University of Windsor

Scholarship at UWindsor

Computer Science Publications

School of Computer Science

2005

Spot Detection and Image Segmentation in DNA Microarray Data

Li Qin

University of Windsor

Luis Rueda

University of Windsor

Ali Adnan

University of Windsor

Alioune Ngom

University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/computersciencepub>

Part of the [Computer Sciences Commons](#)

Recommended Citation

Qin, Li; Rueda, Luis; Adnan, Ali; and Ngom, Alioune. (2005). Spot Detection and Image Segmentation in DNA Microarray Data. *Applied Bioinformatics*, 4 (1), 1-11.

<https://scholar.uwindsor.ca/computersciencepub/7>

This Article is brought to you for free and open access by the School of Computer Science at Scholarship at UWindsor. It has been accepted for inclusion in Computer Science Publications by an authorized administrator of Scholarship at UWindsor. For more information, please contact scholarship@uwindsor.ca.

Spot Detection and Image Segmentation in DNA Microarray Data

Li Qin¹, Luis Rueda^{1*}, Adnan Ali² and Alioune Ngom¹

¹School of Computer Science

²Department of Biological Sciences

University of Windsor

401 Sunset Ave., Windsor, ON N9B 3P4, Canada

{qin1, lrueda, aali, angom}@uwindsor.ca

* Contacting author. Tel. +1 (519) 253-3000 Ext. 3780, Fax +1 (519) 973-7093

This article has been sent to Open Mind Journals for publication. When accepted for publication this version may be removed.

Abstract

The applications and development of microarray technology have been growing exponentially in the past few years, since their discovery in 1994. There are numerous applications of this technology, including clinical diagnosis and treatment, drug design and discovery, tumor detection, and in the environmental health research. One of the key issues in the experimental approaches that utilize microarrays is to extract quantitative information from the *spots*, which represent *genes* in a given experiment. For this process, the initial stages are quite important and influential in future steps of the analysis. Thus, identifying the spots and separating background from the foreground is a fundamental problem in DNA microarray data analysis. In this paper, we present an overview of the state-of-the-art methods for *microarray image segmentation*. We discuss the foundations of the circle-shaped approach, the adaptive shape segmentation, the histogram-based methods, and the recently introduced clustering-based techniques. We analytically show that the latter is equivalent to the one-dimensional standard well-known k -means clustering algorithm that utilizes the Euclidean distance.

Keywords

microarray data analysis, microarray image segmentation, DNA microarrays

1. Introduction

Microarray technology makes use of the sequence resources created by current genome projects and other sequencing efforts to identify the genes, which are expressed in a particular cell type or an organism (Brown et al. 1999). Measuring gene expression levels in variable conditions provides biologists with a better understanding of gene functions, and has wide applications in life sciences. For example, microarrays allow comparison of gene expression between normal and cancerous cells. The technology has been referred by various names: DNA microarrays, DNA arrays, DNA chips, and gene chips.

A microarray is typically a glass slide, onto which DNA molecules are attached at fixed locations, i.e. *spots*, each related to a single gene. Microarrays exploit the theory of preferential binding of complementary single-stranded DNA sequences (cDNA, for short), i.e. complementary single stranded DNA sequences tend to attract each other and the longer the complementary parts, the stronger the attraction (Brazma and Vilo 2000). Most of the microarray experiments compare gene-expression from two samples, one called *target* (or experimental) and the other is called *control*. The two samples are labeled by synthesizing single stranded cDNAs that are complementary to the extracted mRNA.

A typical microarray experiment is depicted in Figure 1. The spots are either printed on the microarrays by a robot, or synthesized by photolithography or ink-jet printing. After the target genes

are generated and laid out on the chip surface at defined positions, the cDNA extracted from two samples, labeled with fluorescence dyes, is hybridized to the chip. The result of the binding of cDNA is detected by fluorescence by laser excitation. In order to read out the comparative abundance of the cDNAs, Cy5 and Cy3 fluorescent probes are commonly used and compared. The dyes enable the amount of sample bound to a spot to be measured by the level of fluorescence emitted when excited by a laser and detected with a scanning confocal microscope. The relative intensity of Cy5/Cy3

Figure 1. The production of a microarray. The genes of interest are spotted on the surface of a glass slide. Two samples of mRNA are converted into cDNA by reverse transcription and labeled with fluorescent dyes. After the labeled sample is then hybridized to the microarray surface, the microarray chip is washed.

The result of a microarray experiment is represented as a vector, each element being a spot. The value of an element in the vector is the ratio of the estimated abundance of mRNA in two samples. Figure 2 shows a portion of a sample microarray image¹ that contains a few genes in an experiment where we used human cultured cell line to look at the toxicogenomic effects of two pesticides that are found in the rural drinking water. On the microarray, each gene corresponds to a spot in the composite image. The physical dimension of such an array is about one inch or less, and the spot diameter is of the order of 0.1mm, some microarray types being even smaller. For a given spot, if the mRNA from the sample in the condition that is labeled using Cy3 is in abundance, the spot will turn to be green. Otherwise, the spot will appear as red, if the mRNA from the sample in the condition that is labeled using Cy5 is in abundance. If both sample types are in equal abundance, then the spot will be appearing as yellow, whereas if neither sample is present it will appear black. Thus, from the fluorescence intensities and colors of each spot, the relative expression levels of the genes in both samples can be estimated. Figure 2 shows the color reflected by variable abundance of fluorescence intensities in the two channels.

¹ The total RNA was isolated from human cells (control and following pesticide treatments), reverse transcribed, labeled with fluorescent dyes, and then hybridized to human EST microarray. The figure shows a composite image of control and treated samples differentially labeled with Cy3 and Cy5 fluorescent dyes.

Figure 2. An enlarged illuminated microarray image. The picture shows a portion of a microarray chip, each spot standing for a gene. The color of each spot reflects the relative abundance of the two fluorescence intensities.

The raw data produced from microarray experiments are called the *hybridized microarray images*. To obtain information about gene expression levels, these images have to be analyzed: each spot on the array is identified, then its intensity measured and compared to the background. This process is called *image quantitation*. As microarray technology is still rapidly growing, it is natural that, at present, there are no established standards for microarray experiments and how the raw data should be processed. There are also no standard measurement units for gene expression levels.

Biological experiments that employ microarrays result in massive amounts of data. These data

can help biologists gain insights into underlying biological processes, only if they are carefully extracted and stored in databases, where they can subsequently be retrieved and analyzed. Some of the published data can be accessed publicly, such as Stanford yeast data, which covers 6,000 genes over 7 time points, and the Cancer Genome Anatomy Project (CGAP) data at NCI/NCBI.

The overwhelming use of bioinformatics tools for microarray analysis is a significant achievement in biology, because no other technology has used such sophisticated tools, has combined expertise from many different disciplines, and has provided such detailed information about bio-sequences.

As DNA microarray technologies emerge, they conform to simple, yet efficient tools for experimental explorations of genomic structures, gene expression programs, gene functions, and cell and organism biology. It is widely believed that gene expression data contain information that allows us to understand higher-order structures of organisms and their behavior. Besides their scientific significance, gene expression data have important applications in pharmaceutical and clinical research. For example, the comparison of the gene expression levels before and after cancer treatment may help identify the genes affected by the drug treatment. Thus, this process helps provide a quicker and more accurate diagnosis and the subsequent treatment of the disease.

Although microarrays are a new emerging technology, they have already been widely adopted, and many users are now going beyond exploratory studies. Microarrays are being exploited in human diseases, drug discovery, and genetic screening and diagnostics (Schena 2002). The most promising

commercial application of microarrays is their potential use in clinical diagnostics. Its potential application goes from drug discovery to gene-based diagnostics, and gene-based treatments. The most appropriate treatments can be reached by studying the dynamics of gene expressions over time, among tissues, and disease status. In addition, microarrays have a huge potential impact in the areas of preventative medicine, ability to diagnose accurately the disease, and design and screen the drugs that can be used to treat certain disease states.

2. Microarray Data Processing

Measuring gene expression levels in different conditions provides biologists with a better understanding of the cellular processes. In general, the analysis of DNA microarray gene expression data involves two steps. The first step is image quantitation, i.e. the extraction of gene expression data. The microarray image is processed in order to obtain the ratios of the intensities of the two probes for each gene. This is a very important step, since the accuracy of the resulting data is essential in posterior analyses. The second step is gene expression data analysis. After the ratios of the intensities are obtained, various methods can be applied to cluster the genes into different function groups based on the ratios retrieved in the first step.

In the past few years, more than twenty commercial software and free packages have been devised to extract the intensities of each gene and conduct further analysis. Some well-known tools are ScanAlyze (Eisen 1999), GenePix (Axon 1999), ScanArray Express (GSI 1999), and Spot

(Buckly 2000).

2.1 Scanning

The hybridized arrays are scanned to measure the fluorescence intensities of each sample for every spot in the gene array. These fluorescence intensities correspond to the level of hybridization of the two samples to the DNA sequences laid on the slide. The scanned original images are stored as a pair of 16-bit TIFF files, typically from 2.5 to 20MB in size, where one file corresponds to the testing sample and the other to the reference sample.

Gene expression levels in an array can vary in a wide range of order of magnitude. It is necessary to measure signals over a wide dynamic range, from saturated signals to signals that are lost in the background noise. Additionally, the integration and scaling of the two data sets involve difficult tasks. The two commonly used fluorescence intensities have different physical properties. The spots are widely separated relative to their individual size, so that the spots can be hybridized, washed, and scanned. These principles of microarray production lead to the fact that spots are not of the same size and some spots have high or low intensity. In Figure 3, a portion of a microarray image from the Apo A1 data (Callow et al. 2000) shows spots that have different sizes.

Figure 3. A portion of a sample microarray image obtained from the Apo A1 microarray data. It is typical that the sizes and the intensities of the spots vary in a wide range in a microarray image.

Using a fluorescent signal, the diameter of the spots printed on glass slide range from 25 to 100 μm (1000 μm = 1mm). The typical resolution is 3 to 5 μm per pixel. The minimum resolution of spots can be as low as 8 pixels/dimension, i.e. 64 pixels/spot. There are two main kinds of microarray chips, traditional microarrays and Affymetrix arrays. The size of a traditional microarray is 1x3 inches, i.e. 25x76x0.94 mm, whereas the size for an Affymetrix array is 0.5x0.5 inches. The range of quotient of brightest and dimmest signals is in a 1000-fold dynamic range, while reducing the background can increase the dynamic range when detecting signals (Schena 2002). The most commonly used devices are laser scanners equipped with photomultiplier tubes, whereas the most

advanced scanners can detect from 0.1 to 0.01 fluors/ μm^2 .

2.2 Addressing

The process of *addressing*, which is also called *gridding*, refers to the identification of the center coordinates of each spot. The basic structure of a microarray image is often provided by the manufacturer. This includes the number of sub-grids in the image, the number of rows and columns in each sub-grid, and often, the coordinate of a marker position. After gridding, each spot is identified and linked to a unique identifier. In Figure 4, we show a gridded microarray image. For better results, the foreground region should be located at the center of each grid.

The following issues need to be addressed when processing the image: the misregistration of the red and green channels, the skewness of the sub-grids in the image, and the rotation of the grid axes relative to the image. The task of addressing is often done manually or interactively, usually by drag-and-drop of the mouse, sometimes together with providing initial parameters. User intervention may make the process very slow. However, when automatic addressing is desired, the following problems also need to be considered: the contamination of the slide surface, the variety of the spot shape and size, and the highly dynamic signal intensities. Usually, an initial guess of such parameters is given as input for an automatic addressing techniques to start the process. Sample parameters include the number of sub-grids, columns and rows in a sub-grid, row spacing, row and column resolution, tip spacing, and spot width and height. A few recent studies focus on automatic

microarray image analysis (Katzner et al. 2002) (Jain et al. 2002) (Steinfath 2001). A reliable addressing procedure is desirable to ensure the accuracy of the subsequent process discussed below.

2.3 Segmentation

In general, segmentation of an image refers to the process of partitioning the image into several regions, each having its own properties (Soille 1999). In microarray image processing, segmentation

Figure 4. A gridded Apo A1 microarray image. Horizontal and vertical lines separate the spots. For the best results, the foreground should locate at the center of each grid.

refers to the classification of pixels as either the signal or the surrounding area, i.e. *foreground* or *background*. The microarray image contains noisy pixels that come from contamination during different stages when producing the chips or otherwise during the hybridization process. The process

of segmentation should be able to distinguish noisy pixels and true foreground pixels.

As the intensities of some spots are weak in nature, image processing of microarray data must be able to identify weak spots. The estimation of the signal background components is crucial, especially for weak spots, which in general correspond to rare transcripts that are extremely important biologically. It is well known in microarray analysis that a general rule for data extraction is that signals lower than 1.5-fold above the median background are considered unreliable (Skena 2002).

2.4 Background Correction

After the identification of spots, or the foreground, the next step in processing a microarray image is background correction. The pioneering paper on the statistical robustness of background correction and normalization is (Chen et al. 1997).

The measured fluorescence intensity of a spot includes a contribution that is not due to the hybridization of mRNA samples to the spotted DNA. Besides contamination, the background also contains information about scan and hybridization effects, which can be used to correct spot signals. To reduce bias, background correction is recommended. A common procedure is to subtract the background intensity from that of the foreground for each channel before calculating the ratio of the two channels, using the following equation:

$$I_t = I_f - I_b , \tag{1}$$

where I_t is the true intensity of the spot, i.e. the intensity after eliminating the influence of the contamination and scan effects and hybridization effects, I_f is the foreground intensity, which is measured in the foreground region, and I_b is the background intensity, which is measured in the background region.

Equation (1) sometimes results in negative true intensity values for weak spots, thus making the data for those spots unusable. An approach for improving background correction for low intensity spots using a Bayesian-based method can be found in (Koopberg et al. 2002).

Although recent experiments suggest that DNA on slides mask background, which means that the latter is not additive, local background correction is recommended for most microarray images because even small variations in the background signal can affect the ratio values. In fact, background correction is preformed by most of the microarray processing packages.

2.5 Normalization

Normalization is the process of removing the systematic variations that are introduced in microarray experiments, such as the differences of the labeling efficiency of the two dyes. It has been conjectured that normalization contributes more than background correction in the estimation of gene expression levels (Goryachev and Macgregor 2001). Normalization is conducted either within a single microarray slide or among multiple slides. Usually, the normalization factor is independent of the location, while in some cases it is not. Thus, this should be checked for each experimental

setup. Also, the identification of weak spots can improve the performance of normalization (Yang et al. 2001). Various normalization methods exist, most of them falling into one of the following categories.

i) *Housekeeping genes*: Chen *et al.* proposed this method in 1997 (Chen 1997). Housekeeping genes are selected based on both a biological basis and their experimental behavior. The ratio of housekeeping genes in the same group of experiments is close to 1.0. The main problem is how to identify such genes in the experiment. This method is useful when cells reach a steady state (Schuchhardt et al. 2000).

ii) *Control spots*: Exogenous mRNA are added as housekeeping genes to both target and control samples (Schuchhardt et al. 2000). The drawback of this approach is that an offset may be induced because all expression ratios are divided by an unknown factor, therefore, a systematic preference for incorporation of Cy5 or Cy3 may exist.

iii) *Constant majority*: Assumes that the majority of genes do not change their expression level. Rigorous treatment for this process can be found in (Chen et al. 1997). The corresponding coefficient of variation does not need to be estimated, and does not rely on knowing the subset of genes that is constant with the condition that genes are not spotted on the array in any particular order.

iv) *Integral balance*: Assumes that the total levels of gene expression in the target and control samples are the same, provided that cells will not significantly increase or decrease the total

level of mRNA transcription.

After performing the above-mentioned five steps, the ratio or logarithm of the ratio of the intensities for the two channels has to be extracted. Since the experiments are performed under different conditions, it is natural that the ratio between the intensities for the two channels are prone to what is called *systematic bias*. This type of bias is usually eliminated by performing the normalization process described above.

2.6 Gene Expression Data Analysis

While common *features*, or *patterns*, are regarded to be important for the biological functions of macromolecules, a wide range of domains is involved in analyzing sequence data, among which, the most prolific field is pattern recognition.

Whether the training samples are available or not, a pattern recognition system can fall into one of two categories: *supervised learning* or *unsupervised learning* (Duda et al. 2000). In supervised learning, each sample in the training set is labeled and the cost of mislabeling samples is given. After the system is adjusted to achieve the minimum cost according to the result of classifying the training samples, it is used to assign labels to unknown objects. In unsupervised learning, no prior knowledge about the object is assumed, and hence the system attempts to group datasets into *natural groups*, or *clusters*.

In supervised learning, support vector machines (SVMs) are one of the most successful methods

in gene expression data classification (see (Jaakkola et al. 1999), (Mukherjee et al. 1998), (Spellman et al. 1998), (Zien et al. 2000), (Cai et al. 2001), (Scholkopf et al. 2003), (Ding and Dubchak 2001), (Campanini et al. 2002), and (Guyon et al. 2002)). Gene expression data classification is typically a high-dimensional problem, i.e. it involves an overwhelming number of features compared to the relatively small number of samples. Because of the powerful features of SVMs and the nature of gene-expression data, SVMs are well suited for higher-dimensional classification problems.

In unsupervised learning, hierarchical clustering is one of the most widely used methods in gene expression analysis (see (Spellman et al. 1998), (Wen et al. 1998), (Alon et al. 1999), (Tibshirani et al. 1999), (Perou et al. 1999), (Furey et al. 2000), (Bittner et al. 2000), and (Bicciato et al. 2001)), even though Tamayo *et al.* (Tamayo et al. 1996) mentioned that hierarchical clustering lacks robustness and the solution may not be unique because the result is dependent on the order of the data. Apart from this drawback, hierarchical clustering has difficulties in dealing with noise.

Other pattern recognition techniques found in the literature include nearest-neighbor classifiers, aggregating classifiers, the naive Bayesian approach, self-organizing maps, hidden Markov models, k -means clustering, neural networks, evolutionary methods, simulated annealing, and classifier combination or fusion (Duda et al. 2000).

3. Microarray Image Segmentation

In this section, we discuss the microarray image segmentation problem. The aim is to partition the

microarray image pixels into different regions or groups. As a result, foreground pixels fall into one group, and background pixels fall into another group. There may exist other types of pixels, such as noisy pixels, which are contaminated pixels produced during microarray production and scanning, and should be excluded from the background or the foreground region during segmentation. Depending on the approaches to classify the pixels, another possible type of pixels includes the edge pixels surrounding the foreground region. Because the intensities of these pixels fall in between the foreground and the background, including or excluding them will lead to different signal to noise ratios. In a few words, it can be said that the goal of segmentation is to obtain the foreground intensity and background intensity of each spot in the microarray image. The problem can be stated more formally as follows.

Let R be an m -by- n integer-valued matrix that represents the image corresponding to Cy5, $\{R(i,j) \mid i=1,2,\dots,m; j=1,2,\dots,n\}$. Let G be an m -by- n integer-valued matrix that represents the image corresponding to Cy3, $\{G(i,j) \mid i=1,2,\dots,m; j=1,2,\dots,n\}$. A pixel is an element of an image. We use $R(i,j)$ to refer to the pixel p_{ij} at row i column j of an image R . We define I as the image obtained after combining R and G using some arbitrary function $f(.,.)$, i.e. $I(i,j) = f(R(i,j), G(i,j))$.

We assume that we deal with c clusters $\{\omega_1, \dots, \omega_c\}$, each representing one of the c categories of pixel intensities. In general, it is assumed that there are two clusters of interest, namely ω_1 and ω_2 , which represent foreground and background pixels respectively. In our context, we use a real-valued, d -dimensional feature vector $\mathbf{x}=[x_1, \dots, x_d]^T$ to represent the features (or information) that we can

extract from a pixel p .

The *segmentation problem* consists of assigning each pixel of R , G , or I , to one of the pre-defined classes, $\omega_1, \dots, \omega_c$. In particular, if we are dealing with the two-class problem, the result of the segmentation method will be a black and white or binary image B , $\{B(i,j) \mid i=1,2,\dots,m; j=1,2,\dots,n\}$ where $B(i,j)$ equals to either 0 or 255. After a label (or a class) is assigned to every pixel in the image, the foreground and background intensity can be computed using many different statistical measures of the two sets.

In general, image segmentation is the process of distinguishing objects from the background (Asano et al. 1996). Image segmentation is usually the first step in vision systems, and is the basis for further processing such as description or recognition. The goal of segmentation is to extract important features from images. Segmentation of an image can also be seen, in practice, as the classification of each image pixel to be assigned to one of the image compositions.

Most image segmentation approaches can be placed in one of five categories: clustering or threshold-based methods, boundary detection methods, region growing methods, shape-based methods, and hybrid methods. Many image segmentation approaches are intended for specific application domains to yield better results, for example, real-time image segmentation, color image segmentation, 3-D image segmentation, and motion image segmentation.

Clustering or thresholding methods are one of the earliest image segmentation techniques (see (Puzicha et al. 1999)). In these methods, the information about the pixel and its neighbors is used to

classify the pixel into one of the many regions.

Boundary detection or edge-based methods focus on contour detection. The image is segmented based on spatial discontinuity or edge finding and linking. This method is implemented as the convolution of mathematical gradient operators, or template matching operators, that use multiple templates at different orientations of the image. Sobel, Prewitt and Laplacian operators are examples of edge detection operators.

The region growing method performs image segmentation based on spatial similarity among pixels. The image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving the homogeneity or sharpness of the region boundaries.

Shape-based methods utilize some knowledge about the shape of the object to be segmented, e.g., mathematical morphology and template matching. Although many methods exist for general image segmentation, specialized methods have been designed for microarray image analysis. These methods are able to consider the characteristics of the microarray image. While there are quite a few, most of them discussed in this chapter, they are being perfected so as to maximize the information being extracted from the microarray image.

4. Circle-shaped Segmentation

4.1 Fixed Circle Segmentation

Fixed circle segmentation was first used in ScanAlyze. It assigns the same size (diameter) and shape (circle) to all the spots. It uses a constant-diameter circle as the shape of all the spots in the image.

GenePix and ScanArray Express also provide the option for fixed circle method.

Fixed circle method is simple to implement, and works well when all the spots are circular and have approximately the same size. However, the shape and size of spots vary in practice, and hence circular segmentation clearly cannot satisfy the needs. Figure 5 shows the resulting image after applying the fixed circle approach to the Apo A1 microarray image shown in Figure 3. We observe that some regions within the high intensity areas are left out of the foreground, and some regions within the low intensity areas are included in the foreground regions.

Figure 5. The segmentation of a microarray image from Apo A1 microarray data, which shows the limitations of the fixed circle approach. Even though the actual size of spots varies, the radius of all the spots is the same.

4.2 Adaptive Circle Segmentation

Roughly speaking, adaptive circle segmentation considers the shape of each spot as a circle, where the center and diameter of the circle are estimated for each spot. An implementation of this approach can be found in GenePix, ScanAlyze, ScanArray Express, Imagene, and Dapple (Buhler et al. 2000).

Adaptive circle segmentation involves two steps. First, the center of each spot needs to be estimated. Second, the diameter of the circle has to be adjusted. There are different implementations

for this approach. A typical edge detection technique, e.g. Laplacian transformation, can be applied to automatically estimate the diameter of the circle (Buhler et al. 2000). Another algorithm considers all pixels above a user-specified threshold to be foreground and finds the circle with the highest percentage of pixels that are foreground (Davidson 2003). As opposed to automatic approaches, some software packages allow the user to manually adjust the diameter of each spot.

The adopted algorithm for the adaptive circle segmentation in this paper is shown in Algorithm **Adaptive_Circle_Laplacian**. The centers of each spot are obtained as the center mean obtained from the weighted average of the pixel coordinates using the intensity of the pixel as the weight. Then, the Laplacian filter is applied in order to perform convolution on the image. Convolution provides a way of “multiplying together” two arrays of numbers, generally of different sizes, but having the same dimensionality, to produce a third array of numbers of the same dimensionality. Equation (2)

$$O(i, j) = \sum_{k=1}^m \sum_{l=1}^n I(i+k-1, j+l-1)K(k, l) \quad (2)$$

shows the function of convolution for two-dimensional images. After the edge pixels are detected, the diameter of the spots can be estimated by computing the median of the radius of the edge pixels.

The worst-case time complexity of **Adaptive_Circle_Laplacian** is proportional to the size of the image (i.e. $n \times m$, where n and m are the width and height of the image, respectively).

Algorithm 1 Adaptive_Circle_Laplacian

Input: a pair of microarray images, G and R .

Output: a binary image, B .

Method:

$I(i, j) \leftarrow G(i, j) + R(i, j)$

```

k1 ← CreateFilter('ave',[3,3])
I2 ← Convolution(I,k1) // smooth the image
k2 ← CreateFilter('log',[5,5],0.5)
I3 ← Convolution(I2,k2) // apply Laplacian of Gaussian filter
for i ← 1 to m do // count zero crossings
    for j ← 1 to n do
        edge(i,j) ← CountZeroCrossing(I3) // find the pixel where the second derivatives is zero
    endfor
endfor
for n ← 1 to numberOfSpots // find the radius of the foreground region
    for i ← 1 to spotWidth
        for j ← 1 to spotHeight
            if edge(i,j) == 255 then
                distance(i,j) ← GetDistance(i,j)
            endif
        endfor
    endfor
endfor
radius ← Mean(distance)
for i ← 1 to spotWidth
    for j ← 1 to spotHeight
        if distance(i,j) < radius then
            B(i,j) ← 255
        else
            B(i,j) ← 0
        endif
    endfor
endfor
endfor
end Algorithm Adaptive_Circle_Laplacian

```

One of the drawbacks of the Laplacian transformation is its extreme sensitivity to noise. Prior to the Laplacian transformation, the image has to be enhanced by applying a smoothing operator, such as the average filter, the Gaussian blur, or the Laplacian of Gaussian (LoG) filter in a single step. In our algorithm, we apply both the average filter and the LoG filter. Using the detected edges to estimate the diameter can avoid the drawback of disconnectivity of Laplacian filters. By smoothing the pictures and carefully selecting the parameters of the filters, the typical “double edge” problem of

the Laplacian filter can also be avoided. Figure 6 depicts the resulting image after applying our adaptive circle segmentation algorithm to the sample Apo A1 microarray image shown in Figure 3.

The main drawback of the adaptive circle segmentation algorithm is that it restricts the shape of the spots to be circles. However, the spots in a microarray image can take shapes including ellipses,

Figure 6. The result of our proposed adaptive circle method applied to an Apo A1 microarray image. The centers of every spot are calculated as the weighted mean of coordinates, and the radii of the spots are estimated by applying Laplacian transformation.

donuts, and other irregular shapes.

5. Adaptive Shape Segmentation

Seeded region growing (SRG) and watershed are common techniques that deal with different shapes in image segmentation. In SRG, the regions grow outwards from the seed points, preferentially, based on the difference between the pixel value and the running mean of values in an adjoining region (Adams and Bishop 1994). These two methods require an initial point to be known, which is called the *seed*. One of the advantages of using SRG in microarray image processing is that the location of foreground pixels and background pixels can be estimated.

SRG has been implemented in microarray processing package Spot (Buckly 2000). In this package, the foreground seed is chosen as the center pixel of the horizontal and vertical grid line. To avoid the situation when the spot is small and the grid center is slipped out of the spot foreground, a small numbers of $n \times n$ square pixels, whose center has the maximum intensity in a small area around the grid center, are taken as foreground seeds. The background seed is chosen as the point in which the grid lines intersect. After obtaining the seeds, the process is repeated simultaneously for both foreground and background regions until all the pixels are assigned to either foreground or background. Those pixels that are adjacent to a region are assigned first according to its intensity (Yang et al. 2002).

Figure 7. Spot's result of seeded region growing on an Apo A1 microarray image, which is the same image used in Figures 5 and 6. The shapes of spots are not restricted to circles.

Although SRG can be applied to microarray images containing spots of any shape and size, it has the disadvantage that the spots generated are generally smaller than the actual size. Refer to Figure 7 in which, the software package Spot is used to process the same Apo A1 microarray image as in Figures 5 and 6. We observe that the resulting shapes of the spots are irregular.

6. Histogram-based Segmentation

Using histograms to classify a pixel into either foreground or background is a simple and intuitive

idea. In (Chen et al. 1997), a new segmentation method is introduced, which uses a circular target mask to cover all the foreground pixels, and computes a threshold using Mann-Whitney test. If the pixel intensity is greater than a certain threshold, it is assigned to the foreground region; otherwise it is assigned to the background region. They use a statistical test to find sets of signal pixels with intensities significantly different from the local background. The application of Chen's method can be found in ScanArray Express, which also provides another version of a histogram method that uses a square target mask, and defines the ratio of foreground and background as the mean intensities between predefined percentile values, usually 5%-20% for background, 80%-95% for foreground.

Histogram methods are simple in concept, but a suitable size of the mask is difficult to choose in comparison with the spot size. When the mask is too small, it may not cover all the foreground pixels. In contrary, when the mask is too large, it may overlap with neighboring spots. According to Yang *et al.*, Chen's method is not as accurate as other recently introduced segmentation methods (Yang et al. 2002).

7. Clustering-based Segmentation

Clustering is one of the pioneering approaches in image segmentation. The idea of clustering can be summarized as follows. Consider a dataset $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{id}]^t$ is a d -dimensional feature vector representing the *features* of each pixel in an image. Clustering in the feature space attempts to find an indicator of similarity of the input samples, and has been

successfully used for segmentation purposes.

Clustering methods have some advantages when applied for microarray image segmentation, since they are not restricted to a particular shape and size for the spot. Although a clustering method has been recently proposed in microarray image analysis (Wu and Han 2003), no commercial microarray processing software has adopted this method yet.

Wu *et al.* used a k -means clustering algorithm in microarray image segmentation, which we refer to as k -means single-feature clustering microarray image segmentation (KSCMIS). They attempt to cluster the pixels into two groups, one for foreground, and the other for background. Thus, in KSCMIS, the feature vector is reduced to a variable in the Euclidean one-dimensional space. The

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_i, \quad x_i \in \omega_j \quad (3)$$

first step of KSCMIS consists of initializing the class label for each pixel and calculating the means for each cluster. Let x_{min} and x_{max} be the minimum and maximum values for the intensities in the spot. If $|x_i - x_{min}| > |x_i - x_{max}|$, x_i is assigned to foreground, or equivalently the label of the pixel x_i is set to 1. Otherwise, x_i belongs to background, thus x_i is labeled 0. After this process, the *mean* (or centroid) for each class, foreground or background, is calculated as follows:

Despite this method requires initialization and an iterative process, it is quite efficient in practice. After the initialization, the second step of the algorithm is the re-calculation of the means and the adjustment of the label of each pixel by the following criteria.

Assign $\omega_i = 2$ for all the x_i whose label is 1, if

$$\frac{N_1 * |x_i - \mu_1|}{N_1 - 1} > \frac{N_2 * |x_i - \mu_2|}{N_2 + 1}, \quad (4)$$

Figure 8. The result of KSCMIS on an Apo A1 microarray image. The foreground and the edges after eliminating the noisy pixels using LCR method show that the shapes of the spots are not restricted to circles.

otherwise assign $\omega_i = 1$. This step is repeated until no change in the means has been observed.

After implementing KSCMIS, we found out that using this method alone may not deal well enough with noisy data and weak spots, and improvements can be made to achieve better accuracy. There are many techniques that can be used for clustering. By choosing a suitable method and tuning the parameters, efficient models suitable for microarray image segmentation can be derived.

One of the properties of KSCMIS is that for a large number of pixels, its clustering method is equivalent to the standard k -means algorithm. This result is shown below. In the theorem, we assume

that the feature vectors lie in the one-dimensional Euclidean space.

Theorem 1: Let $D = \{x_1, \dots, x_n\}$, which has to be clustered into two classes. If $n \rightarrow \infty$, KSCMIS produces the same results as the standard k -means algorithm, where the Euclidean distance is used.

Proof: Let $n = N_1 + N_2$. Since empty clusters are not allowed, then it is true that as $n \rightarrow \infty$, it implies that $N_1, N_2 \rightarrow \infty$. We can then write the asymptotic behavior of (4) as follows:

$$\lim_{N_1, N_2 \rightarrow \infty} \frac{N_1 |x_i - \mu_1|}{N_1 - 1} > \frac{N_2 |x_i - \mu_2|}{N_2 + 1} \quad (5)$$

$$= |x_i - \mu_1| > |x_i - \mu_2| \quad (6)$$

Additionally, it is straightforward that, in the one-dimensional Euclidean space, (6) is equivalent to $(x_i - \mu_1)^2 > (x_i - \mu_2)^2$. Clearly, Equation (6) is the criterion used by the standard k -means algorithm, and thus, the result follows.

It is important to remark that the result of Theorem 1 is also applicable to the case in which N_1 and N_2 are large, but not necessarily ∞ . In typical microarray images, N_1 and N_2 can take values, for

$$1.005 |x_i - \mu_1| > 0.995 |x_i - \mu_2| \quad \cong \quad |x_i - \mu_1| > 0.990 |x_i - \mu_2|$$

example, above 200, and, thus, (5) results in:

This example shows that, in practice, KSCMIS produces similar results to those that would have been delivered by a clustering-based microarray image segmentation approach that implements the standard k -means algorithm.

8. Discussions and Future Work

Microarray image segmentation is a specific sub-field in image segmentation. Although many methods exist for image segmentation, in general, custom-designed methods for microarray image segmentation are desirable to achieve better accuracy by considering the characteristics of the microarray image.

Methods found in the literature can be grouped into five categories: fixed-circle method, adaptive circle approach, adaptive shape techniques, histogram-based methods, and clustering-based methods. The first two methods that we discuss in this paper are shape-based segmentations techniques. While the first one is too simple and naive to produce good results, the adaptive circle method achieves better results for circle-shaped spots. The histogram, adaptive shape, and clustering-based methods do not restrict to the shape and size of the spots to a specific form or value. Adaptive shape methods, in general, produce smaller foreground area than the actual spots. Histogram methods have been found to obtain good results, but suffer from the difficulty in choosing a suitable mask size. When using clustering-based techniques to separate background from foreground in microarray images, the current methods can not deal well enough with noisy data and weak spots. By choosing a suitable method and tuning the parameters, some models suitable for microarray image segmentation can be derived.

As we have mentioned that microarray technology is still growing rapidly, there are no

established standards for microarray experiments and how the raw data should be processed. There are quite a few problems that remain open and deserve investigation. In the microarray image segmentation, it is not enough to just group the pixels into foreground and background. In some cases, the noisy pixels also need to be identified. We suggest that techniques such as Adaline, or SVMs, can be used to detect noise. In the process of background correction, an adaptive formula could be used to reveal the true foreground intensity instead of using the rigid formula described in Equation (1). Neural networks could also be applied to obtain the true foreground intensity. When conducting the gene expression data analysis, although hierarchical clustering is a widely used method, it suffers from drawbacks such as dealing with noise and providing a non-unique solution. These two problems are currently being investigated.

References

- Adams R, and Bishop L. 1994. Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641-647.
- Alon U, Barkai N, Notterman D, Gish K, Ybarra S, Mack D, and Levine A. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays. *Cell Biology*, 96:6745--6750.
- Ansorge W, and Quackenbush J. 2001. Quick napkin calculations in Schnookeloch in Heidelberg, October 2001.

Callow MJ, Dudoit S, Gong EL, Speed TP, and Rubin EM. 2000. Microarray expression profiling identifies genes with altered expression in HDL deficient mice. *Genome Research*, Vol. 10, No. 12, p. 2022-2029.

Asano T, Chen D, Katoh N, and Tokuyama T. 1996. Polynomial-time solutions to image segmentation, *Proc. of the 7th Ann. SIAM-ACM Conference on Discrete Algorithms*, 104-113.

Axon Instruments, Inc. 1999. GenePix 4000A: User's manual.

Bicciato S, Pandin M, Didone G, and Di Bello C. 2001. Analysis of an associative memory neural network for pattern identification in gene expression data. *In Proceedings of BIOKDD'01*, 22-30.

Bittner M, Meltzer XP, Chen XY, Jiang Y, Seftor E, Hendrix M, Radmacher M, Simon R, Yakhini Z, Brazma A, and Vilo J. 2000. Gene expression data analysis. *FEBS Letters* 480:17-24.

Brown P, and Botstein D. 1999. Exploring the new world of the genome with DNA microarrays. *Nat Genet* Jan; 21(1 Suppl):33-37.

Buckly M. 2000. The Spot user's guide. CSIRO Mathematical and Information Sciences.

Buhler J, Ideker T, and Haynor D. 2000. Dapple: Improved Techniques for Finding Spots on DNA Microarrays. *Technical Report UWTR 2000-08-05*, University of Washington.

Cai Y, Liu X, Xu X, and Zhou G. 2001. Support Vector Machines for predicting protein structural class, *BMC Bioinformatics* 2 (1):3.

Campanini R, Dongiovanni D, Lanconelli N, Palermo G, and Riccardi A. 2002. A Support Vector Machine Classifier based on Recursive Feature Elimination for Microarray Data in Breast Cancer

Characterization. *AIIA Journal*, 28-30.

Chen Y, Dougherty E, and Bittner M. 1997. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *Journal of Biomedical Optics*, 2:364-374.

Ding C, and Dubchak I. 2001. Multi-class protein fold recognition using support vector machines and neural networks, *Bioinformatics*, 17:349-358.

Duda R, Hart P, and Stork D. 2000. Pattern Classification, 2nd Edition, Wiley-Interscience.

Eisen M. 1999. ScanAlyze User Manual.

Furey T, Cristianini N, Duffy N, Bednarski D, Schummer M, and Haussler D. 2000. Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data, *Bioinformatics*, 16(10):906-914.

Goryachev A, Macgregor P, and Edwards A. 2001. Unfolding of microarray data. *Journal of Computational Biology*. 8(4):443-461.

GSI Lumonics. 1999. QuantArray Analysis Software, Operator's Manual.

Guyon I, Weston J, Barnhill S, and Vapnik V. 2002. Gene Selection for Cancer Classification using Support Vector Machines, *Machine Learning* 46(1/3): 389-422.

Jaakkola T., Diekhans M, and Haussler D. 1999. Using the Fisher kernel method to detect remote protein homologies. In T. Lengauer et al., editors, *Proc. 7th Int. Conf. on Intelligent Syst. for Molecular Biology (ISMB-99)*, 149-158.

Jain A, Tokuyasu T, Snijders A, Segraves R, Albertson D, and Pinkel D. 2002. Fully Automatic

Quantification of Microarray Image Data. *Genome Res.*,12(2):325–332.

Katzer M, Kummert F, and Sagerer G. 2002. Robust Automatic Microarray Image Analysis. In *Proceedings of the International Conference on Bioinformatics:North-South Networking*, Bangkok.

Kooperberg C, Fazzio T, Tsukiyama T. 2002. Improved background correction for spotted DNA microarrays, *J Comput Biol.*9(1):55-66.

Davidson College. MicroArray Genome Imaging and Clustering Tool MAGIC Tool User Guide, Department of Biology.

Mukherjee S, Tamayo P, Mesirov J, Slonim D, Verri A, and Poggio T. 1999. Support vector machine classification of microarray data. *Technical Report* 182, AI Memo 1676, CBCL.

Perou C, Jeffrey S, van de Rijn M, Rees C, Eisen M, Ross D, Pergamenschikov A, Williams C, Zhu S, Lee J, Lashkari D, Shalon D, Brown P, and Botstein D.1999. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. *Proc. Natl Acad. Sci. USA*, 96, 9212-9217.

Puzicha J, Buhmann J, and Hofmann T. 1999. Histogram Clustering for Unsupervised Image Segmentation, *Computer Vision and Pattern Recognition*, Volume 2, 2602-2608.

Schuchhardt J, Beule D, Malik A, Wolski E, Eickho H, Lehrach H, and Herzelt H. 2000, Normalization strategies for cDNA microarrays. *Nucleic Acids Res*, 28:E47.

Schena M. 2002. Microarray analysis. John Wiley & Sons.

Scholkopf B, Guyon I, and Weston J. 2003. Statistical learning and kernel methods in bioinformatics.

In P. Frasconi and R. Shamir, editors, *Artificial Intelligence and Heuristic Methods in Bioinformatics*,

IOS Press, 1-21.

Soille P. 1999. Morphological Image Analysis: Principles and Applications. Springer.

Spellman P, Sherlock G, Zhang M, Iyer V, Anders K, Eisen M, Brown P, Botstein D, and Futcher B. 1998. Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell*, Vol. 9, 3273-3297.

Steinfath M, Wruck W, and Seidel H. 2001. Automated image analysis for array hybridization experiments. *Bioinformatics -OXFORD-*, 2001, Vol. 17, T. 7,S. 634-641.

Tamayo P, Slonim D, Mesirov J, Zhu Q, Kitareewan S, Dmitrovsky E, Lander E, and Golub T. 1996. Interpreting patterns of gene expression with selforganizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, 2907-2912.

Tibshirani R, Hastie T, Eisen M, Ross D, Botstein D, and Brown P. 1999. Clustering methods for the analysis of DNA microarray data. *Technical report*, Department of Statistics, Stanford University.

Wen X, Fuhrman S, Michaels G, Carr D, Smith S, Barker J, and Somogyi R. 1998. Large-scale temporal gene expression mapping of central nervous system development, *Proc. Natl. Acad. Sci. USA*, vol. 95, pp. 334-339.

Wu H, and Yan H. 2003. Microarray Image Processing Based on Clustering and Morphological Analysis. *Proc. of First Asia Pacific Bioinformatics Conference*, Adelaide, Australia, 111-118.

Yang M, Ruan Q, Yang J, Eckenrode S, Wu S, McIndoe R, and She J. 2001. A statistical procedure for flagging weak spots greatly improves normalization and ratio estimates in microarray

experiments. *Physiol Genomics*, 7(1): 45-53.

Yang Y, Buckley M, Dudoit S, and Speed T. 2002. Comparison of Methods for Image Analysis on cDNA Microarray Data. *Journal of Computational and Graphical Statistics*, 11:108-136.

Zien A, Ratsch G, Mika S, Scholkopf B, Lengauer T, and Muller KR. 2000. Engineering support vector machine kernels that recognize translation initiation sites, *Bioinformatics*, 16(9):799-807.