A Novel Neural Network Approach to cDNA Microarray Image Segmentation

Zidong Wang[a,*], Bachar Zineddin[a], Jinling Liang[b], Nianyin Zeng[c], Yurong Li[c], Min Du[c], Jie Cao[d] and Xiaohui Liu[a]

Abstract

Microarray technology 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. Microarray images typically contain several thousands of small spots, each of which represents a different gene in the experiment. One of the key steps in extracting information from a microarray image is the segmentation whose aim is to identify which pixels within an image represent which gene. This task is greatly complicated by noise within the image and a wide degree of variation in the values of the pixels belonging to a typical spot. In the past there have been many methods proposed for the segmentation of microarray image. In this paper, a new method utilizing a series of artificial neural networks, which are based on multi-layer perceptron (MLP) and Kohonen networks, is proposed. The proposed method is applied to a set of real-world cDNA images. Quantitative comparisons between the proposed method and commercial software GenePix® are carried out in terms of the peak signal-to-noise ratio (PSNR). This method is shown to not only deliver results comparable and even superior to existing techniques but also have a faster run time.

Keywords

Artificial Neural Networks, Microarray image, Adaptive Segmentation, Kohonen Neural Networks

I. Introduction

Microarray technology has provided the biology community with a powerful tool for exploring the field of functional genomics. A robotic arrayer prints thousands of DNA sequences onto glass slides. Each slide typically contains several sub-grids that are two dimensional arrays of DNA spots, see Fig. 1. In the cDNA microarray experiment [1–5], Ribonucleic Acid (RNA) is first isolated from both control and experimental samples of the cells of interest. The extracted RNAs are then converted into cDNAs [6]. These two populations are usually labeled with fluorescent dyes such as Cy3 (Green) and Cy5 (Red). The mixture of these samples is then hybridized to a glass slide. After hybridization the slide is scanned with red and green laser. The difference in fluorescence between the two color channels shows the relative difference of gene's expression between the two sources [3]. The end product of the comparative hybridization experiment [7, 8] is a scanned array image. Gene's expression data is derived from these images and can be used for further steps of analysis [9].

Generally speaking, microarray image analysis consists of four stages [5, 10]: 1) the filtering stage which can be used to remove the small contaminations or the background trend [11]; 2) the spotting stage, usually called gridding, which is used for the individual localization of spots' areas; 3) the

[a] Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, United Kingdom. Email: `Zidong.Wang@brunel.ac.uk`, Fax: ++44/1895 251686.

[b] Department of Mathematics, Southeast University, Nanjing 210096, China.

[c] College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350002, China.

[d] School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China.
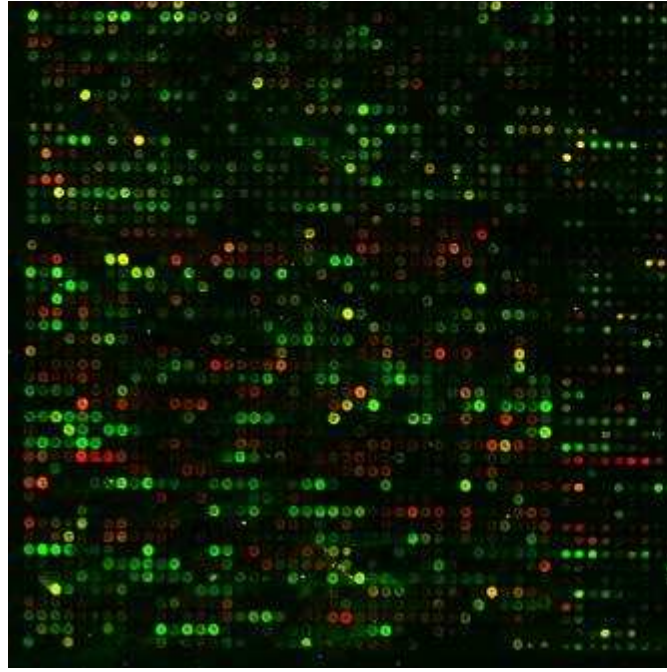
* Corresponding author.

Fig. 1. Two-Channel cDNA microarray image

segmentation stage which classifies pixels in a region immediately surrounding the gene as belonging to either the foreground or background domains; and 4) feature extraction stage which is the process of analyzing every spot in order to determine the corresponding gene expression level. For a given single region that is supposed to be the local area of a spot, the discrimination between the spot pixels and background pixels may at first seem a trivial one. Unfortunately, this is not the case. The spots within the image vary significantly in size and shape. The spot's pixels' values are typically inconsistent, and they may overlap with the values of the local background. In addition, the spot region may contain high valued noise artefacts, which should not be categorized as foreground and arguably not background either.

There have been many existing techniques proposed for dealing with the segmentation problem. In ScanAlyse [12] and GenePix [13] software, fixed and adaptive circle segmentation methods have been used. In these methods, all spots are assumed to be circular with a fixed or adaptive radius. However, the spots in the microarray image vary in size and shape and are not printed as precisely as they should be in the ideal case, and this gives rise to the main drawback of these methods. In the ImaGene [14] software, histogram-based segmentation method has been used. This technique assumes that the peaks in the histogram can be utilized in order to specify a threshold for the discrimination between the gene spot and the background regions. Unfortunately, this assumption could lead to incorrect observations. For example, in noisy images there could be no peaks (valleys) that can be used to infer a threshold as the range of intensities is very small. Therefore, it is almost impossible to find a threshold value or a set of threshold values that will result in a single connected region matching the set of spot pixels that a biologist would determine to be the spot pixels. In the Spot [15] software, the seeded region growing (SRG) algorithm has been employed [16]. SRG method is perhaps the most powerful one with respect to shape identification, but it relies on a seed growing methodology as well as the selection of the initial seeds [17].

It should be pointed out that the large number of spots (usually in thousands) as well as the

spot's shape and position irregularities [18, 19] could result in processing errors propagating through succeeding analysis steps [3, 20, 21]. It has been anticipated that the spot intensity value would be independent of the segmentation algorithm if a background correction method is used [5]. However, it has been proved in [22] that the segmentation method can significantly influence the identification of gene expression values and subsequent analysis. Furthermore, the time-consuming manual processing of the microarrays has led to the recent interest in using a fully automated procedure to accomplish the task [4, 23, 24]. Although recognition of spots in either control or experimental channels seems to be straightforward, the task is indeed complicated and challenging.

In order to accomplish the segmentation of cDNA microarray image, the idea of clustering over a full image area has been proposed by Bozinov and Rahnenfuhrer [25], but this might not be computationally feasible with current processing power. To overcome such a computational issue, an abstraction of the k-means clustering technique was proposed in [26]. Also, the active contour techniques were applied in [27] that are sensitive to noise. Other methods such as the applications of wavelets [21, 28] and Markov random fields [24, 29, 30] showed great promise as well. Lawrence et al. [31] presented a Bayesian approach in order to process images produced by these arrays that seek posterior distributions over the size and positions of the spots. Blekas et al. proposed a microarray image analysis framework in [32, 33] that provided a method of automatically addressing each spot area in the image by exploiting the Gaussian Mixture Model (GMM). Yet, these methods ignore the spatial dependencies among adjacent pixels. More recently, Fraser et al. [34] presented a Copasetic Clustering (CC) method that allows the clustering of large data sets (e.g. large images) by traditional clustering algorithms such as k-means, and Lukac et al. [20] utilized the multi-channel nature of the cDNA image data.

To summarize the discussions made so far, we have the following observations: 1) microarray image segmentation is an important yet challenging problem since the discrimination between the foreground and the background signal strongly affects the gene expression value for every spot; 2) improving the quality of the extracted gene expression is one of the ultimate goals of microarray image processing; 3) it is essential that the segmentation stage be given significant attention; 4) many microarray image segmentation (clustering) methods have been proposed with some producing better results than others. In general, the most effective approaches require considerable run time (processing) power to process an entire image; and 5) although many approaches have been proposed in the literature, there has been little progress on developing sufficiently fast, efficient yet effective algorithms to segment a microarray image by using up-to-date techniques. In search of a suitable algorithm catering for the above five observations, artificial neural networks (ANN) approach appears to a competitive candidate. As such, in this paper, we aim to propose a novel method segmenting microarray images with hope to produce results that are as good as, if not better than, the results of most advanced microarray segmentation algorithms *but with less running time.*

In this paper, a new segmentation method, which is based on artificial neural networks, is investigated. Towards the end of the paper, the method is tested on a set of real cDNA microarray images. Quantitative comparisons between the proposed algorithm and commercial GenePix® software are carried out in terms of subjective comparison methods. It is shown that the proposed technique is not only very powerful in clustering but also very efficient in terms of runtime. To deliver the proposed method and justify its advantages, the remainder of the paper is arranged as follows: 1) the method of gridding images used to create training and test data for the neural networks is briefly discussed; 2) the approach of determining the optimal number of spot region classes as well as the competitive neural network with Kohonen learning algorithm, which is used to classify the spot's region, is described; 3) the proposed ANN for the segmentation of the spot's local area is put forward; 4) the results of the

new clustering approach are analyzed; and 5) the results from this method are validated on real-world microarray images and also compared with the commercially available software GenePix®.

II. The proposed approach

The task of spot segmentation falls within the category of classification, that is, assigning pixels into spot and non-spot classes. Neural networks (NNs) are a well established tool for classification problems [35]. Once trained, NNs can produce very impressive classification results in a significant short runtime. NNs are motivated by an interest in modeling the working of neurons [36, 37], the cells that comprise the brain. Just like a brain, neural networks can be taught new skills on how to specifically recognize patterns. The training of a Multi-Layer Perceptron (MLP) neural network (with delta rule learning algorithm) involves the use of a training set, a set of input values and the desired response for each. A natural question is that how to obtain the quality training data in the case of the spot segmentation problem. The most logical answer is to train the network with output from one of the best spot segmentation algorithms developed to date, namely the CC approach [38]. In addition, GenePix® is used to produce another training set in order to validate the proposed method.

A. Creation of Training Sets

The training sets for the Neural Networks, which are used in the subsequent stages of our new segmentation approach, have to be created for the use of ANN. These sets contain: 1) a set of inputs as spots' regions - areas taken from a raw microarray image with each containing a single spot as well as some background and possibly noise pixels; and 2) the desired outputs, which are corresponding to these inputs, as binary images.

A Complete blind microarray image gridding framework developed in [39] is used to accomplish the spotting task. The input of the framework is the microarray image that can be at any resolution and the gridding is accomplished without any prior assumptions. The framework includes an Evolutionary Algorithm (EA) and several methods for various stages of the gridding process including sub-grid detection. Actually it is not critical which method is used to accomplish these tasks as long as the result is appropriate. Nevertheless, the results of [39] have proven to be both robust and effective for this task.

The image chosen features many of the characteristics that hinder traditional segmentation approaches. Such characteristics include, but are not limited to, poorly expressed spots, malformed spots, high valued noise artefacts and uneven background. A square region is taken around each spot centre as the spot region. These regions are taken from the raw microarray image to form the input set. Also, the corresponding regions are taken from the binary output of the CC clustering algorithm [38] in order to form the desired output set. A second output set is created by taking output of GenePix® package, which will be used to validate our developed ANN-based segmentation approach.

B. Single Neural Network Implementation

The first attempt made at segmentation is performed using a three layer neural network with Back-Propagation (BP) learning algorithm. This network uses 50% of the CC data set as training set and 30% of the same set for validation purposes.

The output of the neural network for a given spot region is a real valued image of the same size. The higher a given pixel value the more likely that it should belong to the foreground. Each output region is, by threshold, assigned into either foreground or background. The threshold value is obtained using Otsu's method [40]. A possible future development would be to replace Otsu's method with the more
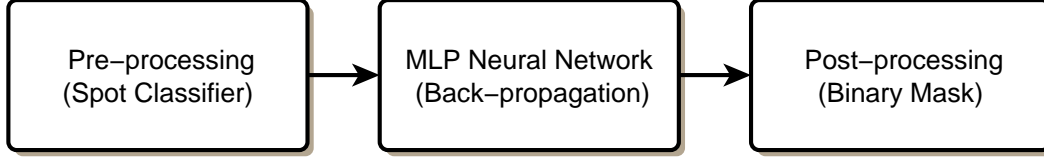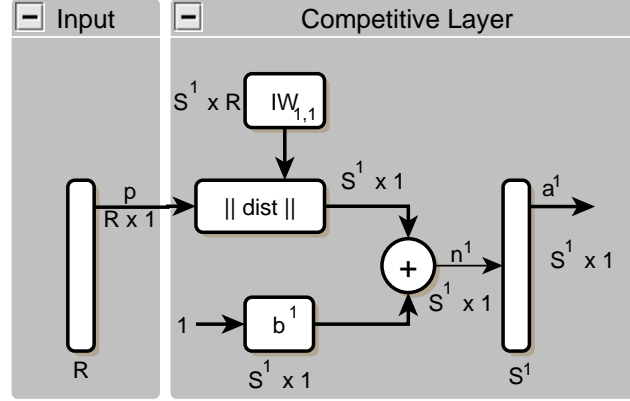
Fig. 2. Processing steps

Fig. 3. The architecture for the competitive later.

recent approach in [41] which claims to perform faster, although the improvement may be negligible due to the small size of the spot regions.

The trained network is then tested using approximately 20% of the spots from the CC data set. The outcome for the majority of the spots is a very poor match to the desired output. These results highlight the inability of the network to cope with poorly expressed spots whose values are close to or below their local background values.

With a typical microarray image, there is a wide variety of spots. Intensity wise, the spots range from ideal spots, which are valued well above their local background, down to very poorly expressed spots, which may be valued below their local background. Considering the spot's shape, the spots range from perfectly formed round shape spots with clearly defined edges down to disfigured spots with blurred edges and even disconnected regions. Clearly, it is inconceivable for a single BP neural network to learn how to segment all these contradictory types of spots. A way to simplify this task, with the ability to process all the spots within an image, is to classify the spots within a microarray image into one of the several classes. For each class, a BP neural network can achieve the segmentation for that specific class, see Fig. 2.

C. Spot Classifier

One of the most appropriate solutions to classify the spot's regions is to employ an unsupervised competitive network. This network seeks to find patterns/regularity in the input data. Figure 3 shows an illustration of the architecture of the competitive layer [42].

In order to obtain better results, the training data is pre-processed in the way depicted in Figure 4 as follows: 1) the values within each spot region are eliminated by applying a median filter with window size 3×3; 2) the values within each spot region are normalized between values of 0 and 1; and
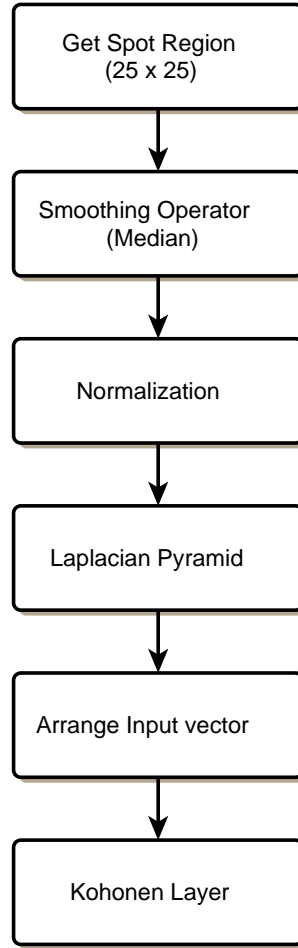
Fig. 4. Spot classifier.

3) a Laplacian pyramid filtering technique [43] is applied in order to extract gradient information of the cropped image. Therefore, less important information in the image has been eliminated.

The learning algorithm used to train the network is the Kohonen Learning algorithm (Winner-Take-All) [44, 45]. In this kind of learning, the units of the network update their weights by forming a new weight vector that is a linear combination of the old weight vector and the current input vector. Typically, the cell whose weight vector is closest to the input vector is allowed to learn. The measure of closeness/distance can be specified by two methods, both of which are based on the assumption that the weight vector of each cluster cell serves as an exemplar for the input vectors which have been assigned to that cell in the learning stage. In the first method, the smallest squared Euclidian distance between the input vector and weight vectors marks the winning cell. In the second method, the largest dot product marks the winning cell, where the dot product can be interpreted as a correlation measure between the input and weight vectors [46]. Using Kohonen training method the individual neurons of the network learn to specialize on ensembles of similar patterns; in so doing they become feature detectors for different classes of input patterns.

In order to determine how many classes the spot regions could be grouped into an iterative training approach is used. The network is trained to classify the spot regions into $3, 4, 5, ..., N$ classes. Working with all the spots from real microarray images, this method can define 9 classes of spot regions. Figure 5 shows ten spot regions from each of the 9 classes.
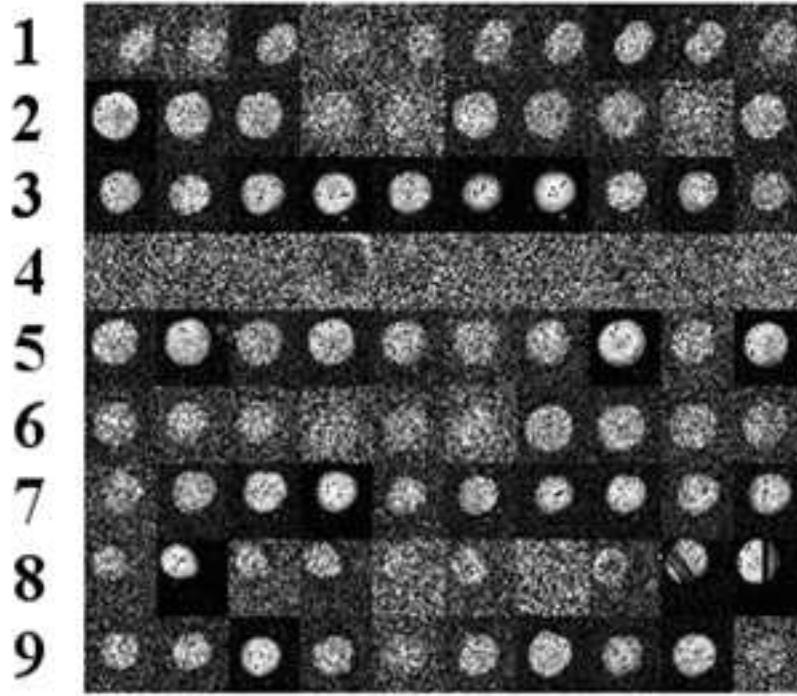
Fig. 5.   10 examples from each of the 9 classes of spots determined by the network.

By investigating the classification outputs, some key rules for the membership can be inferred such as the degree of noise, the size of spot, the position of the spot and the distribution of the pixels' values. However, in some of the classes the rules for membership are more obvious than others. Class 4, for example, is clearly the class that contains the noisiest spot regions while class 3 contains spots that are in high contrast to their background. Not only has the iterative training approach detected the number of classes (9) to divide the many spot regions within an image into, but it has also provided a means to classify quickly a given spot region into one of the 9 classes.

D. Multiple Neural Networks Implementation

Having determined the number of classes to divide the spot regions into (9) classes, the next task is to train a neural network for each class. This is accomplished in the same manner as the single neural network attempt described above. However, it is worth mentioning that each network is trained with its own custom training set where all the spot regions, within each training set, belong to the same spot region class.

One possible way to accomplish this task is to use Pattern Association [44]. In such a type of neural networks, learning is the process of forming association between related patterns. A key question for all associative nets is that how many patterns can be stored before the net starts to forget the learned pattern. Many factors influence the number of patterns that can be learned. The complexity of the patterns and the similarity of the input patterns, which are associated with significantly different output patterns, both play a role.

Based on the previous discussion, the associative nets are not a good solution for the addressed problem. Therefore, a new, partially connected and 3-dimensional MLP topology is proposed to tackle this problem, see Figure 6. In addition to the pre-processed intensity value for each pixel, additional inputs are determined to result in a better segmentation process. A 3×3 window is placed around every pixel, and the mean and standard deviation of the window are used as inputs to the network. The
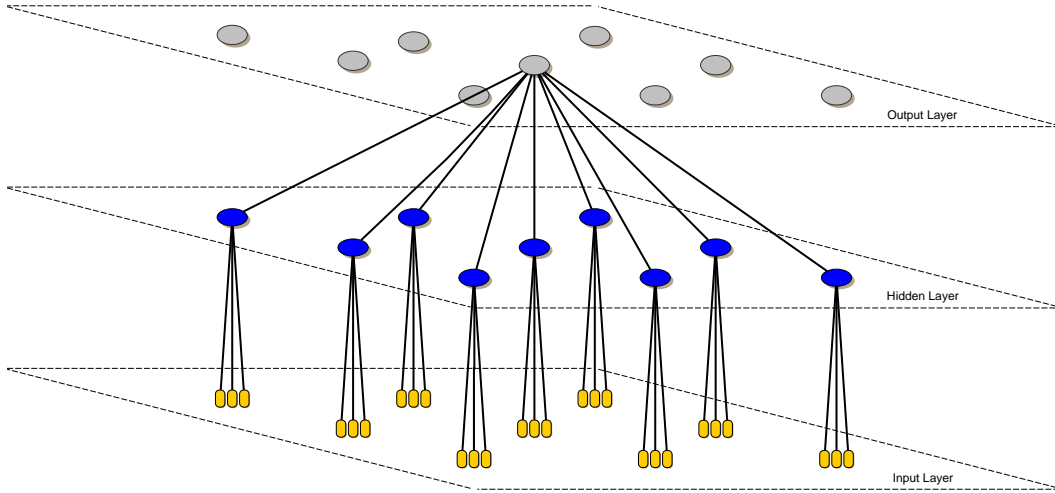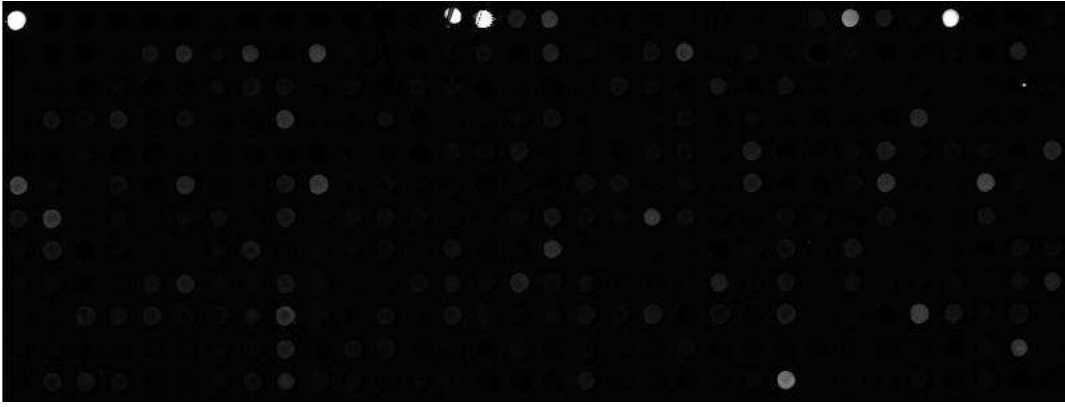
Fig. 6. Network architecture

Fig. 7. Sample input

network, therefore, features 1875 inputs, a hidden layer with 625 neurons and 625 outputs. Each of the output and hidden layers is arranged in 2D 25×25 array. More specifically, every pixel (represented by pixel's value, the mean and the standard deviation) is connected to one hidden cell. Every cell in the output layer is connected to the corresponding hidden cell and its neighboring cells located within a prescribed sphere of influence N_r of radius $r = 1$ centered at this hidden cell.

Two sets of nine neural networks are trained. The first is trained with the segmentation results of the GenePix® package, then the second is trained with the segmentation results from the CC algorithm. Training data is taken from the images (the raw image, GenePix® output image and the CC output image) used to produce an image with pixels' values falling within [0 1]. Therefore, further processing is required to get the final binary image.

In order to produce a binary mask, an Otsu's thresholding technique [47] is used. Figure 7 and Figure 8 show the input sample and its output.

III. EXPERIMENTAL RESULTS

In order to quantify the performance of different filtering methods, a quality measure is required in order to evaluate the validity of the pixels selected for a given spot. Note that the proposed algorithm produces a binary mask that classifies the pixels as belonging to either signal (the gene spots) or
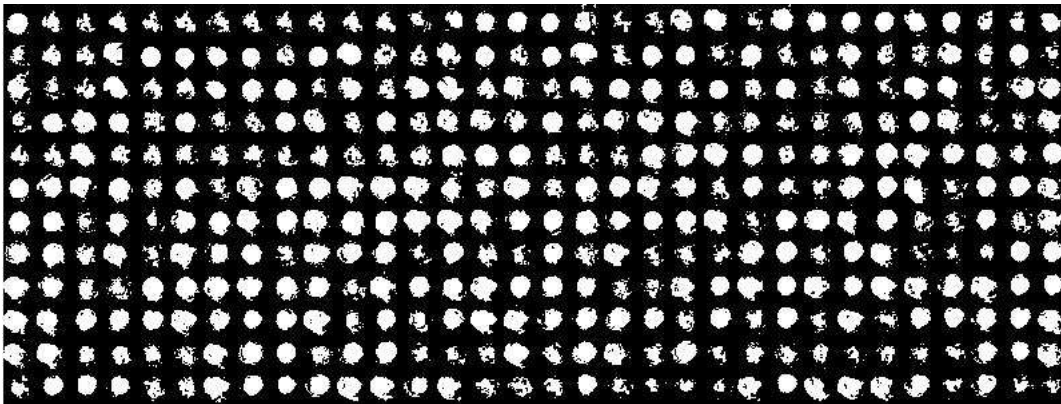
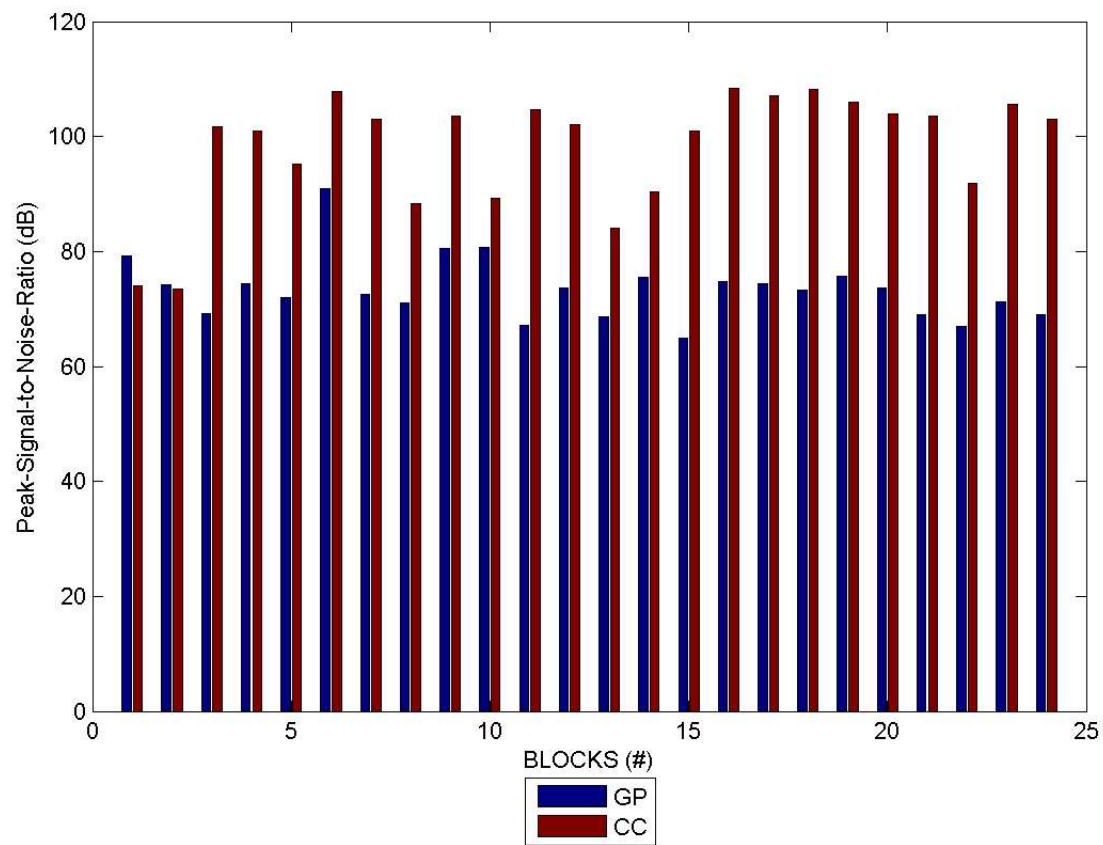Fig. 8. Example of the algorithm output.

Fig. 9. Comparison of segmentation results for 24 subgrids between GenePix (GP) and the copasetic clustering (CC) algorithm.

noise (the local background). For this purpose, an image quality measurement, known as the peak signal-to-noise ratio (PSNR) [48], is used and the rational is justified as follows.

The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics frequently used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image. The lower the value of MSE, the lower the error. The PSNR represents a measure of the peak error. The PSNR is most commonly used as a measure of quality of reconstruction of lossy compression codecs (e.g. for image compression). The signal in this case is the original data, and the noise is the error introduced by compression. Though a higher PSNR would normally indicate that the reconstruction is of higher quality, in some cases one reconstruction with a lower PSNR may appear to be closer to the original than another.

To compute the PSNR, the block first calculates the mean-squared error using the following equation:

$$MSE = \frac{\sum_{m,n}[I_1(m,n) - I_2(m,n)]^2}{m * n} \tag{1}$$

where m and m are the numbers of rows and columns in the input images, respectively, I_1 is the grayscale and I_2 is the mask image. Then, we obtain the PSNR using the following equation:

$$PSNR = 10 \log \left[\frac{R^2}{MSE}\right] \tag{2}$$

where R is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

The PSNR returns the ratio between the maximum value in the signal and the magnitude of the signal's background noise. The output of the PSNR is decibel units (dB). An increase of 20dB corresponds to a ten-fold decrease in the MSE difference between two images. The higher the PSNR value the more strongly the binary spot mask fits with the raw image surface. The PSNR is a much better measure of the suitability of a spot mask to the raw image area containing the spot as it is more resilient against large intensity ranges within the spot area compared to the MSE [38]. Figure 9 shows the PSNR comparison between the results of segmenting an image using the CC method and the GenePix® software. Comparisons are made between all 2 subgrids within the image. Clearly, using copasetic clustering gives a much better segmentation outcome.

Having trained the two sets of nine neural networks, we use them to segment the image that they are trained from. The results are shown in Figure 10, where the four columns represent the peak signal to noise ratio (PSNR) for, respectively, GenePix®, the neural network trained with the GenePix® data, CC and the neural network trained with the CC data. In the first two subgrids, the neural network results are approximately equal to the results of segmentation using GenePix® and the CC algorithm. Both neural networks even slightly surpass the results of the approaches from which they are trained in the second subgrid. In all other subgrids, the CC algorithm still produces the best results, but the NN results are clearly very competitive with the GenePix® results.

An advantage of the NN based approach is a computational time saving. Once a network has been trained it can be applied to many other images and achieve the task faster than GenePix® and CC approaches. Figure 11 shows the results of the clustering of 48 subgrids in an image. The neural networks used are the same ones as those used to produce Figure 10. The CC algorithm does produce the best segmentation result for all 48 subgrids, but the NN results are clearly comparable to the results obtained using the GenePix® package.
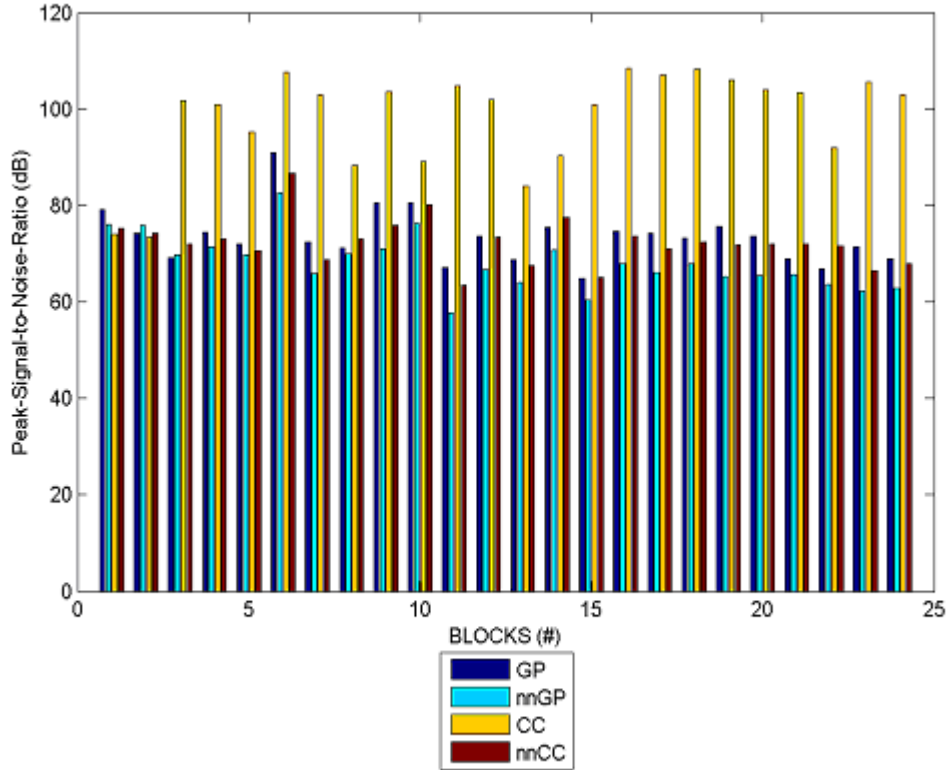
Fig. 10. Comparison of segmentation results between genepix, the neural network trained with the genepix data (nnGP), CC and the neural network trained with the CC data (nnCC).

IV. CONCLUSIONS AND FUTURE WORK

The task of spot segmentation is a critical one in the processing of any microarray image. Also, the impact of the accuracy and consistency with which spot pixels are identified is very significant. In this paper several methods of segmenting microarray images have been discussed. One of the most advanced approaches is the so-called copasetic clustering algorithm [38], which has been shown to surpass regularly the clustering results of the widely used GenePix® package that relies heavily upon manual interaction. The main contribution of this paper is the development of a new neural network based method for spot segmentation. This method not only produces very impressive results that are very competitive with the results obtained using the GenePix® package, but it can also produce the outputs more quickly than previous approaches. Clearly, this is a valuable addition to the area of microarray segmentation.

Future work in this area will allow an improvement on the results of the segmentations achieved using a neural network. It may even be possible to train a network to produce consistently better segmentation results than the CC algorithm.

In order to improve the algorithm, many alternatives could be utilized. Among others, we list three future topics here.

• Instead of the Kohonen network, self-organizing map competitive neural network [46] could be used where the neurons are placed at the nodes of a lattice that is usually one or two dimensional. The neurons become selectively tuned to various input patterns or classes of input patterns throughout a
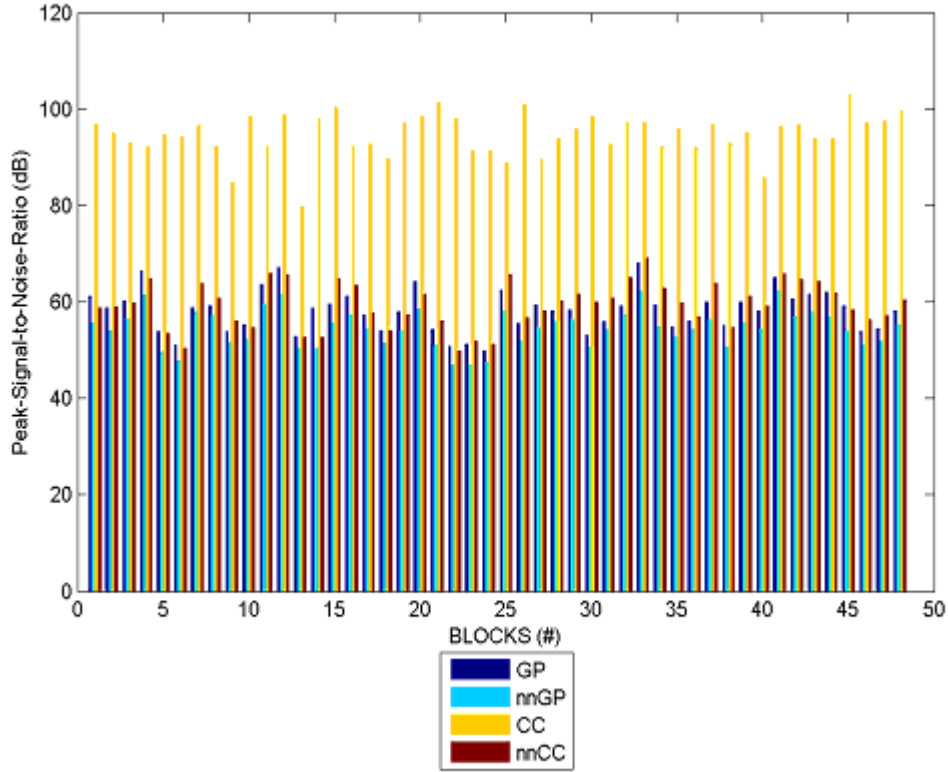
Fig. 11. Comparison of segmentation results for 48 subgrids. The neural networks used did not see this image during their training.

competitive learning process. The locations of the neurons so tuned become ordered with respect to each other in such a way that a meaningful coordinate system for different input features is created over the lattice. The self-organizing map is, therefore, characterized by the formation of a topographic map of the input patterns in which the spatial locations of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns. During the second stage, instead of using partly connected multilayer neural network, we could apply pixel based classification using a fully connected MLP network.

• A 5×5 window can be used to calculate some properties based on "sum and difference histogram". Then, these properties would be used as input for fully connected neural network with target value either from CC or GenePix® outputs. Furthermore, The employment of Maximum Covariance Technique may improve the generalization ability.

• Another possible use for the neural network approach presented here could be as a pre-processor to reduce the use of computationally more expensive algorithms such as CC. Therefore, NN could be used to segment an entire image and then to use the CC algorithm upon any spot regions where the NN returns a particularly low PSNR. By proving that the use of a neural network for the task of microarray segmentation is appropriate there are a number of avenues for future work in this area.

References

[1]   J. L. DeRisi, L. Penland, P. O. Brown, M. L. Bittner, P. S. Meltzer, M. Ray, Y. Chen, Y. A. Su, J. M. Trent, Use of a cDNA microarray to analyse gene expression patterns in human cancer, Nature Genetics, 14 (4) (1996) 457–460.

[2]   J. L. DeRisi, V. R. Iyer, P. O. Brown, Exploring the metabolic and genetic control of gene expression on a genomic scale, Science 278 (5338) (1997) 680–686.

[3]   M. B. Eisen, P. O. Brown, DNA arrays for analysis of gene expression, Methods in Enzymology 303 (1999) 179–205.

[4]   A. N. Jain, T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, D. Pinkel, Fully Automatic Quantification of Microarray Image Data, Genome Res. 12 (2) (2002) 325–332.

[5]   Y. H. Yang, M. J. Buckley, S. Dudoit, T. P. Speed, Comparison of Methods for Image Analysis on cDNA Microarray Data, Journal of Computational & Graphical Statistics 11 (2002) 108–136.

[6]   A. K. Whitchurch, Gene expression microarrays, IEEE Potentials 21 (1) (2002) 30–34.

[7]   S. K. Moore, Making chips to probe genes, IEEE Spectrum 38 (3) (2001) 54–60.

[8]   C. A. Orengo, D. T. Jones, J. M. Thornton, Bioinformatics: Genes, Proteins and Computers, Oxford, UK: BIOS Scientific Publishers, 2003.

[9]   J. Mata, R. Lyne, G. Burns, J. Bahler, The transcriptional program of meiosis and sporulation in fission yeast, Nature Genetics 32 (1) (2002) 143–147.

[10] M. Schena, D. Shalon, R. W. Davis, P. O. Brown, Quantitative monitoring of gene expression patterns with a complementary DNA microarray, Science 270 (5235) (1995) 467–470.

[11] E. Wit, J. McClure, Statistics for Microarrays: Design, Analysis and Inference, Wiley, 2004.

[12] M. B. Eisen, ScanAlayse, Online, March 2010. [Online]. Available: http://rana.lbl.gov/eisensoftware.htm

[13] *GenePix 4000 A User's Guide*, 1999.

[14] *ImaGene 6.1 User Manual*, 2008.

[15] M. J. Buckley, The Spot User's Guide, 2000.

[16] R. Adams, L. Bischof, Seeded region growing, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (6) (1994) 641–647.

[17] D. Tran, M. Wagner, Y. W. Lau, M. Gen, Fuzzy Methods for Voice-Based Person Authentication, Transactions of the Institute of Electrical Engineers of Japan 124 (10) (2004) 1958–1963.

[18] R. Lukac, K. N. Plataniotis, B. Smolka, A. N. Venetsanopoulos, cDNA microarray image processing using fuzzy vector filtering framework, Fuzzy Sets and Systems 152 (1) (2005) 17–35.

[19] X. H. Wang, R. S. H. Istepanian, Y. H. Song, Microarray image enhancement by denoising using stationary wavelet transform, IEEE Transactions on Nanobioscience 2 (4) (2003) 184–189.

[20] R. Lukac, K. N. Plataniotis, B. Smolka, A. N. Venetsanopoulos, A multichannel order-statistic technique for cDNA microarray image processing, IEEE Transactions on Nanobioscience 3 (4) (2004) 272–285.

[21] X. H. Wang, R. S. H. Istepanian, S. Y. Hua, Application of wavelet modulus maxima in microarray spots recognition, IEEE Transactions on Nanobioscience 2 (4) (2003) 190–192.

[22] A. Lehmussola, P. Ruusuvuori, O. Yli-Harja, Evaluating the performance of microarray segmentation algorithms, Bioinformatics 22 (23) (2006) 2910–2917.

[23] P. Bajcsy, Gridline: automatic grid alignment DNA microarray scans, IEEE Transactions on Image Processing 13 (1) (2004) 15–25.

[24] M. Katzer, F. Kummert, G. Sagerer, Methods for automatic microarray image segmentation, IEEE Transactions on Nanobioscience 2 (4) (2003) 202–214.

[25] D. Bozinov, J. Rahnenfuhrer, Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering, Bioinformatics 18 (5) (2002) 747–756.

[26] D. Bozinov, Autonomous system for web-based microarray image analysis, IEEE Transactions on Nanobioscience 2 (4) (2003) 215–220.

[27] T. Srinark, C. Kambhamettu, A microarray image analysis system based on multiple snakes, Journal of Biological systems 12 (2004) 127–157.

[28] H. Noda, M. N. Shirazi, E. Kawaguchi, MRF-based texture segmentation using wavelet decomposed images, Pattern Recognition 35 (4) (2002) 771–782.

[29] O. Demirkaya, M. H. Asyali, M. M. Shoukri, Segmentation of cDNA Microarray Spots Using Markov Random Field Modeling, Bioinformatics 21 (13) (2005) 2994–3000.

[30] S. Li, Markov random field models in computer vision, in: Computer Vision, Berlin / Heidelberg: Springer, 1994, pp. 361–370.

[31] N. D. Lawrence, M. Milo, M. Niranjan, P. Rashbass, S. Soullier, Bayesian processing of microarray images, In: Proc. 2003 IEEE 13th Workshop on Neural Networks for Signal Processing, pp. 71–80, Toulouse, France, September 17-19, 2003.

[32] K. Blekas, N. P. Galatsanos, I. Georgiou, An unsupervised artifact correction approach for the analysis of DNA microarray images, in: IEEE International Conference on Image Processing (ICIP 2003), N. P. Galatsanos, Ed., Barcelona, 2003, pp. 165–168.

[33] K. Blekas, N. P. Galatsanos, A. Likas, I. E. Lagaris, Mixture model analysis of DNA microarray images, IEEE Transactions on Medical Imaging 24 (7) (2005) 901–909.

[34] K. Fraser, P. O'Neill, Z. Wang, X. Liu, Copasetic analysis: a framework for the blind analysis of microarray imagery, IEE Proceedings Systems Biology 1 (1) (2004) 190–196.

[35] C. M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press.

[36] D. O. Hebb, The Organization of Behavior: A Neuropsychological Theory, Lawrence Erlbaum Associates Inc.

[37] W. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, The Bulletin of Mathematical Biophysics 5 (4) (1943) 115–133.

[38] K. Fraser, Z. Wang, X. Liu, Microarray Image Analysis: An Algorithmic Approach, Chapman & Hall/CRC, London, 2010.

[39] D. Morris, Blind Microarray Gridding: A New Framework, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38 (1) (2008) 33–41.

[40] N. Otsu, A threshold selection method from gray-level histograms, Automatica 11 (1975) 285–296.

[41] P.-S. Liao, T.-S. Chen, P.-C. Chung, A Fast Algorithm for Multilevel Thresholding, Journal of Information Science and Engineering 17 (2001) 713–727.

[42] H. Demuth, M. Beale, M. Hagan, Neural Network Toolbox5: User's Guide, 2007.

[43] P. Burt, E. Adelson, The Laplacian Pyramid as a Compact Image Code, IEEE Transactions on Communications 31 (4) (1983) 532–540.

[44] L. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, US: Prentice Hall, 1994.

[45] F. Ham, I. Kostanic, Principles of Neurocomputing for Science and Engineering, McGraw-Hill, 2000.

[46] S. Haykin, Neural Networks: A Comprehensive Foundation, US: Prentice Hall, 1999.

[47] MathWorks, Image Processing Toolbox5: User's Guide, 2007.

[48] B. Zineddin, Z. Wang, K. Fraser, X. Liu, Investigation on filtering cDNA microarray image based multiview anaalysis, in: the 14th International Conference on Automation & Computing, S. Zhang and D. Li, Eds. London, UK: Pacilantic International Ltd., 2008, pp. 201–206.