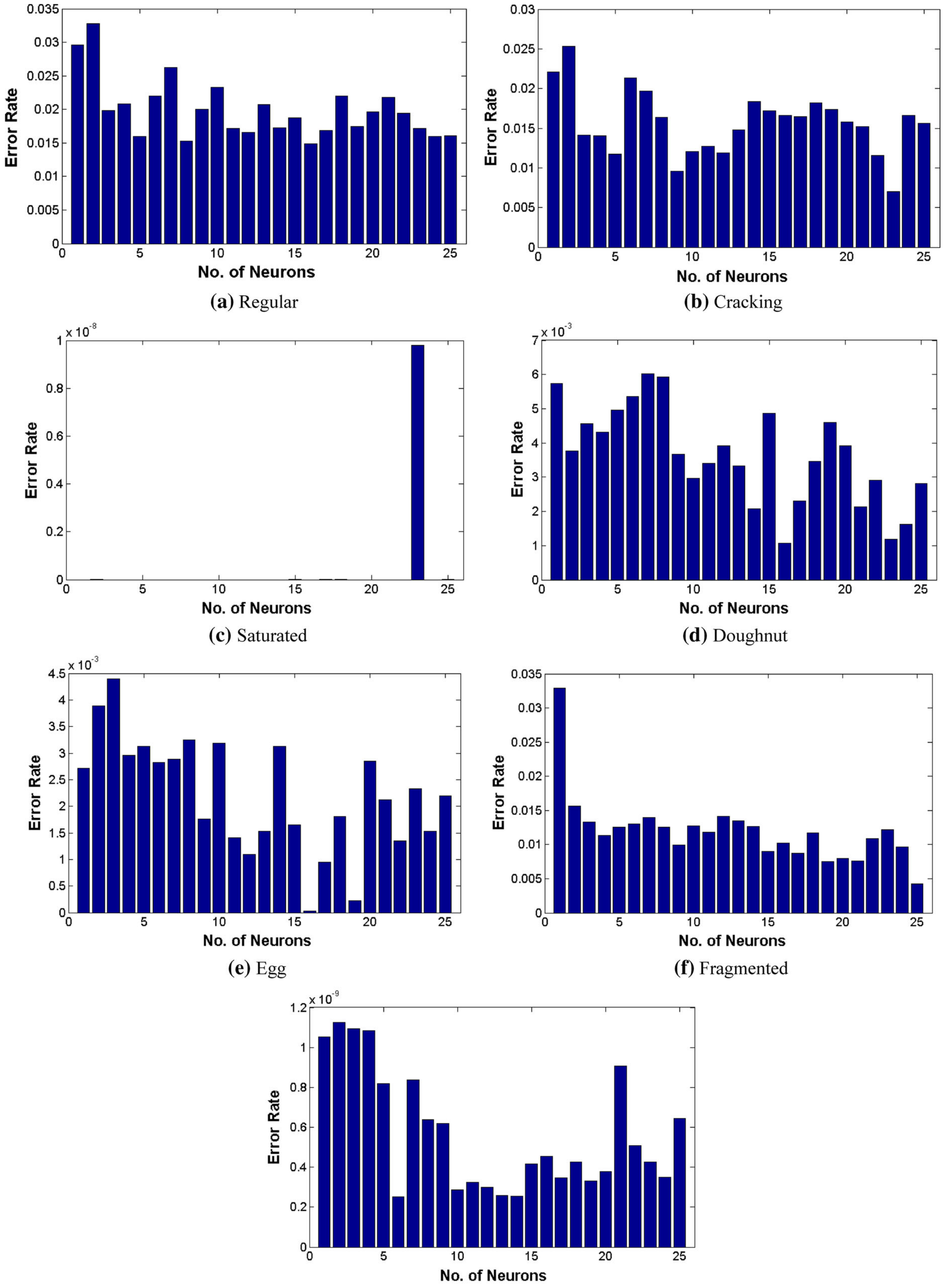
5.1 Results of the classification process

Table 3 shows the performance of the classifier selected for each spot class in terms of percentage of hits and misses in the classification during the training. The hit rate corresponds to the addition of the true positives (TP) and the true negatives (TN) given by each network; the error rate corresponds to the addition of the false positives (FP) and the false negatives (FN). For the outputs, a threshold value of 0.5 was used. If the output of the network is greater or equal than the threshold, then the spot is assigned to that class; otherwise, if the output is smaller than the threshold, the spot belongs to the other class. These rates are obtained for the three datasets used (training, validation and testing) and correspond to the best selected networks.

Precision and recall values were calculated for the test set (Table 3). Results prove that for new data (that have not been used for training), the precision of the classifier is quite high, being 1 for some of the classes and greater than 0.9 in all the cases. The same good results are obtained for the recall indicator.

The whole classifier (the ensemble of the neural networks) has been applied to 725 images of the training set. The output was compared to the ground truth, and statistics of hits and errors were calculated. Table 4 summaries these results. Out of these 725 images of the training dataset, only three of them were erroneously classified, giving a global hit percentage of 99.59%.

Even more, for testing the effectiveness of the classifier, two sub-grids that had not been used for the training were



(a) Regular

(b) Cracking

(c) Saturated

(d) Doughnut

(e) Egg

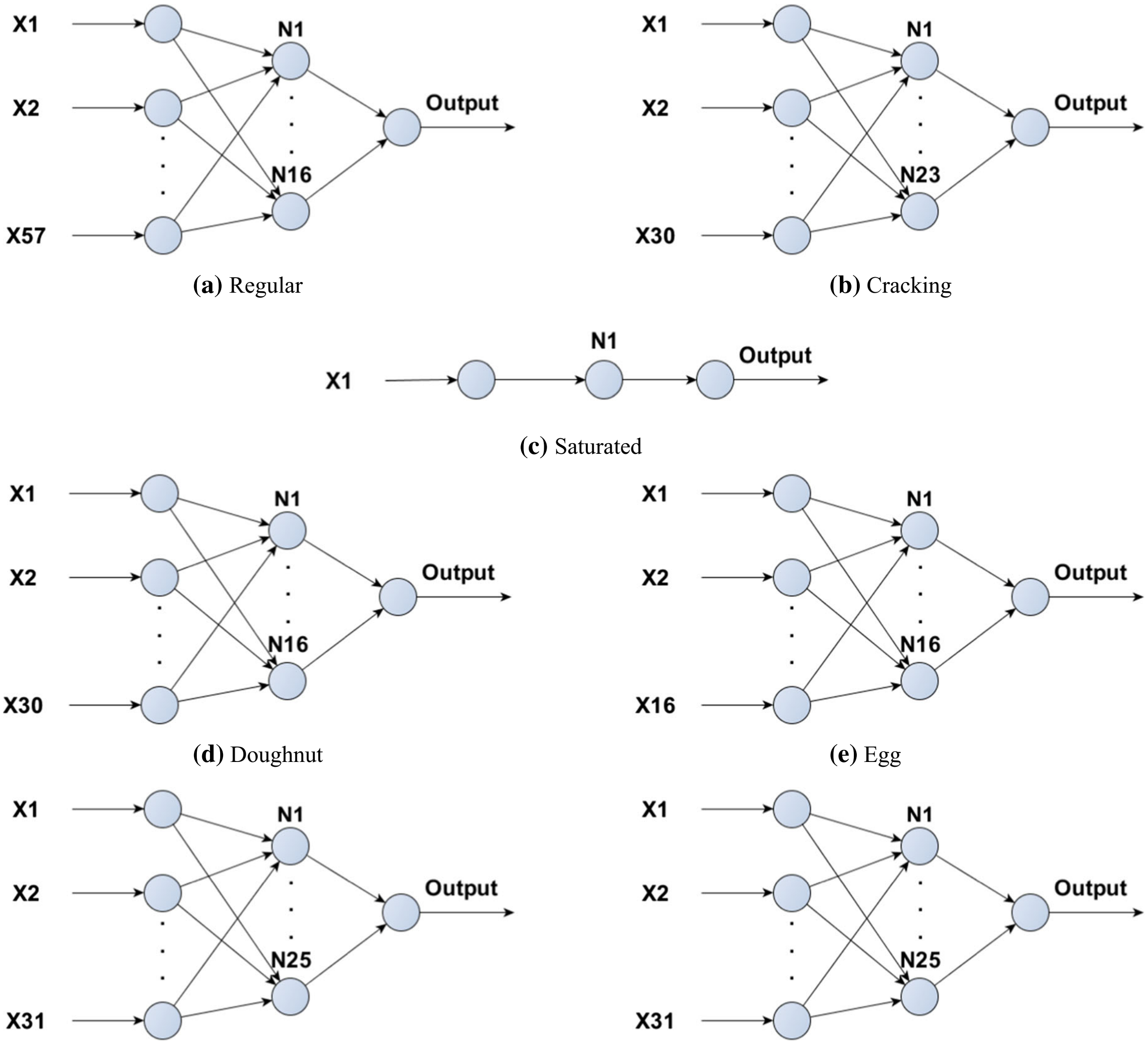
(f) Fragmented

(g) Empty

Fig. 3 Evolution of the error rate of the best networks and the number of neurons of the hidden layer

Table 2 Number of neurons of the hidden layer for each class

Spot class	Regular	Cracking	Saturated	Doughnut	Egg	Fragmented	Empty
# Neurons	16	23	1	16	16	25	6



(a) Regular

(b) Cracking

(c) Saturated

(d) Doughnut

(e) Egg

(f) Fragmented

(g) Empty

Fig. 4 Representation of the neural classifier of each class of spot

selected. After the gridding process, individual cells were extracted and two different testing databases were generated. Each image of these databases was assigned to its corresponding class by a human expert. Each database contains 168 cell images, giving a total number of 336 images for this testing dataset.

The classifier was tested with each testing database independently, and its hits and errors were registered. Figures 5 and 6 illustrate the images associated with the first series of testing, while Figs. 7 and 8 show the images associated with the second testing database.

In both testing databases, the classifier obtained a high hit rate. The results were 95.8 and 91.1% success for test sets 1 and 2, respectively.

In this first series of experiments, Fig. 5a shows the selected sub-grid. It is noteworthy that, to make it clearer, this image has been scaled from the original 16 bits to 8 bits (256 grey levels) at local level (sub-grid). This way the spots that otherwise will be invisible are now shown. Indeed, in Fig. 5c it is possible to see how in the original image the intensity of the pixels has been scaled to 8 bits at individual cell level, and therefore, all the spots are now visible.

Table 3 Results of each classifier, true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN), for the training, validation and testing sets

Class	Training				Validation				Test					
	Hits %		Error %		Hits %		Error %		Hits %		Error %		Precision	Recall
	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN		
Regular	28.8	71.2	0.0	0.0	22.2	74.7	2.0	1.0	23.2	73.7	1.0	2.0	0.96	0.92
Cracking	22.7	77.3	0.0	0.0	14.1	85.9	0.0	0.0	14.1	83.8	1.0	1.0	0.93	0.93
Saturated	1.5	98.5	0.0	0.0	5.0	95.0	0.0	0.0	1.0	99.0	0.0	0.0	1	1
Doughnut	16.5	83.5	0.0	0.0	15.9	84.1	0.0	0.0	14.0	86.0	0.0	0.0	1	1
Egg	18.8	81.3	0.0	0.0	15.0	85.0	0.0	0.0	15.0	85.0	0.0	0.0	1	1
Fragmented	15.9	84.1	0.0	0.0	16.8	83.2	0.0	0.0	11.2	87.9	0.0	0.9	1	0.93
Empty	7.7	92.3	0.0	0.0	8.3	91.7	0.0	0.0	9.2	90.8	0.0	0.0	1	1

Precision and recall are also listed

Table 4 Summary of the performance of the classifier

	Regular	Cracking	Saturated	Doughnut	Egg	Fragmented	Empty
% Success	99.59	99.86	100	100	100	99.72	100
% Error	0.41	0.14	0	0	0	0.28	0

Figure 5b shows how the gridding algorithm successfully generates the array containing the individual cells where the spots are confined. The gridding appears on the image of the sub-grid.

The class to which the spot belongs is represented by the colour of the cell border (Fig. 5c). The colour code is as follows: red for "regular" spot class, yellow for "doughnut", green for "cracking", light blue for "egg", blue for "saturated", fuchsia for "fragmented" and, finally, white border for "empty" spot class. These are the target classes to be identified by the classifier. It can be observed the prevalence of the "doughnut" spot class in this first series of experiments, followed by the "regular" and "empty" classes. Indeed, the distribution of these target classes in this first grid, in order of importance, is the following: doughnut 81.6%, regular 11.3% and empty 7.1%.

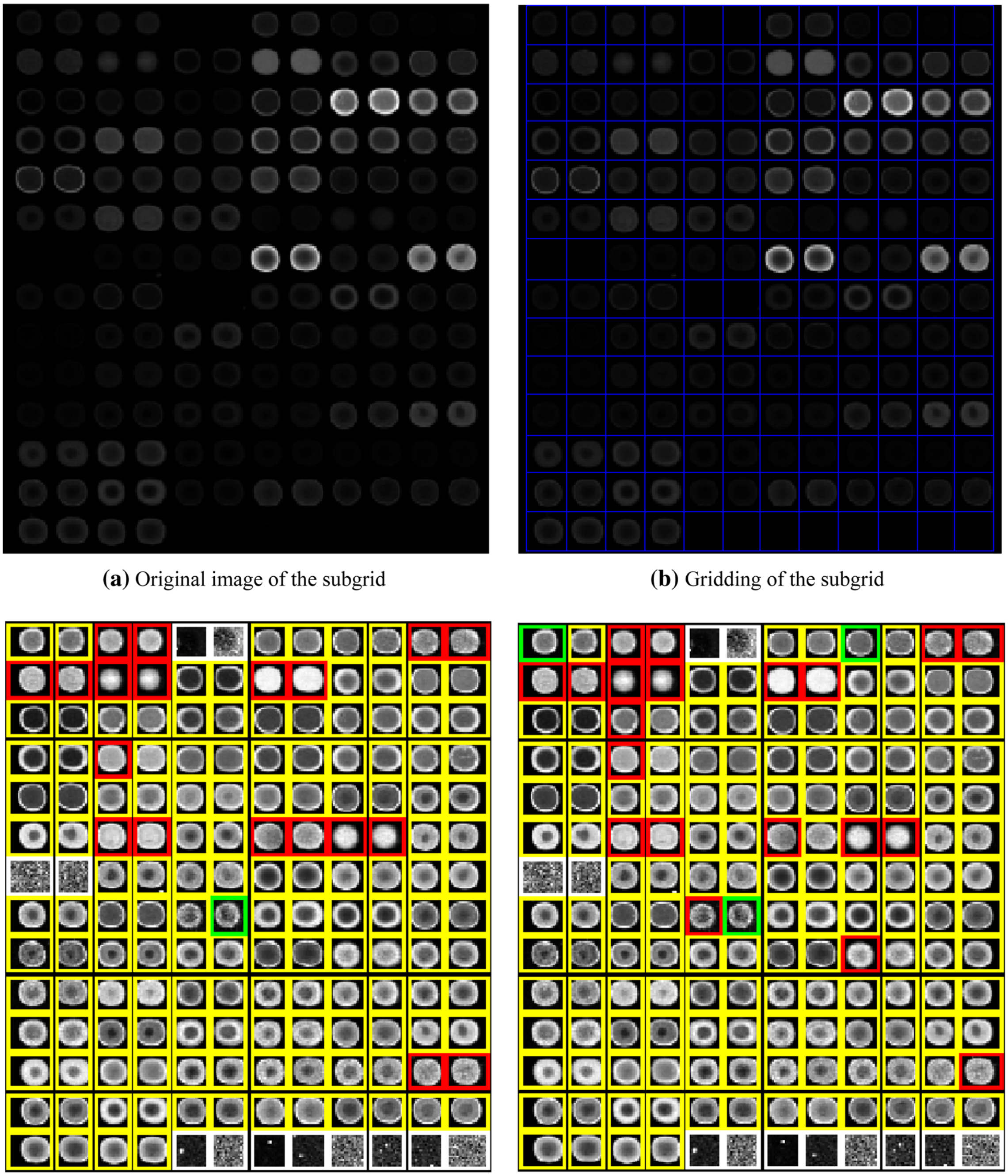
It can be also pointed out that the distribution of the minority classes is not random, but it tends to form clusters within the image. Specifically, it can be seen that the cells of the "empty" class are mostly concentrated in the last row of the grid, suggesting that these cells have been left so intentionally as part of the experiments. However, some of them are also presented in other rows, where other spots would be expected.

In the image, the empty cells show two different textures. Empty cells with granular texture only contain microarray background signal. The other type of empty cell looks mostly black due to the presence of noise. That is, noise intensity is greater than the background signal. Therefore, when applying the change of scale, the background is displayed with a uniform intensity, and only the noise signal is highlighted. Figure 5d shows how the classifier has successfully detected both types of empty cells. Most errors found correspond to "doughnut" spots wrongly classified as "regular" or "cracking".

Results of the classification are shown in Fig. 6, where 95.2% of the spots were rightly classified for this test set (blue squares).

There are two doughnut spots misclassified (red squares) in the central part of the image. It could be due to the irregular and granular appearance, associated with spots of very low intensities. A third doughnut spot wrongly classified as regular in the row below it is better defined, but the contrast between its inner and external regions is weak. Three more doughnut spots, located at the top of the image, have in common the characteristic of having very thin borders with irregular intensity level. Besides, their central areas do not present a sharp contrast regarding the outer rings. Finally, two "regular" spots, one at the bottom of the image and the other near the centre, were misclassified as "doughnut". In these cases, the spots have a granular centre (low intensity) and some higher intensity pixels at the border but without defining a crisp ring.

The same analysis has been done to the results of the classifier on test set 2. Again we have obtained the gridding image from the original one. Figure 7a, b shows the classes of spot of each cell and the output of the classifier. The distribution of the target classes in this grid, in order of importance, is the following: regular 90.4%, empty 4.8%, doughnut 3%, cracking 1.2% and fragmented 0.6%.



(a) Original image of the subgrid

(b) Gridding of the subgrid

(c) Target classes

(d) Output of the classifier

Fig. 5 Sub-grid of the first series of testing (**a**, **b**), with the target classes (**c**) and the output of the neural classifier (**d**). Spot colour code: regular *red square*, doughnut *yellow square*, cracking *green square*, egg *light blue square*, saturated *blue square*, fragmented *fuchsia square* and empty *white border square* (colour figure online)

Unlike in the previous test, and as already mentioned, the majority class is now the "regular" one, followed by the "empty" and "doughnut" classes, with only few cases of "fragmented" and "cracking" spots. It can be seen how again the minority classes tend to appear in clusters and the cells of the "empty" class mainly in the last row, thus confirming the assumption that they have been placed that way during the experiments.

Figure 8 shows the final results in test set 2, where the hits are represented by blue squares and errors by red squares. Most of the errors correspond to "regular" spots classified as "cracking" ones. This result may be due to the

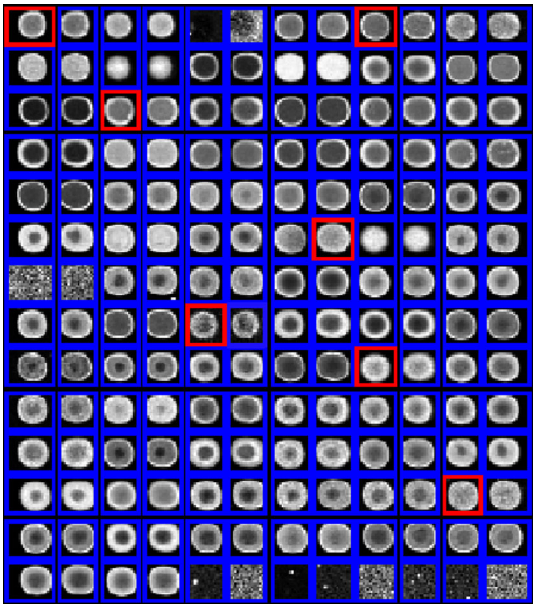


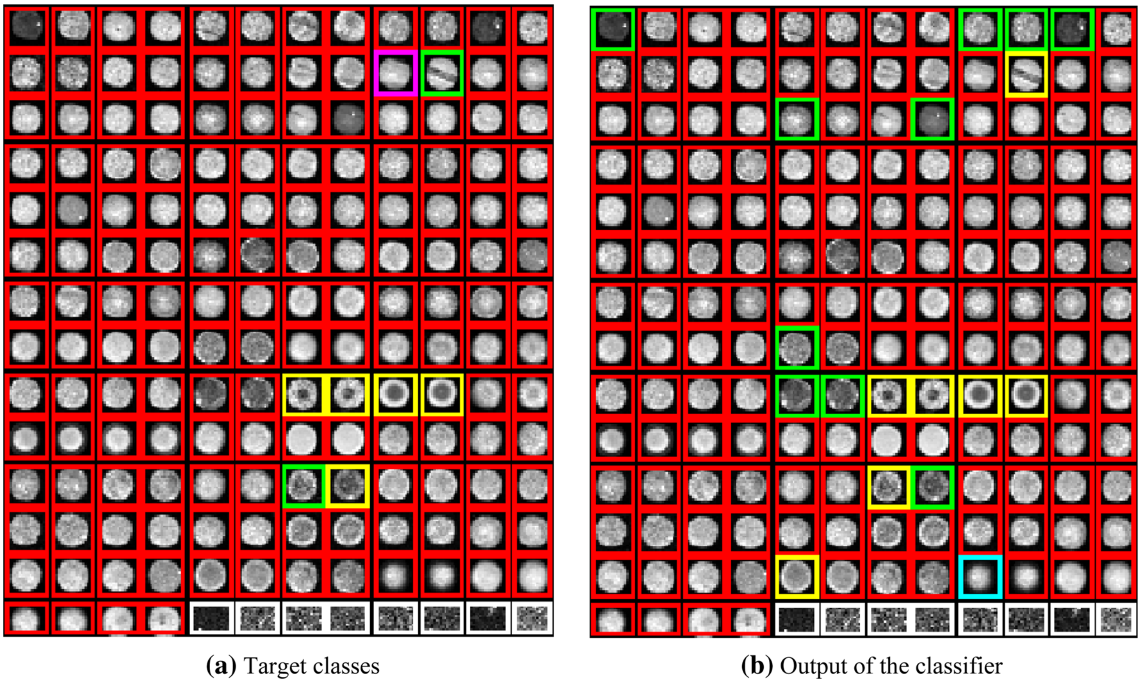
Fig. 6 Hits (*blue squares*) and errors (*red squares*) in the first series of testing (colour figure online)

granular aspect associated with low intensity spots; in addition some of them have noise. At the top of the image, there is a "fragmented" class spot whose fragmentation is really low that has been classified as a regular one. Also at the top part of Fig. 8, there is a "cracking" spot with the peculiarity that it is quite "regular", crossed by a well-defined dark line, right in the middle, which leads the classifier to mistakenly think that it belongs to the "doughnut" class. In the second row from the bottom of Fig. 9, there is a "regular" spot classified as "doughnut". This spot shows a very thin ring in its border. In the same row, there is a "regular" spot classified as "egg". This spot shows a small positive gradient in its intensities, from its border towards its centre.

As a general conclusion, the difference performance of the classifier on both testing series can be explained by the prevalence of different classes of spot in each one of the sub-grids. In the sub-grid of the first series the more abundant spot class is the doughnut one (Fig. 5), in which the classifier showed a very high hit rate during the training process. Nevertheless, in the sub-grid of the second testing set (Fig. 7), the more abundant spot class is the regular one, in which the classifier showed a slightly lower performance during the training. However, Fig. 8 shows how only very few spots were wrongly classified.

5.2 Evaluation of the classification robustness

For generalization purposes in the classification stage, we present an analysis that shows the robustness of the network models. In particular, a random cross-validation analysis is carried out to show that the performance of the models is independent of the set of data used for the generation of the neural networks.



(a) Target classes

(b) Output of the classifier

Fig. 7 Sub-grid of the second series of testing, target classes (**a**) and results of the neural classifier (**b**). Spot colour code: regular *red square*, doughnut *yellow square*, cracking *green square*, egg *light blue square*, saturated *blue square*, fragmented *fuchsia square* and empty *white border square* (colour figure online)

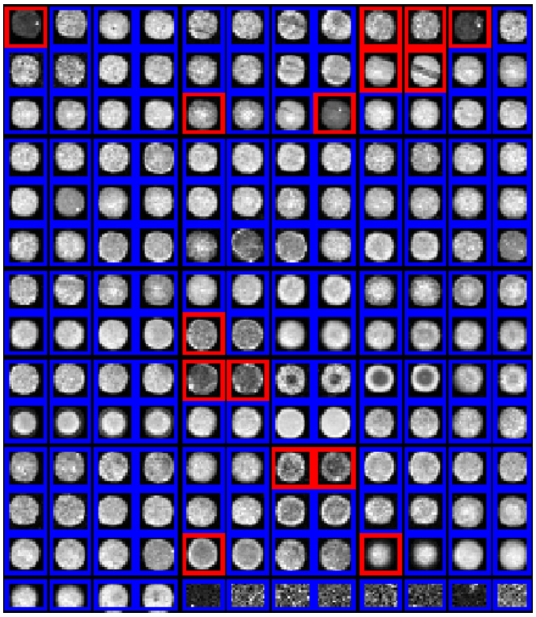


Fig. 8 Hits (*blue squares*) and errors (*red squares*) in the second series of testing (colour figure online)

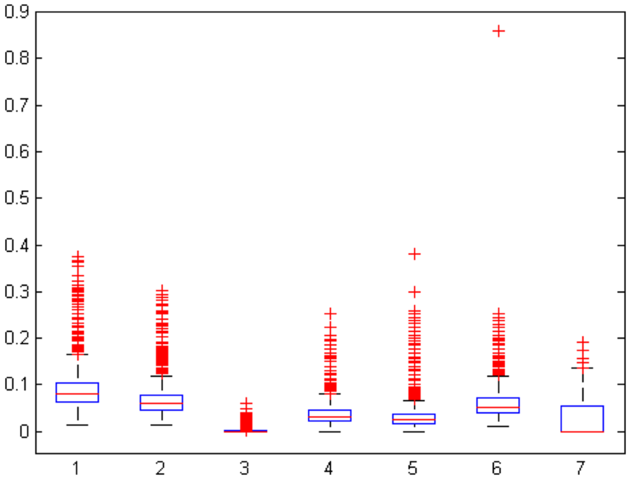


Fig. 9 Classification performance of the test set with 1000 partitions

Once the optimum number of neurons in the hidden layer is found, 1000 network models are estimated considering different sets for training (70%), validation (15%) and test (15%), each time. The validation set is used to stop the network training, while the test set is used to measure the classification performance. The sets are selected by randomly sampling the training dataset.

Figure 9 shows the error percentage over the test set for the 7 networks (1 = regular, 2 = cracked, 3 = saturated, 4 = doughnuts, 5 = egg, 6 = fragmented and 7 = empty). It can be observed that in all the networks the average error and the standard deviation are very small (below 0.4), being the 3rd one the best. This means that the descriptors are appropriate for our purpose.

6 Conclusions and future works

This work provides a pipeline for the processing of DNA microarray images. A new computing method for classifying the spots into morphology-derived classes is proposed. The classification is performed without previous segmentation, after the gridding process. The high accuracy classification rates obtained when tested on sub-grids extracted from real microarray DNA images prove the efficiency of this novel approach. A main conclusion is that the use of this classifier can be used to improve the segmentation process of DNA microarray images.

One of the main contributions is that we perform the classification of the spots into morphology-derived classes in order to assist the segmentation procedure that is traditionally performed after the gridding process. A new approximation for the classification of spots is presented, applying the idea of using the information of the whole cell, without segmentation.

Besides, instead of computing a reduced number of descriptors and showing its discriminant value for the classification, we perform the calculation of a great number of descriptors that are then reduced to a presumably optimal subset using the sequential forward selection algorithm [2].

Another contribution is that, based on the expert knowledge of the DNA microarray images classification, an ensemble of neural networks has been designed. This supervised neural classifier has a tree-like structure made up of seven MLPs. Each branch of the tree corresponds to a neural network specialized in the detection of a specific class. Besides, the configuration of each network has been optimized using an iterative algorithm that minimizes the classification error. Every MLP has been independently configured and trained.

The performance of the competitive classifier is validated with real microarray DNA images, where the final sub-classifiers compete for spot allocation to one class or to another. The neural classifier predicts the spot class with a very high degree of reliability.

The pre-processing of the images, using different scales of grey intensities depending on the class of spot to be detected, as well as the extraction of multiple features from each individual cell that has been later reduced to a supposedly optimal subset by the sequential forward selection algorithm, has helped to improve the performance of the classifier.

Another useful contribution of this work is the generation of two databases of cell images, 725 microarray images for training and 336 images for testing. That is, a total number of 1061 images that covers the whole spectrum of spot classes are now available for the scientific community interested in DNA microarray images processing.[5]

[5] Available at http://www.litrp.cl.

The very good results of this approach encourage further work. The classification errors manifest that the separation between classes is not always well defined, and there are spots that have characteristics of more than one class. This suggests considering the fuzzy approach, dealing with degree of belonging to different classes at the same time. Special attention must be paid to the effect of noise in the images, mainly for certain spot classes, such as the cracking one.

As the analysis of microarray experiment could lead to quantification of thousands of genes, another possible future research line is to consider how to improve and make lighter this quantification by developing adaptive segmentation algorithms [36].

Even if it is not one of the goals of this paper, as future works the performance of the proposed method could be evaluated against other techniques.

Acknowledgements The authors are grateful for the resources made available by the "Laboratorio de Investigaciones Tecnológicas en Reconocimiento de Patrones", Universidad Católica del Maule, Talca, Chile.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Brown PO, Botstein D (1999) Exploring the new world of the genome with DNA microarrays. Nat Genet 21:33–37
2. Alpaydin E (2004) Introduction to machine learning (Adaptive computation and machine learning series). The MIT Press, Cambridge
3. Angulo J (2008) Polar modelling and segmentation of genomic microarray spots using mathematical morphology. Image Anal Stereol 27(2):107–124
4. Angulo J, Serra J (2003) Automatic analysis of DNA microarray images using mathematical morphology. Bioinformatics 19:2003
5. Schumacher S, Muekusch S, Seitz H (2015) Up-to-date applications of microarrays and their way to commercialization. Microarrays 4(2):196–213
6. Draghici S (2003) Data analysis tools for DNA microarrays. CRC Press, Boca Raton
7. Álvarez-Ramos C, Nino E, Santos M (2013) Automatic classification of *Nosema* pathogenic agents through machine vision techniques and kernel based vector machines. In: Computing Colombian conference (8CCC). IEEE, pp 1–5
8. Wang Z, Zineddin B, Liang J, Zeng N, Li Y, Du M, Cao J, Liu X (2013) A novel neural network approach to cDNA microarray image segmentation. Comput Methods Programs Biomed 111(1):189–198
9. Li Q, Fraley C, Bumgarner RE, Yeung KY, Raftery AE (2005) Donuts, scratches and blanks: robust model-based segmentation of microarray images. Bioinformatics 21(12):2875–2882
10. Yang Y, Stafford P, Kim Y (2011) Segmentation and intensity estimation for microarray images with saturated pixels. BMC Bioinformatics 12(1):1–11
11. Glasbey CA, Forster T, Ghazal P (2007) Estimation of expression levels in spotted microarrays with saturated pixels. Stat Appl Genet Mol Biol 6(1):1–15
12. Bozinov D, Rahnenfuhrer J (2002) Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering. Bioinformatics 18(5):747–756
13. Giannakeas N, Karvelis PS, Exarchos TP, Kalatzis FG, Fotiadis DI (2013) Segmentation of microarray images using pixel classification—comparison with clustering-based methods. Comput Biol Med 43(6):705–716
14. Daskalakis A, Cavouras D, Bougioukos P, Kostopoulos S, Georgiadis P, Kalatzis I, Nikiforidis G (2007) Effective quantification of gene expression levels in microarray images using a spot-adaptive compound clustering-enhancement-segmentation scheme. In Computational science and its applications–ICCSA 2007. Springer, Berlin, pp 555–565
15. Shao G, Li T, Zuo W, Wu S, Liu T (2015) A combinational clustering based method for cDNA microarray image segmentation. PLoS ONE 10(8):e0133025
16. Belean B, Borda M, Ackermann J, Koch I, Balacescu O (2015) Unsupervised image segmentation for microarray spots with irregular contours and inner holes. BMC Bioinformatics 16(1):412
17. Biju VG, Mythili P (2015) Fuzzy clustering algorithms for cDNA microarray image spots segmentation. Procedia Comput Sci 46:417–424
18. Zhao Y, Wang G, Yin Y, Li Y, Wang Z (2016) Improving ELM-based microarray data classification by diversified sequence features selection. Neural Comput Appl 27(1):155–166
19. Nanni L, Lumini A, Brahnam S (2010) Advanced machine learning techniques for microarray spot quality classification. Neural Comput Appl 19(3):471–475
20. Garro BA, Rodríguez K, Vázquez RA (2016) Classification of DNA microarrays using artificial neural networks and ABC algorithm. Appl Soft Comput 38:548–560
21. Wang Z, Zineddin B, Liang J, Zeng N, Li Y, Du M, Liu X (2014) cDNA microarray adaptive segmentation. Neurocomputing 142:408–418
22. Wu H, Wang L, Zhang F, Wen Z (2015) Automatic leaf recognition from a big hierarchical image database. Int J Intell Syst 30(8):871–886
23. Alhadidi B, Fakhouri HN, AlMousa OS (2006) cDNA Microarray genome image processing using fixed spot position. Am J Appl Sci 3(2):1730–1734
24. Santos M, Cantos A (2010) Classification of plasma signals by genetic algorithms. Fusion Sci Technol 58(2):706–713
25. Vergara JR, Estévez PA (2014) A review of feature selection methods based on mutual information. Neural Comput Appl 24(1):175–186
26. Mery D, Pedreschi F, Soto A (2013) Automated design of a computer vision system for visual food quality evaluation. Food Bioprocess Technol 6(8):2093–2108
27. Haralick RM (1979) Statistical and structural approaches to texture. Proc IEEE 67(5):786–804
28. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal Mach Intell 24(7):971–987
29. Gonzalez R, Woods R (2008) Digital image processing, 3rd edn. Prentice-Hall, Upper Saddle River
30. Hu MK (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory 8(2):179–187
31. Funahashi KI (1989) On the approximate realization of continuous mappings by neural networks. Neural Netw 2(3):183–192
32. Peláez J, Doña J, Fornari J, Serra G (2014) Ischemia classification via ECG using MLP neural networks. Int J Comput Intell Syst 7(2):344–352

33. Foresee FD, Hagan MT (1997) Gauss–Newton approximation to Bayesian learning. In International conference on neural networks, 1997, vol 3. IEEE, pp 1930–1935

34. Hagan MT, Menhaj MB (1994) Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Netw 5(6):989–993

35. Daskalakis A, Glotsos D, Kostopoulos S, Cavouras D, Nikiforidis G (2009) A comparative study of individual and ensemble majority vote cDNA microarray image segmentation schemes, originating from a spot-adjustable based restoration framework. Comput Methods Programs Biomed 95(1):72–88

36. Athanasiadis E, Cavouras D, Kostopoulos S, Glotsos D, Kalatzis I, Nikiforidis G (2011) A wavelet-based Markov random field segmentation model in segmenting microarray experiments. Comput Methods Programs Biomed 104(3):307–315