Machine Vision and Applications (2004) 15: 121–138
Digital Object Identifier (DOI) 10.1007/s00138-003-0134-1

Machine Vision and Applications

Automated analysis of DNA hybridization images for high-throughput genomics

Suchendra M. Bhandarkar, Tongzhang Jiang, Kunal Verma, Nan Li

Department of Computer Science, 415 Boyd Graduate Studies Research Center, The University of Georgia, Athens, GA 30602–7404, USA

Received: 25 June 2002 / Accepted: 11 November 2003 /
Published online: 17 February 2004 –

Abstract. The design and implementation of a computer vision system called DNAScan for the automated analysis of DNA hybridization images is presented. The hybridization of a DNA clone with a radioactively tagged probe manifests itself as a spot on the hybridization membrane. The imaging of the hybridization membranes and the automated analysis of the resulting images are imperative for high-throughput genomics experiments. A recursive segmentation procedure is designed and implemented to extract spotlike features in the hybridization images in the presence of a highly inhomogeneous background. Positive hybridization signals (hits) are extracted from the spotlike features using grouping and decomposition algorithms based on computational geometry. A mathematical model for the positive hybridization patterns and a Bayesian pattern classifier based on shape-based moments are proposed and implemented to distinguish between the clone-probe hybridization signals. Experimental results on real hybridization membrane images are presented.

Key words: DNA hybridization – Physical mapping – Image analysis – High-throughput genomics – Bayesian classification

1. Introduction

With the increasing use of robotics for rapid generation of experimental data in modern genomics experiments, it has become imperative to be able to analyze the resulting data in an expeditious and reliable manner. The relative paucity of automated techniques for reliable and rapid analysis of genomic data has proved to be the rate-limiting step or bottleneck in most high-throughput genomics experiments [13]. Most of the experimental data in genomics experiments can be acquired and represented in the form of images. Thus, automated analysis of such data entails the design and implementation of appropriate image analysis and machine vision algorithms. To this end, we present the design and implementation of a computer vision system called DNAScan for the automated analysis of DNA hybridization images.

Correspondence to: S.M. Bhandarkar (e-mail: suchi@cs.uga.edu)

There are two classes of genomics experiments that call for the acquisition and subsequent analysis of image data: gene expression analysis experiments and hybridization experiments for physical mapping of chromosomes. In recent times, both classes of experiments have been extensively automated with the use of fast, high-precision robotic equipment. This has resulted in the generation of experimental data at an extremely rapid rate.

1.1. Gene expression analysis

Gene expression analysis experiments are typically performed using DNA microarray technology, and the goal is to quantify the extent to which different genes are activated or expressed in response to environmental stimuli. Genes that are expressed result in the production of messenger RNA (mRNA) in the nucleus of a cell by a process called transcription. The mRNA is subsequently transcribed into proteins in the cell. The extent to which a gene is expressed (i.e., the gene transcription level) can be quantified by the amount of corresponding mRNA produced in the cell. However, since mRNA is chemically unstable, it is converted to complementary DNA (cDNA) by a reverse transcription process. The cDNA is called thus on account of the fact that its base sequence is complementary to that of the corresponding gene.

In gene expression analysis experiments, a DNA microarray is populated (i.e., spotted) with DNA sequences from different genes or open reading frames (ORFs) of an organism. The cDNA samples are extracted from the organism after subjecting it to different stimuli. Alternatively, the organism is subject to a single stimulus and the cDNA is sampled at different time points. Distinct cDNA samples are labeled with fluors of a specific color or wavelength and are referred to as probes. Two cDNA probes are tested by having them chemically react, by a process called hybridization, to the ORF spots on a DNA microarray. The microarray is scanned to determine how much of each probe is bound to each ORF spot by measuring the emittance intensity of each fluor when excited with a laser of the same wavelength. The ratio of the emittances of two fluors is a measure of the relative expression or transcription level change for the given gene or ORF. A series of DNA microarray experiments results in a pattern vector for

each ORF. The components of the pattern vector represent the relative transcription level of the ORF in response to different stimuli or in response to a single stimulus at different points in time.

1.2. Hybridization for physical mapping

In a typical hybridization experiment for physical mapping of chromosomes, pieces of a single-stranded chromosome called *clones* are fixed at predetermined positions on a nylon membrane grid. The membrane grid is then exposed to radioactively or fluorescently tagged *probes*. A probe is a short distinguishable DNA fragment whose DNA sequence is known, in contrast to a clone, whose sequence is typically unknown. A probe attaches to a clone if there exists a site on the clone with a DNA sequence complementary to the DNA sequence of the probe. In this situation, the probe is said to have *hybridized* to the clone. The clone-probe hybridization can be determined by the presence of radioactivity at the clone site on the membrane grid. After the membrane grid is washed to remove the excess probe residue, it is exposed to a film. The radioactively tagged probe leaves a signal in the form of a bright spot at the same location on the film as the clone site on the membrane grid. The film is scanned and converted to a digital format for further analysis. In some modern instruments, the membrane grid is digitally imaged directly without recourse to film.

A series of hybridization experiments results in a binary signature for each clone where a 0 or 1 in the i-th location in the signature denotes the absence or presence of hybridization, respectively, to the i-th probe. Clones with similar binary signatures are deemed to be spatially proximate on the chromosome. The similarity of binary signatures is used to order the clones along the length of the chromosome to generate a physical map of the chromosome. The Hamming distance between the binary clonal signatures could be used as a dissimilarity metric [9]. The desired ordering could be deemed to be the one that minimizes the sum of the Hamming distances between the signatures of successive clones in the ordering. Computing such an ordering can be shown to be isomorphic to the Traveling Salesman Problem (TSP), which is a classical NP-hard combinatorial optimization problem [30]. Having ordered the clones along the chromosome, a minimal subset of clones that spans the entire length of the chromosome (i.e., the minimal tiling) is then determined. The base sequence of each clone in the minimal tiling is determined using a sequence assembly procedure [40]. Once the ordering of clones and the base sequence of each clone in the minimal tiling are known, the base sequence of the entire chromosome can be constructed using a clone overlap determination procedure [40].

In order to automatically analyze the hybridization signals on the film, knowledge of the positions of the clones on the membrane grid is critical. The term *hybridization protocol* refers to the spatial arrangement of clones on the nylon membrane grid. In our case, a single film contains 2304 squares arranged in the form of a 48×48 2-D array [22]. Each square contains $4 \times 4 = 16$ cells, where each cell contains eight clones, with each clone spotted in duplicate. A clone-probe hybridization results in two spots on the film corresponding to the positions of the two cells containing that clone. These two spots are referred to as a *positive hit*. The clone-probe hybridization, however, is determined not only by the actual positions of the spots on the film but also by the relationship between the two spots, which includes the distance between the two spots and the orientation of the straight line joining the two spots. Each pair of spots is referred to as a *pattern class* and is characterized by the position, distance, and orientation measurements mentioned above. Determining the spatial locations of the clones in the 4×4 array such that each pattern class can be distinguished from the other pattern classes is the key issue in hybridization protocol design. Depending on the number of probes that a given membrane grid is exposed to, the number of hybridizations in a 4×4 square varies from 0 to n. The case where $n = 1$ is called a single positive hit, and one where $n \geq 2$ is called a multiple hit. In most hybridization experiments, $n = 0$ or $n = 1$ for most squares. We have developed algorithms to classify a single positive hit to a pattern class given a hybridization protocol. For the case where $n \geq 2$, we have developed algorithms to decompose the resulting signal into several single positive hits for further classification.

2. Review of previous work

There have been attempts to automate the analysis of image data arising from high-throughput genomic experiments. Analysis of gene expression data derived from DNA microarrays has received much attention in recent times. Schena et al. [56] and Lashkari et al. [41] describe DNA microarray experiments for human T cells and yeast ORFs respectively. A number of researchers have investigated algorithms for clustering of pattern vectors for ORFs resulting from a series of microarray experiments. Eisen et al. [24] describe a hierarchical clustering algorithm that uses a greedy heuristic based on the average linkage method [35]. A correlation coefficient is used as a similarity metric between the pattern vectors describing the expression levels of the yeast genome ORFs sampled over time. Michaels et al. [45] compare clustering algorithms using the Euclidean distance and mutual information as similarity metrics. Leach and Hunter [42] present a comparative study of clustering algorithms and clustering metrics for gene expression profiles from microarray data. A comparison of clustering techniques such as k-means, Bayesian mixture of models, and hierarchical clustering and of clustering metrics such as the Euclidean distance, correlation, and mutual information are presented. Ben-Dor and Yakhini [6] describe a clustering algorithm for gene expression patterns based on a stochastic model of the input data. A polynomial-time algorithm is described that recovers the true clusters with high probability. The clustering algorithm is also evaluated using the same input data model. A graph-theoretic cluster analysis algorithm for gene expression data is described in [33]. A similarity graph is defined and clusters in the graph correspond to highly connected subgraphs. A polynomial-time clustering algorithm is presented. In later work [57], the performance of the clustering algorithm is improved by use of connectivity kernels and the use of heuristics for kernel enlargement and merging of clusters.

Some recent research has focused on formal statistical analyses of clustering algorithms for gene expression profiles from microarray data. A statistical model based on a mixture

of Gaussian distributions in the context of hierarchical clustering of gene expression data from DNA microarray experiments has been proposed by [31]. The model is fit to the data using the expectation-maximization (EM) algorithm [23]. The initial parameter values for the EM algorithm are computed using hierarchical agglomerative clustering. The number of clusters is determined using a statistical technique based on Bayes factors [39]. A model based on a mixture of t distributions has been proposed by [44]. A subset of relevant genes for the clustering of tissue samples is selected by fitting the model and ranking the genes in increasing size of the likelihood ratio statistic for the test of one vs. two components in the mixture model. A mixture of factor analyzers is used to reduce the dimensionality of the feature space of gene expression data for more effective clustering. Pan et al. [46,47] compare three statistical methods for discovering differentially expressed genes in replicated microarray experiments, the t-test, a regression modeling approach, and a mixture of models approach. The three methods are shown to differ in how they associate a significance level with the corresponding statistic, leading to a possibly large difference in the resulting significance levels and number of genes detected. A program called UCSF Spot for fully automatic quantification of DNA microarrays is described in [37]. The program automatically locates both subarray grids and individual spots while requiring no user identification of any of the image coordinates. Transcription ratios for the ORFs are computed based on explicit segmentation of each spot. Manduchi et al. [43] describe a protocol by which discrete values are used to provide an easily interpretable description of differential expression. Novel statistical methods are proposed to attach confidence levels to the hypothesis that changes in expression levels represent true changes. Brown et al. [14] present a statistical analysis of DNA microarrays where the normalized standard deviation of ratio measurement, termed the spot ratio variability (SRV), is used to segment spots from an inhomogeneous background. The SRV is then used to assign significance estimates to gene expression ratios.

The gene expression data derived from microarrays can be used to deduce gene regulatory networks [21]. In [21,26], techniques for deriving a Boolean (and hence discrete) gene regulatory network from microarray time series data are presented. A continuous differential equation model for gene regulatory networks is proposed in [19]. Friedman et al. [28] have proposed a Bayesian network for describing the interactions between genes. An efficient algorithm for learning these networks and a statistical method for confidence assessment are provided. A methodology for deriving cell signaling networks from gene expression data has been proposed in [8]. Jansen et al. [38] have investigated techniques for relating gene expression data to protein-protein interactions. There have been several software packages developed by academia and research institutions [5,15,16,18,62] as well as by commercial vendors [1,2,10,55,58,59] for analyzing microarray data.

There have also been several attempts to automate the process of analysis and interpretation of DNA hybridization images [4,11,12,20,49,52,61]. However, most of the techniques therein have employed ad hoc heuristics that work well for high-resolution, noise-free images but not for images that have limited resolution and are corrupted by several hybridization artifacts. For example, the recursive segmentation algorithm in [4] extracts spotlike features from a highly inhomogeneous background. For high-quality, high-resolution films with sparse spotlike features, the result of the recursive segmentation algorithm can be used directly to score for positive hits. However, in the case of low-resolution films with dense spotlike features, merging of spotlike features can pose a barrier to proper pattern classification. The segmentation and pattern classification algorithms developed and presented in this paper are robust to the merging of spotlike features and presence of noisy artifacts.

Chen et al. [20] and Brandle et al. [11,12] describe a system for reliably fitting parametric and semiparametric models to spots in high-density arrays. Their system performs background estimation using a pyramid-based algorithm followed by model fitting to the spot intensity array. The parametric model is a Gaussian spot model whose mean and covariance matrix are estimated using maximum likelihood (ML) estimators and minimization of sum of squared errors. The parametric model is susceptible to outliers and is hence modified to one that is semiparametric. The semiparametric model fitting is done using the relative squared error as a test statistic. Overlapping spots are handled by subtracting neighboring models. Grid fitting is done using a combination of spot magnification, rotation estimation, initial grid placement via optimal search, and grid placement refinement via the Radon transform. Steinfath et al. [61] also describe a similar image analysis system for hybridization experiments. Their system consists of two stages. The first stage consists of signal amplification, noise removal, alignment of the spot array via corner detection, and localization of the spot centers within the array. The second stage determines the intensity of the detected spots by fitting a 2-D Gaussian distribution. The background intensity is assumed to be constant. The mean and covariance matrix of the Gaussian distribution are estimated using maximum likelihood and statistical regression methods.

One of the major drawbacks of the aforementioned systems is that they are focused primarily on spot detection. DNAScan, on the other hand, is capable of not only spot detection but also spot grouping and spot pattern classification. Spot grouping and spot pattern classification are of particular importance in hybridization experiments since the spatial distribution of the clones on the membrane grid (as dictated by the hybridization protocol) carries valuable information. The spot pattern classification is performed using shape-based moments, but our technique differs from traditional moment-based classifiers. Moments that are invariant to rotation and scale were first proposed in [34] and subsequently used in several applications such as aircraft identification [53]. However, rotation-invariant moments are not useful in our case since the classification of positive hits partially depends on the mutual orientation of the two spots comprising the positive hit. Another important consideration in the case of low-resolution and noisy hybridization images is the fact that high-order moments are typically sensitive to noise. Piper et al. [49] proposed a classification scheme based on contour features and the axes of orientation of hybridization signals in high-resolution images, but their technique is not robust when the resolution of the input images is limited. The current version of DNAScan is designed to deal with gray-level intensity images where the probes are radioactively tagged. However, it could be easily extended to include classification techniques based on the color

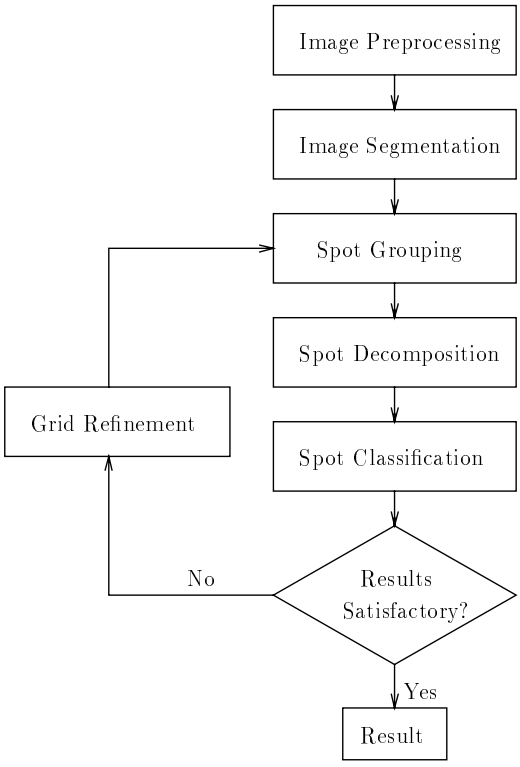


Fig. 1. Flowchart of the DNAScan software

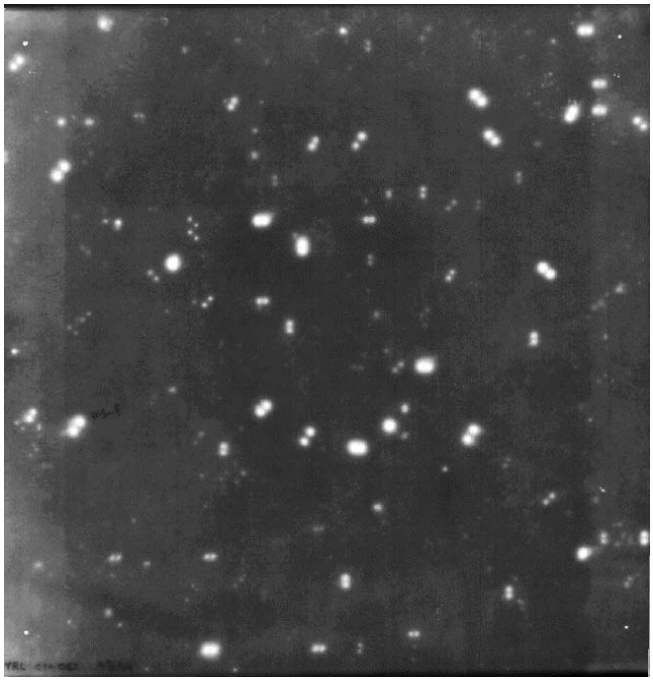


Fig. 2. Typical hybridization image

of the hybridization signals where the color images are generated using fluorescent probes [60]. Figure 1 gives the overall flowchart of the DNAScan software. The output of DNAScan has been used in several genome mapping experiments in the Department of Genetics and the Applied Genetics Technology (AGTech) Center at the University of Georgia [3, 7]. In particular, it was shown that using the spot gray levels instead of binary values in the clone signature improved the performance of physical mapping algorithms [32].

3. Image preprocessing

Figure 2 shows a typical hybridization image. These images were captured on film and transformed into a digital format using a scanner. The typical size of these images is 950×950 pixels, where each pixel encodes an 8-bit grayscale value. The image in Fig. 2 clearly exhibits a highly inhomogeneous background and considerable variation in spot size. The inhomogeneous background is caused by the radioactive residue carried by labeled probes. The variation in spot size is due to variations in exposure times, spotting concentrations, temperature, and hybridization reaction strengths. In addition, the hybridization image is corrupted by noisy artifacts. Some of these artifacts are generated by static and manifest themselves as streaks across the film. Other artifacts are caused by dust or by the film development procedure and typically manifest themselves as tiny specks on the film.

Image preprocessing includes histogram equalization, rotation of the membrane grid to align it with the image coordinate axes, and background subtraction. The variations in film exposure times, film developing procedures, and the operating parameters of the scanner make histogram equalization necessary. However, the relative intensity between pixels is preserved during the histogram equalization procedure to prevent any change in the shape of the hybridization spots [36].

The images are rotated in order to align the borders of the membrane grid with the image coordinate axes. There are two possible approaches to determining the rotation angle. One approach determines the rotation angle by using the coordinates of reference points that are manually marked on the film. Typically, the reference points are at the corners of the film. The primary advantage of this approach is its high accuracy. Its major drawback is that it needs human intervention since the user needs to input the coordinates of the reference points.

The rotation angle can also be determined in a fully automatic manner, i.e., without any human intervention. The automatic approach is less accurate than the manual approach but is better suited for situations where a higher degree of automation is called for. Automatic determination of the rotation angle is done by performing the p-tile thresholding and projection operations [36]. The p-tile thresholding technique uses the area or size of the object within the image to separate the object from the background. In our case, a typical film contains the membrane grid surrounded by a border of low image intensity. The percentage of the membrane grid area over the overall image area is a constant for a given hybridization protocol and fixed scanner resolution and can be used to set the value of p in the p-tile thresholding operation. The p-tile thresholding operation computes the histogram of the image and sets the threshold such that $p\%$ of the pixels in the image belong to the membrane grid and the rest to the border. The pixels belonging to the membrane grid are set to 1, and those belonging to the border are set to 0. Projecting the thresholded image on the x and y axes gives two arrays that contain the column and row sums, respectively. The optimal rotation angle is deemed to be the one that maximizes the number of 0 entries in both arrays. Since in practice the misalignment is small, the

optimal rotation angle is determined via an exhaustive search in a small angular range.

Background subtraction is performed to reduce the inhomogeneities in the image background resulting from the radioactive residue carried by labeled probes. Since the background inhomogeneities are sufficiently complex, they cannot be modeled as a simple bilinear (i.e., planar) variation across the entire image with sufficient accuracy using standard linear regression techniques [51]. Modeling the background variation across the entire image as a single bivariate function of higher order (such as biquadratic or bicubic) would make the procedure for estimating the model coefficients computationally intensive. As a compromise between accuracy and computational efficiency, the background variation is modeled as a *piecewise* bilinear function in the image plane coordinates x and y, i.e., $B(x, y) = ax + by + c$, where (a, b, c) are the coefficients of the bilinear function. The piecewise bilinear function fitting is performed in a hierarchical manner using a quadtree data structure [54]. The root node in the quadtree corresponds to the entire image. If the bilinear fit to the subimage corresponding to the current node results in a fitting error that is greater than a certain threshold, then the node is expanded into four child nodes by splitting the subimage into four equal quadrants. The bilinear fit is performed recursively in the subimages corresponding to each of the four child nodes. The recursive procedure starts with the root node of the quadtree and halts when the fitting error for a node is below a certain threshold or when the subimage is a single pixel. After the coefficients of a bilinear fit for a subimage are computed, the value of the bilinear function is subtracted from the value of each subimage pixel. Once the background subtraction is performed at each pixel in the original image and the resulting image renormalized to have grayscale values in the range [0,255], the threshold for the fitting error is a decreasing function of the depth of the node in the quadtree. An outline of the background subtraction algorithm is given in Fig. 3.

4. Segmentation and feature extraction

The extraction of spotlike features involves three major steps: (1) segmentation of the hybridization signals from a nonuniform background, (2) grouping of spotlike signals that constitute a single positive hit, and (3) decomposition of spotlike signals that are a result of multiple probe hits in a single cell.

4.1. Recursive segmentation

A recursive segmentation technique similar to the one proposed in [4] is used to extract spotlike features from a nonuniform background. Although the background subtraction procedure described earlier does remove most of the background inhomogeneities, it does not result in an absolutely uniform background. A nonuniform background renders a global thresholding technique unsuitable. Instead, a recursive algorithm is used to find a local threshold value. Initially, a very small threshold value i_t, typically close to 0, is applied to the entire image. The output is a set of regions of arbitrary shape. A bounding box is associated with each of these regions, and a corresponding subimage is extracted. A new threshold value $i_t^{'}$ is computed as follows and used to further split the subimage into several new regions:

$$i_t^{'} = \max\{i_t + 1, i_t + \alpha(i_{\max} - i_t)\}. \quad (1)$$

Here $i_{\max}$ is the maximum intensity of the subimage and α is a constant coefficient where $0 < \alpha < 1$. In our case, $\alpha = 0.1$.

Figure 4 illustrates the recursive segmentation procedure with a 1-D intensity profile. The image is first segmented using an initial threshold $i_0 = 0$. As a result, a region R_1 is detected. A new threshold value i_1 is computed based on the values of i_0 and the local intensity maximum in region R_1. Applying threshold i_1 to R_1 results in a smaller region R_2. Next, a threshold value i_2 is computed based on i_1 and the local intensity maximum of R_2. Applying threshold i_2 to R_2 splits R_2 into two regions, R_3 and $R_3^{'}$. Two new threshold values i_3 and $i_3^{'}$ are computed based on i_2 and the local intensity maxima of R_3 and $R_3^{'}$, respectively. Threshold values i_3 and $i_3^{'}$ are applied to regions R_3 and $R_3^{'}$, respectively. This process is continued until the termination criterion for the recursion is reached.

The termination criterion for the recursion is that the region under consideration is identified as a *spotlike feature* or $i_t^{'} \geq i_{\max}$. The condition $i_t^{'} \geq i_{\max}$ is self-explanatory; the key issue, therefore, is the appropriate definition of a *spotlike feature*. Since the size and intensity of the spots are variable, they cannot be used as the sole criteria to identify spotlike features. In our case, a spotlike feature is deemed to fall into one of the following categories:

Category 1: A small spot that is one of the two spots comprising a positive hit. Typically, these small spots are weak in terms of intensity value, or

Category 2: A large, high-intensity region resulting from the merging of two or more spots.

Given the above two categories and the assumption that a spot is homogeneous in terms of gray level, the following criteria are used to classify a region as a spotlike feature:

(a) A region is homogeneous with regard to intensity in that the variance of pixel intensities within the region is less than a certain threshold and one of the following conditions holds:

(b.1) The region area is under the maximum spot size. The maximum spot size is determined by the size of the cell (in pixels) on the nylon membrane. In our case the size of a typical grid is 850×850 pixels and consists of 48×48 squares, where each square has 4×4 cells. Thus, the size of each cell is ≈ 20 pixels. This criterion is designed for Category 1 spots.

(b.2) For at least two successive recursive calls of the procedure, a parent region always creates a single child region. This criterion is designed to retain the shape of Category 2 spots for further decomposition.

The spots in the two categories are differentiated using an area threshold determined by the membrane cell size. At the end of the recursive segmentation process, there typically exist in the image, spots arising from noisy artifacts. These are removed using area-based and elongation-based filters. Spots whose area is below a threshold (one fourth the size of a single cell $\approx$ 5 pixels) are considered to be noisy specks arising from dust or imperfections in the film development procedure. True spots are expected to be roughly circular. The elongation E of a spot is defined as $E = \frac{I_{\max}}{I_{min}}$, where $I_{\max}$ and I_{min} are, respectively, the maximum and minimum moments

```
Node_queue = root node of the quadtree (representing the entire image);
While not empty (Node_queue)
beginwhile
    Current_node = dequeue(Node_queue);
    Perform bilinear function fitting in the subimage that the Current_node represents and compute the bilinear function coefficients (a,b,c);
    Compute fitting_error;
    If (fitting_error ≤ threshold), then
    beginif
        Perform background subtraction on the subimage that Current_node represents;
        Renormalize the image after background subtraction;
        endif
    Else
    beginelse
        Expand the Current_node into its 4 child nodes;
        If the child nodes are not a single pixel, enqueue them in the Node_queue;
    endelse
endwhile
```

Fig. 3. Outline of the hierarchical background subtraction algorithm

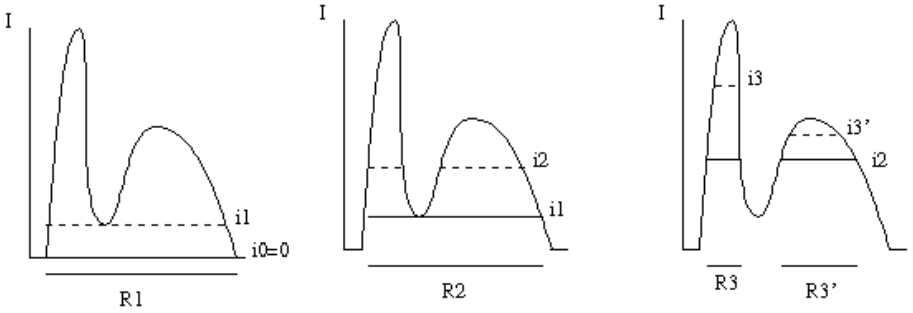


Fig. 4. Recursive segmentation

of inertia. Note that circular spots have $E \approx 1$. Spots with elongation higher than a specified threshold are assumed to be noisy streaks arising from static. Figure 5 shows the result of the segmentation process.

4.2. Grouping

The recursive segmentation procedure also results in the extraction of undesired artifacts such as stray marks on the film caused by probe residue and digitization noise. Although area- and elongation-based filters can remove some of these artifacts, we still need to exploit the properties of positive hits themselves in order to remove the artifacts more effectively. The properties of positive hits include the relationship between the spots and the shape of spots involved in the positive hits. To this end, a grouping algorithm was designed and implemented to pair the Category 1 spots described above. At the end of the grouping procedure we expect to retain the true Category 1 and Category 2 spots.

A Category 1 spot and its partner are assumed to be similar in terms of their key features such as area, intensity, perimeter, and elongation. The distance (or difference) between the two spots in image space and in terms of the above key properties is used to determine their (dis)similarity. The dissimilarity index $DS(S_1, S_2)$ for a pair of Category 1 spots (S_1, S_2) is defined as:

$$DS(S_1, S_2) = D_a(S_1, S_2) + D_p(S_1, S_2) + D_i(S_1, S_2) + D_e(S_1, S_2), \quad (2)$$

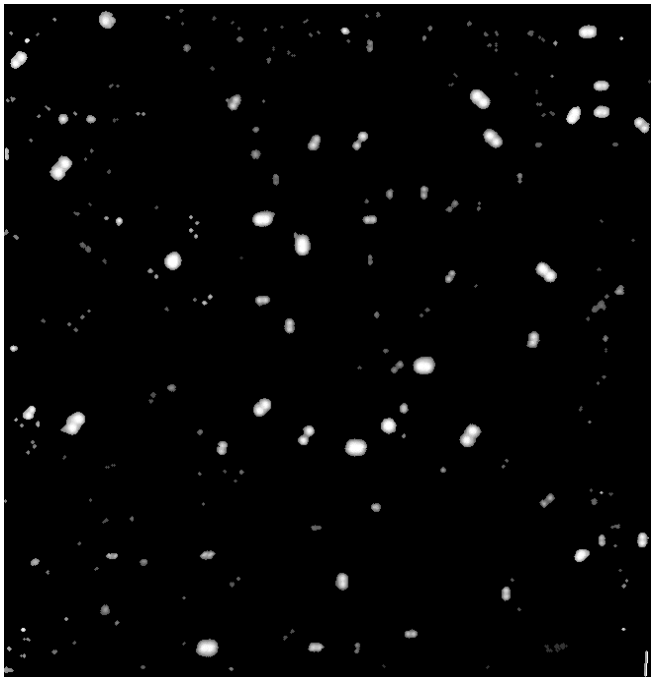


Fig. 5. Result of image segmentation

where $D_a(S_1, S_2)$, $D_p(S_1, S_2)$,$D_i(S_1, S_2)$ and $D_e(S_1, S_2)$ are the distance measures along the spot features such as area, perimeter, average spot intensity, and elongation, respectively.

By definition,

$$D_a(S_1, S_2) \stackrel{\text{def}}{=} \frac{|area(S_1) - area(S_2)|}{\min(area(S_1), area(S_2))}. \quad (3)$$

$D_p(S_1, S_2)$, $D_i(S_1, S_2)$, and $D_e(S_1, S_2)$ are similarly defined except that different features are used. We define $D(S_1, S_2)$ as the Euclidean distance between the centroids of spots S_1 and S_2.

The goal of the grouping algorithm is to find for every spot S_1 in Category 1, a qualified Category 1 spot S_2 such that $DS(S_1, S_2)$ is minimized over all qualified pairs of Category 1 spots containing S_1. If no such Category 1 spot S_2 exists, then S_1 is considered a noisy artifact and removed from further consideration. A pair of Category 1 spots (S_1, S_2) is a qualified pair if each of the distances $D(S_1, S_2)$, $D_a(S_1, S_2)$, $D_p(S_1, S_2)$, $D_i(S_1, S_2)$, and $D_e(S_1, S_2)$ defined above is below its corresponding threshold value. A simple grouping algorithm for pairing similar spots is based on a greedy heuristic. The basic idea is to choose a pair of Category 1 spots with the smallest dissimilarity index from all the qualified pairs at each step and repeat this process until no more qualified pairs remain. At the end of the grouping process, the remaining unpaired spots are either considered to be Category 2 spots if their area exceeds a certain threshold or false isolated spots otherwise.

The greedy grouping algorithm described above gives correct results in about 80% of cases. In particular, it fails to handle the case where one spot in a pair is spatially proximate and similar to a spot in another pair. The greedy grouping algorithm will first group these spots because of their small dissimilarity index, and the remaining two spots in each of the pairs will most likely be discarded. This is a serious limitation of the greedy grouping algorithm.

An improved algorithm considers grouping from a global perspective. It first partitions the entire image into disjoint spot clusters such that each spot in a cluster is distant enough (in terms of the dissimilarity index) from every spot in any other cluster. Thus only two spots from the same cluster could possibly be grouped. Each cluster is treated as a fully connected graph where the spots are the vertices. Edges between vertices are weighted such that edges between spots with a higher dissimilarity index have a lower weight. Edges between spots with a dissimilarity index larger than a specified threshold are ignored. A vertex matching the largest total edge weight is determined within each cluster using the algorithm proposed in [29]. In our case, since each graph within a cluster typically has fewer than six vertices and ten edges, the maximum weight vertex matching algorithm is not much slower than the greedy algorithm. An outline of the improved grouping algorithm is given in Fig. 6.

4.3. Decomposition

A decomposition algorithm was developed to decompose large spots resulting from the merging of more than two spots. Typically, the large spots are the result of multiple probe hits within a 4×4 square on the membrane. This situation is commonly encountered when a clone grid is hybridized to several probes simultaneously – an experimental protocol referred to as probe multiplexing [17,25]. There are two approaches to decompose a multiple-hit spot into several single-hit spots. We present an algorithm based on nonconvex polygon decomposition followed by an algorithm based on the Hough transform.

4.3.1. Nonconvex polygon decomposition

The basic idea behind this approach is to decompose a nonconvex polygon into the smallest number of convex polygons, a classical problem in computational geometry [50]. A single-hit spot is typically convex, whereas a spot arising from multiple hits is typically nonconvex. If the spot is approximated by a polygon, then the key issue in decomposing a nonconvex polygon is, first, determining the break points on the contour of the polygon, and second, determining the proper sequence of break points (if there are more than two break points) for decomposition.

All nonconvex corners of a nonconvex polygon are candidate break points. A nonconvex corner A is determined by computing the fraction P of the area of triangle ABC that is covered by the polygon, where pixels B and C are neighbors of pixel A on the polygon contour, as follows:

$$P = \frac{\text{Area}(ABC \cap \text{polygon})}{\text{Area}(ABC)}. \quad (4)$$

If P is below a certain threshold, we consider A to be a nonconvex corner. To reduce the noise introduced by contour digitization, pixels B and C are not chosen to be the immediate neighbors of A but typically those that are four or five pixels away from A on the spot contour. Figure 7a shows a spot resulting from the merging of several hits (i.e., merged spot), and Fig. 7b shows the candidate break points.

Since the corner detection is inherently noise prone, we use a threshold to retain only the significant nonconvex corner pixels. The thresholding criterion is one based on k-curvature [27], i.e., only those nonconvex corner pixels with k-curvature values over a certain threshold are considered. To further reduce the effect of noise, we use aggregation to reduce a group of potential corner pixels to one pixel, where a group is defined as the set of potential corner pixels such that the distance along the polygon contour between any pair of pixels is below a certain threshold. For each group, we determine the pixel with the highest k-curvature value and make it the representative pixel for that group. Figure 7c shows the break points after the aggregation procedure.

Since the above procedure is expected to result in more than two break points in most situations, there are several ways in which to decompose a merged spot. Hence determining the right sequence of break points to decompose a nonconvex polygon is critical. Since a single-hit spot is roughly circular or elliptical, we exploit the fact that a circle has the maximum compactness value among all 2-D shapes. The compactness of a 2-D shape is the ratio of its area to its perimeter. Hence at each stage we select a pair of break points that would result in (sub)spots with the maximum total compactness. The procedure is halted when all the (sub)spots are convex. It should be mentioned that the decomposition of a merged spot introduces new edges and contours. Since these new edges are simply represented by a straight line between the corresponding break points, they are not as smooth as the original edges.

Input: a list containing all spots detected.
Output: a list containing the resulting Category 1 and Category 2 spots. This is initialized to an empty list.
Phase 1: Generate spot clusters. Each spot belongs to a single cluster.
For each spot $s \in$ *Input*
beginfor
  If $s \notin$ any spot cluster, then
  beginif
    Create a new spot cluster C (implemented as a queue).
    Add s to C.
    For every spot $s' \in C$
    beginfor
      Define a window W (currently 10×10 pixels) centered at the centroid of spot s.
      Add every spot $s'' \in W$ to C.
    endfor
  endif
endfor
Phase 2: Generate disjoint spot pairs within each cluster.
For each cluster C
beginfor
  Create graph $G(V, E)$, where $V = \{s : s \in C\}$ and $E = \phi$.
  For every spot pair $\{s, s'\} \subseteq V$
  beginfor
    If $DS(s, s') <$ threshold, then $E = E \cup \{(\{s, s'\}, w_{s,s'})\}$. (Here $w = DS_{\max} - DS(s, s')$ is the edge weight, where $DS_{\max}$ is the maximum possible value for the dissimilarity metric DS).
  endfor
  For graph $G(V, E)$, compute the vertex matching $M_G = \{(s, s', DS(s, s'))\}$ with maximum total edge weight.
  Output = *Output* $\cup M_G$ (Add M_G to *Output*).
endfor
Phase 3: Remove small isolated spots while retaining Category 2 spots.
For each spot $s \in$ *Input*
beginfor
  If $((s, s', i)$ or $(s', s, i) \notin$ *Output*) and (Area(s) $\geq$ Threshold), then insert s into *Output*.
endfor

Fig. 6. Outline of the improved grouping algorithm

The k-curvature computation and nonconvex corner detection procedures cannot be expected to give reasonable results on these newly created edges. Hence, during the decomposition procedure, we restrict ourselves to the break points detected on the original boundary. Figure 7d shows the final result of the decomposition process. Figure 8 gives an outline of the nonconvex polygon decomposition algorithm.

4.3.2. The Hough transform approach

Unlike the nonconvex polygon decomposition approach, which exploits only the nonconvex corner pixels, the Hough transform exploits all the pixels along the shape contour. Since multiple-hit spots can be modeled by the merging of several circles, the Hough transform for circle detection is used to extract the circles underlying the single-hit spots. A circle of the form $(x - a)^2 + (y - b)^2 = r^2$ is represented by the 3-D Hough accumulator denoted by (a, b, r). At the end of the voting procedure, the cells in the (a, b, r) accumulator with a value greater than a certain threshold represent circles in image space. The threshold value used is r-dependent and in our case is $2\pi r \times c$, where c is a constant such that $0 < c < 1$. The value of c used was in the range $[0.4, 0.5]$. Aggregation in the (a, b, r) accumulator is needed to account for noisy edge pixels. Cells $\{P_1, P_2, \ldots, P_n\}$ in the Hough accumulator are considered to constitute a group if the distance in (a, b, r) space between any two cells within the group is below a certain threshold and the distance between a cell in one group and any other cell in another group is over the threshold. A vote-weighted average of the (a, b, r) parameters of the member cells is used to represent the group. Once the parameters of the circles are determined, the merged spot is decomposed. Figures 9a–c depict the stages in spot decomposition using the Hough transform.

5. Pattern classification

In the pattern classification stage the spots are classified as having arisen from hits to specific clones as specified in the hybridization protocol. The pattern classification procedure consists of two steps: (a) identifying the features that would enable distinction between the classes described in the hybridization protocol and (b) designing a classifier that would exploit these features to classify the positive hits to the appropriate clone class specified in the hybridization protocol.

5.1. Feature selection

Feature selection is critical for the subsequent design of the classifier. The basic rule in feature selection is to choose fea-

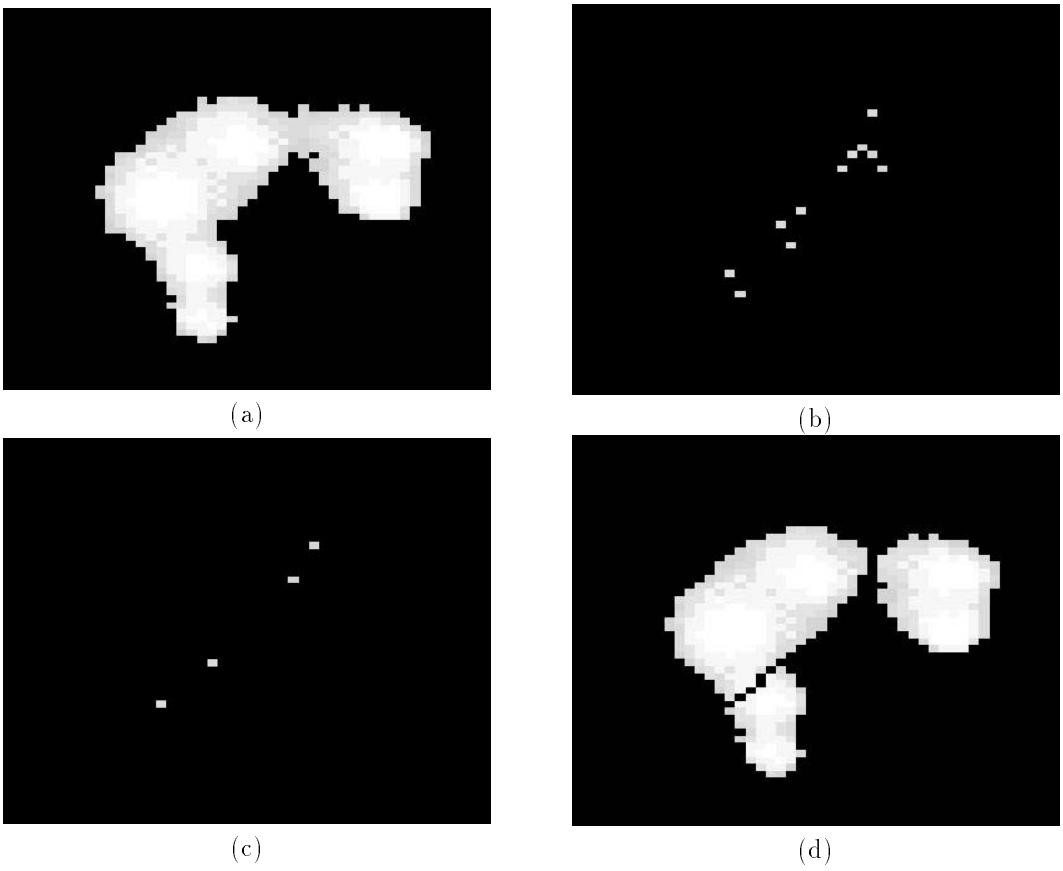


Fig. 7. a Merged spot. **b** Candidate breakpoints before aggregation. **c** Selected breakpoints after aggregation. **d** Decomposed spot

tures that would result in large *interclass distance* and small *intraclass variance* in the feature vector space. Although there is little in the way of a general theory of feature selection for any general problem, it is possible to state some desirable feature properties in our case: (a) the dimensionality of the feature vector is as low as possible while as much discriminatory information is retained as possible, (b) the feature vector is invariant with respect to scaling, (c) the feature vector is sensitive to the shape and orientation of the object, and (d) the feature vector can be computed given an experimental protocol, instead of having to be learned using training examples. The last requirement is of special importance since the goal is to arrive at a mathematical model that would allow one to compute feature vectors for any given experimental clone-probe hybridization protocol.

5.1.1. Mathematical model

For a 2-D shape S in a binary image, the absolute central moments u_{pq} are defined as

$$u_{pq} = \sum_{S} (\mid u - \bar{u} \mid)^p (\mid v - \bar{v} \mid)^q I(u,v), \qquad (5)$$

where $\bar{u}$ and $\bar{v}$ are the coordinates of the centroid of the 2-D shape and $I(u,v)$ is the intensity at pixel (u,v). For background pixels $I(u,v) = 0$, whereas for object pixels $I(u,v) = 1$. Based on the hybridization protocol, each positive hit can be approximated by two circles. We assume that the two circles are similar in size. This assumption holds under most situations. Even though the two circles may, in some cases, merge with each other and form an ellipse, we still use the two-circle model as the basis of our work. Figure 10 shows two circles S_1 and S_2 with equal radii representing the two spots involved in a positive hit, where r is the circle radius and point (α, β) is the midpoint of the line segment joining the two circle centers.

The absolute central moments for the continuous case can be computed as:

$$u_{pq}(S_1, S_2) = \int_{-r}^{r} \int_{-r}^{r} (\mid u - \alpha \mid)^p (\mid v - \beta \mid)^q du dv, \quad (6)$$

where (u,v) is a pixel in either circle S_1 or S_2. Since S_1 and S_2 are symmetric about (α, β), we have

$$u_{pq}(S_1, S_2) = 2 \int_{-r}^{r} \int_{-r}^{r} (\mid u - \alpha \mid)^p (\mid v - \beta \mid)^q du dv, \quad (7)$$

where (u,v) lies in circle S_1. Based on the above equation, we can formulate the moments u_{21}, u_{20}, u_{12}, and u_{02}, which can potentially be used in the feature vector, as follows:

B: binary image of the spots to be decomposed.
S: single spot to be decomposed.
$D(e_1, e_2)$: distance between pixels e_1 and e_2 along the edge contour.
$K(e)$: k-curvature computed at pixel e.
L: list of nonconvex corner pixels initialized to ϕ.
Traverse the contour of S and put the contour pixels in list L.
Delete all pixels in L that do not represent nonconvex corners.
Compute k-curvature for each pixel in L.
Delete all pixels with k-curvature below a certain threshold.
Assign a unique label to each pixel in L.
For each pixel $e_1 \in L$.
beginfor
  For each pixel $e_2 \in L$ such that $e_2 \neq e_1$.
  beginfor
    If $D(e_1, e_2) \leq$ threshold then $Label(e_2) = Label(e_1)$
  endfor
endfor
Sort L in descending value of K.
For each pixel $e_1 \in L$.
beginfor
  For each pixel $e_2 \in L$ such that $e_2 \neq e_1$.
  beginfor
    If $(Label(e_2) = Label(e_1))$ and $(K(e_2) < K(e_1))$, then delete e_2 from L.
  endfor
endfor
While $(L \neq \phi)$
beginwhile
  For each spot S under consideration
  beginfor
    Determine a pair (e_1, e_2) in L such that the straight line between e_1 and e_2 divides S into two parts S_1 and S_2 and maximizes (compactness(S_1) + compactness(S_2)).
    Split S along the straight line between e_1 and e_2 into two parts S_1 and S_2.
    Delete e_1 and e_2 from L.
  endfor
endwhile

Fig. 8. Outline of the nonconvex polygon decomposition algorithm

$$
\begin{aligned}
u_{20}(S_1, S_2) &= 2\int_{-r}^{r}\int_{-r}^{r}(\mid u-\alpha \mid)^2 dudv \\
&= 2\int_{-r}^{r}\int_{-r}^{r}(u-\alpha)^2 dudv \\
&= 2\int_{-r}^{r}(u-\alpha)^2 du\int_{-r}^{r} dv \\
&= 2\times\frac{1}{3}(u-\alpha)^3 \mid_{-r}^{r} \times 2r \\
&= \frac{8r^2}{3}(r^2+2\alpha^2). \qquad (8)
\end{aligned}
$$

Similarly, it can be shown that

$$u_{02}(S_1, S_2) = \frac{8r^2}{3}(r^2+2\beta^2), \qquad (9)$$

$$u_{21}(S_1, S2) = \frac{8}{3}\beta r^2(r^2+2\alpha^2), \qquad (10)$$

$$u_{12}(S_1, S_2) = \frac{8}{3}\alpha r^2(r^2+2\beta^2). \qquad (11)$$

From the above equations we can easily identify two candidate features $\frac{u_{12}}{u_{02}} = \alpha$ and $\frac{u_{21}}{u_{20}} = \beta$. They are independent of spot size r and yet are sensitive to the shape of the spot. At the same time, they are not sufficient to distinguish a spot from another spot with the same shape, but at a different orientation. An orientation feature that equals $\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right)$ is therefore added to the feature vector. Finally, we set the feature vector to be

$$\left[\frac{u_{12}}{u_{02}},\ \frac{u_{21}}{u_{20}},\ \tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right), c_x, c_y\right],$$

where (c_x, c_y) denotes the position of the centroid of the spot pair (Category 1) or a single spot (Category 2) within the 4×4 cell.

However, as r becomes larger the two circles will merge. This will cause the values of u_{pq} to deviate from those derived from the above model for the following reasons:

(a) When two circles merge, $u_{pq} \neq 2\int_{-r}^{r}\int_{-r}^{r}(\mid u-\alpha \mid)^p(\mid v-\beta \mid)^q dudv$, and
(b) $\int_{-r}^{r}(\mid u-\alpha \mid)^p \neq \int_{-r}^{r}(u-\alpha)^p$ when $r^2 > \alpha^2+\beta^2$.

The above analysis implies that the moment values used in the feature vector do not comply with those computed using the above model if r is sufficiently large. However, the model holds in the range of r we are working in, as will be seen in the following case study.

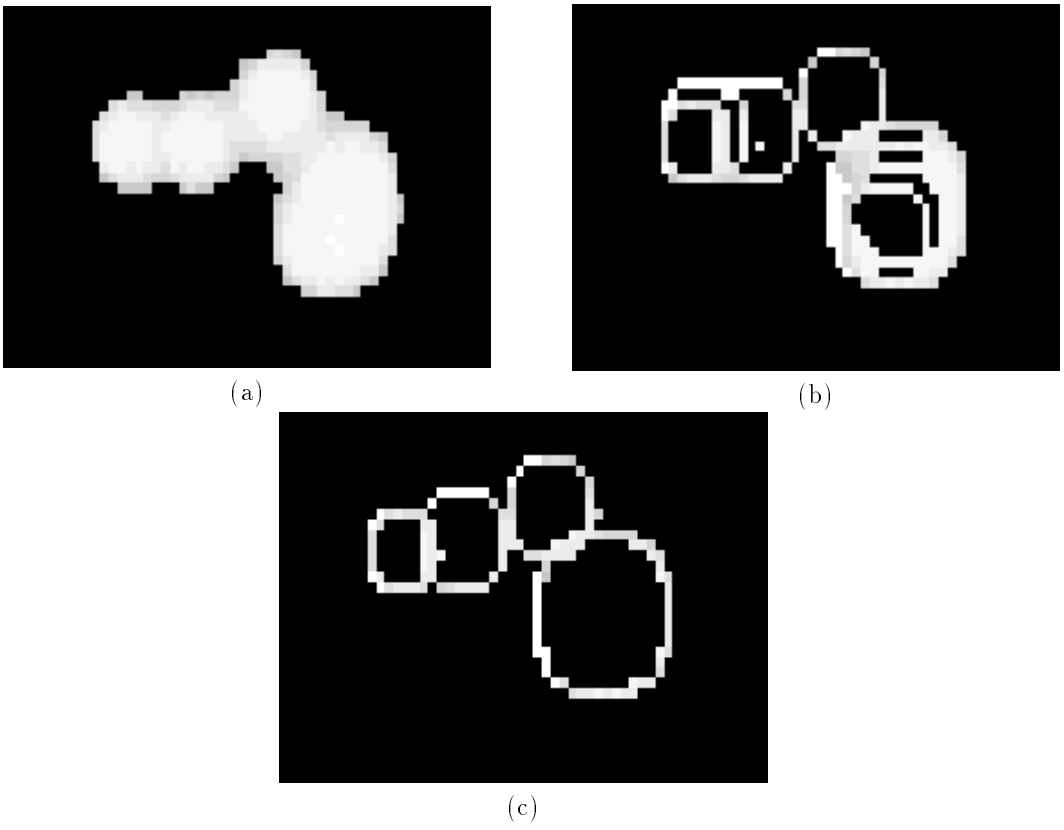


Fig. 9. a Merged spot. **b** Detection of overlapping circles using the Hough transform. **c** Detection of final circles after aggregation in the Hough accumulator

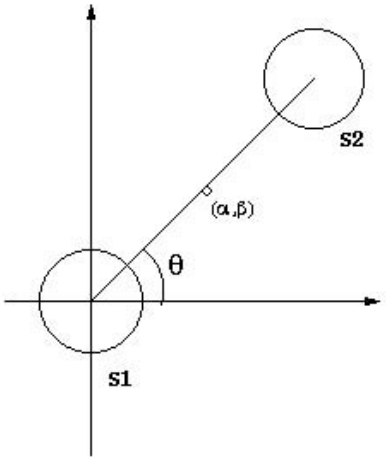


Fig. 10. Geometric model for positive hits

It is worth noting that the moment-based feature vector described above does possess the following desirable characteristics: (a) invariance (to a great extent) to spot size, (b) sensitivity to spot shape, (c) sensitivity to the orientation of the spot or spot pair, and (d) sensitivity to spot (or spot pair) position within the 4×4 cell but invariance to spot (or spot pair) position on the overall membrane grid. Boundary-based shape features such as Fourier descriptors, on the other hand, suffer from the following shortcomings: (a) sensitivity to noise and boundary distortions caused by the merging of spots, and in their standard formulation, (b) sensitivity to the choice of the starting point on the spot boundary, (c) sensitivity to spot size (i.e., scale) and, (d) sensitivity to spot (or spot pair) position on the overall membrane grid (i.e., translation). It was for these reasons that the moment-based features described above were chosen over boundary-based shape features such as Fourier descriptors.

1	4	8	2
3	5	8	3
7	7	5	6
2	4	6	1

Fig. 11. Hybridization protocol

5.1.2. Case study

Figure 11 shows an experimental protocol (4×4 square on the hybridization membrane) designed by the Paterson Laboratory in the Applied Genomics Technology (AGTech) Center at the University of Georgia. Note that the cells with the same label in the 4×4 square contain the same clone. This is a typical protocol and reflects the basic rules of protocol design:

(a) If two patterns share the same orientation, they have different interspot distances (for example, patterns 1 and 5) and
(b) If two patterns share the same interspot distance, their orientations are different (for example, patterns 1 and 2).

Table 1. Comparison of experimental and theoretical values of α, β, and θ

Pattern class	α	$\frac{u_{12}}{u_{02}}$	β	$\frac{u_{21}}{u_{20}}$	θ	$\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right)$
1	4.5	4.5	4.5	4.5	$-\frac{\pi}{4}$	-0.791
2	4.5	4.5	4.5	4.5	$\frac{\pi}{4}$	0.794
3	4.5	4.5	0	1.1	0	0.012
4	0	1.1	4.5	4.5	$\frac{\pi}{2}$	1.622
5	1.5	1.7	1.5	1.7	$-\frac{\pi}{4}$	-0.793
6	1.5	1.7	1.5	1.7	$\frac{\pi}{4}$	0.790
7	1.5	1.7	0	1.1	0	0.013
8	0	1.1	1.5	1.7	$\frac{\pi}{2}$	1.615

We followed the procedure described below to verify the mathematical model. For a particular value of spot area, we simulated the shape of the digitized spots. Since, in practice, each spot is assumed to be a circle, efforts were made to ensure that the digitized spots modeled a circle as closely as possible. The digitized spots were then patched to a binary matrix to simulate positive hits for a specific pattern by maintaining the appropriate orientation and distance between the spot centers. For each value of spot area and for each pattern class, the values of $\frac{u_{12}}{u_{02}}$ and $\frac{u_{21}}{u_{20}}$ were computed from the simulated data.

Table 1 shows the mean values of $\frac{u_{12}}{u_{02}}$, $\frac{u_{21}}{u_{20}}$ and $\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right)$ from the simulated data compared against the theoretical values of α, β, and θ, respectively. The mean values were computed over a range of values for the spot area. We found the values of $\frac{u_{12}}{u_{02}}$, $\frac{u_{21}}{u_{20}}$, and $\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right)$ to be in reasonable compliance with the theoretical values of α, β, and θ within the bounds of discretization error (note that the model was derived for the continuous case).

Figures 12 and 13 show the values of $\frac{u_{12}}{u_{02}}$ and $\frac{u_{21}}{u_{20}}$, respectively, plotted as a function of spot area for each pattern class. It can be seen that the values of $\frac{u_{12}}{u_{02}}$ and $\frac{u_{21}}{u_{20}}$ deviate from the theoretical values of α and β, respectively, as the spot area grows larger and the two spots merge. However, since spot area was observed to be less than 60 pixels in the experiments performed, it is reasonable to use the proposed model for pattern classification while limiting the classification error within acceptable bounds.

5.2. Classifier design

The classifier was designed based on Bayes' decision theory [63]. Recall that the feature vector used in the classifier is $\left[\frac{u_{12}}{u_{02}}, \frac{u_{21}}{u_{20}}, \frac{1}{2}\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right), c_x, c_y\right]$, where $\frac{1}{2}\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right)$ is the orientation of the positive hit, $\left[\frac{u_{12}}{u_{02}}, \frac{u_{21}}{u_{20}}\right]$ represents the shape of the positive hit, and (c_x, c_y) is the position of the centroid within the 4×4 cell. There are two plausible approaches one could take for pattern classification using the above vector:

(a) Design a single classifier for the 5-D feature vector, or

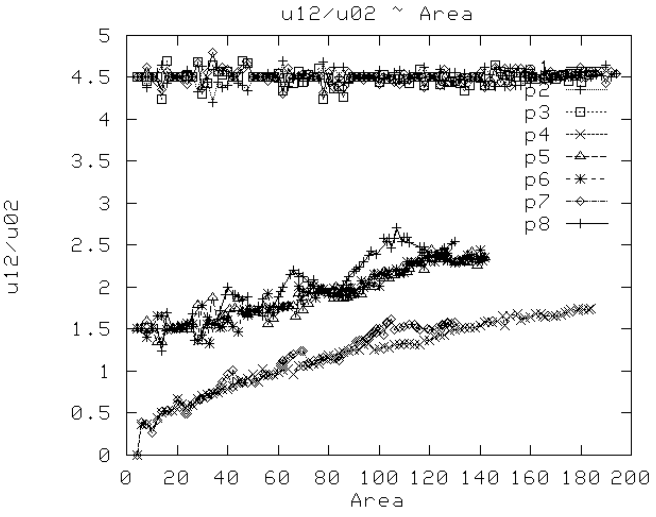


Fig. 12. Plot of $\frac{u_{12}}{u_{02}}$ vs. spot size

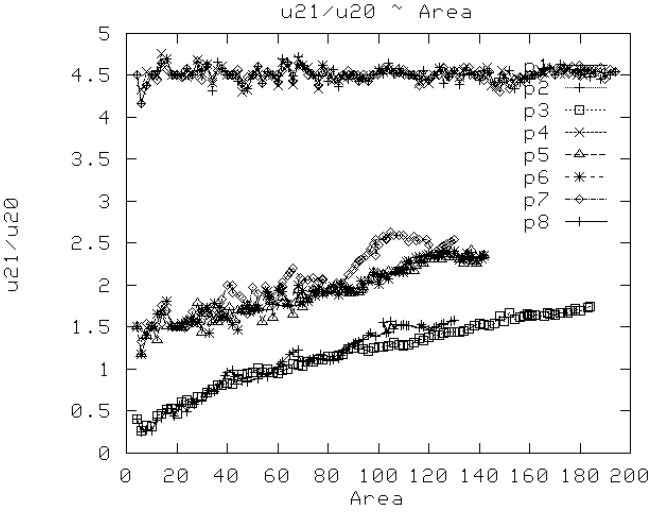


Fig. 13. $\frac{u_{21}}{u_{20}}$ vs. spot area

(b) Decompose the feature vector into two subvectors $\left[\frac{u_{12}}{u_{02}}, \frac{u_{21}}{u_{20}}, c_x, c_y\right]$ and $\left[\frac{1}{2}\tan^{-1}\left(\frac{u_{11}}{u_{20}-u_{02}}\right)\right]$. The orientation feature is used first as a 1-D feature to classify patterns into groups, where a group may contain more than one pattern class. Subvector $\left[\frac{u_{12}}{u_{02}}, \frac{u_{21}}{u_{20}}, c_x, c_y\right]$ is then used as a 4-D feature vector to further classify the patterns in each group.

We prefer the second approach for the following reasons:

(a) In both protocol design and pattern classification by humans, pattern orientation and pattern shape and position are considered separately. For an input pattern, the typical classification procedure employed by a human is to first classify the input pattern to a group based on pattern orientation (a group may contain more than one pattern class). This is followed by further subclassification of the patterns in a group into pattern classes based on pattern shape and position. We have emulated this procedure in our classifier design.

Table 2. Groups of pattern classes based on θ

Group	Pattern class	θ
1	3,7	0
2	2,6	$\frac{\pi}{4}$
3	4,8	$\frac{\pi}{2}$
4	1,5	$-\frac{\pi}{4}$

(b) Feature space decomposition reduces the computational complexity significantly.
(c) Orientation is computed using second-order moments, which makes it more robust to noise than the subvector $\left[\frac{u_{12}}{u_{02}}, \frac{u_{21}}{u_{20}}, c_x, c_y\right]$, which entails the computation of third-order moments.

In statistical terms, we assume that the orientation, shape, and position features are statistically independent.

5.2.1. Orientation-based classifier

A 1-D orientation-based Bayesian classifier [27,63] was designed. The eight patterns described in our protocol (Fig. 11) can be divided into four groups, each containing two pattern classes, based on the pattern orientation θ as shown in Table 2.

The group-conditional probability distribution for the measured orientation θ given by $p(\theta|g_i)$ is assumed to be Gaussian:

$$p(\theta|g_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(\theta-\mu_i)^2}{2\sigma_i^2}}, \tag{12}$$

where μ_i is the mean orientation of each group (given in Table 2) and σ_i^2 is the variance associated with the orientation. The orientation variance for each group was estimated using positive hit samples from each pattern class.

The choice of the Gaussian probability distribution function is motivated by the fact that the computation of the moment-based features involves the weighted sum of pixel coordinate values. If the pixel coordinate values are statistically independent and identically distributed, then the moment-based features can be characterized by a Gaussian distribution in the asymptotic limit based on the Central Limit Theorem [48]. This assumes that each spot has a sufficiently large number of pixels. In practice, if the number of statistically independent and identically distributed random variables is greater than 12, then the probability distribution of their sum can be approximated by a Gaussian probability distribution function with less than 5% error [48]. In our case, since each spot has about 20 pixels, the assumption of a Gaussian probability distribution function for the moment-based features is justified. Another advantage is that the Gaussian distribution is analytically tractable and is completely characterized by only two parameters, the mean and the variance, which can be efficiently estimated from training samples.

Using the Bayesian formalism the a posteriori probability distribution $p(g_i|\theta)$ can be computed as

$$p(g_i|\theta) = \frac{p(\theta|g_i)}{p(\theta)} p(g_i). \tag{13}$$

The goal in Bayesian classification is to assign the pattern to the group with the highest a posteriori probability $p(g_i|\theta)$.

Table 3. Loss matrix for the Bayesian classifier

	C_3	C_7	C_0
C_3	0	1	0.25
C_7	1	0	0.25

Under the assumption that all the groups g_i occur with equal probability and since $p(\theta)$ does not depend on any particular group, maximizing $p(g_i|\theta)$ is tantamount to maximizing the group-conditional probability $p(\theta|g_i)$. Since $p(\theta|g_i)$ is Gaussian, maximizing $p(\theta|g_i)$ is equivalent to minimizing the Mahalanobis distance D_M [63] given by

$$D_M = \frac{(\theta-\mu_i)^2}{2\sigma_i^2}. \tag{14}$$

In short, the input pattern is assigned to the group g_i with the shortest Mahalanobis distance D_M from the input pattern [63].

5.2.2. Shape and position-based classifier

We assume the class-conditional probability density function for the shape and position to be a multivariate Gaussian distribution in $x = \frac{u_{12}}{u_{02}}$, $y = \frac{u_{21}}{u_{20}}$, c_x, and c_y. If the shape and position features are assumed to be statistically independent, then the multivariate Gaussian distribution can be decomposed into a product of two bivariate Gaussian distributions, one in (x, y) and the other in (c_x, c_y). The classification approach is one based on minimization of Bayesian risk [27,63]. For each group, a loss matrix is defined that quantifies the risk of misclassification. For example, Table 3 shows the loss matrix for group 1 consisting of pattern classes 3 and 7. Class C_0 refers to the *unknown* class.

The goal of Bayesian classification is to minimize the overall risk of misclassification for every input pattern $[x,\ y,\ c_x,\ c_y]$ within its corresponding group G determined by the orientation classifier. The overall risk of classifying an input pattern $[x,\ y,\ c_x,\ c_y]$ to a certain class i in G is given by

$$R_i(x, y, c_x, c_y) = \sum_{j \neq i} P_{ij} \times p(C_j|x, y, c_x, c_y), \tag{15}$$

where P_{ij} is the penalty of misclassifying a pattern from class i to class j and

$$p(C_j|x, y, c_x, c_y) = \frac{1}{|G|} \times \frac{p(x, y|C_j)p(c_x, c_y|C_j)}{\sum_{i \in G} p(x, y|C_i)p(c_x, c_y|C_i)}. \tag{16}$$

The pattern is assigned to the class that minimizes the overall risk. Here $p(x, y|C_i)$ and $p(c_x, c_y|C_i)$ are the class-conditional probability density functions, each of which is assumed to be a bivariate Gaussian distribution and $|G|$ is the number of pattern classes that comprise group G. The pattern classes within a group are assumed to be equiprobable with probability $\frac{1}{|G|}$. The mean vector and covariance matrix for each of the bivariate Gaussian distributions are estimated from the training samples.

5.2.3. Automatic grid refinement

Since the spot centroid position within the 4×4 cell is used as one of the features in the classifier, it is important to be able to position the grid accurately on the hybridization image. The initial position of the grid is determined using reference points marked on the film. These reference points denote the corners of the grid. Due to errors in the marking of the reference points, the initial grid placement may be inaccurate. This could have an adverse effect on the classification accuracy. With this in mind, an automated technique for refining the initial position of the grid was designed.

The automated technique for grid refinement uses spot patterns that are classified with high confidence as reference points. For a given spot pattern the product $P = D_M \cdot R$ is computed, where D_M is the Mahalonobis distance and R the Bayesian risk associated with the classification of that pattern. Smaller values of P can be associated with patterns that have been classified with high confidence. In our case, we choose patterns that are in the 5th percentile of the P values as ones that have been classified with high confidence. The sum of the P values of these patterns is regarded as the objective function to be minimized.

The grid refinement procedure is essentially an exhaustive search performed within a local window centered at the initial position of the grid. The grid is moved along the x and y axes in constant increments/decrements within the local window, scaled up and down within a scale range ([0.9, 1.1] in our case) in constant scale increments/decrements and rotated within an angular range ($[-5^o, +5^o]$ in our case) in constant angular increments/decrements. The position, orientation, and scale of the grid corresponding to the local minimum of the objective function within the window is deemed to be the best position. This procedure makes three key assumptions:

(a) The initial grid placement is close enough to the final optimal placement. This is critical since a local search process is sensitive to the choice of the starting point. If the initial solution is close enough to the desired globally optimal solution, then a local search process would yield the globally optimal solution.
(b) The initial grid placement is accurate enough to ensure that 5% of the spots have been classified correctly. This is tantamount to saying that the spots in the 5th percentile of the P values have the correct class labels assigned to them.
(c) The increments/decrements along the x and y axes, the scale range, and the angular range are small enough not to miss the locally optimum grid placement within the search window.

In our experiments, these assumptions were seen to be justified. Since the grid position, orientation, and scale affect the spot grouping and classification procedures, the output of the grid refinement procedure is fed back to the spot grouping procedure, as shown in Fig. 1. The grid refinement procedure was seen to improve the resulting classification accuracy, as shown in the experimental results.

6. Experimental results

The DNAScan program was tested on a set of 600 hybridization images. Manual classification of these images by trained laboratory personnel was used as the ground truth. Of these 600 images, 100 images were used for training, i.e., estimating the means and covariance matrices of the pattern classes. Results of manual classification indicated that there are a total of 16,257 positive hits. The deviation of the output of the DNAScan program from the result of manual classification includes

(1) Missing positive hits that cannot be detected due to low background contrast,
(2) Wrong grouping of spotlike objects, and
(3) Misclassification of patterns.

The classification results are summarized in Tables 4 and 5. Table 4 tabulates the results prior to the refinement of the grid, whereas Table 5 tabulates the results after the grid refinement procedure has been carried out. In both cases, the grouping was performed using the maximum weight vertex matching algorithm [29]. The first column in each of the tables denotes the true class labels (as determined via manual classification). Thus, each row represents a set of positive hits that have been manually assigned to the class indicated by the first column of that row. Each cell in the row indicates the percentage of positive hits assigned to the class indicated by the corresponding column by the DNAScan program. In Tables 4 and 5, U denotes the unknown class, M a miss due to poor contrast, and WG a miss due to wrong grouping. Table 4 shows that the correct detection rate is between 65% and 95%, depending on the pattern class. Also note that most of the misses are assigned to class U, which is a better situation than the one in which most of the misses are assigned to another pattern class since patterns assigned to class U could be potentially referred to a human for further consideration.

Table 5 shows a significant improvement in the classification results after the grid refinement procedure. Recall that the grid refinement procedure uses the results of the initial classification to refine the position, scale, and orientation of the grid. In our case, the grid refinement procedure was performed once, though in principle it could be carried out iteratively until there is no further improvement in the classification results. As can be seen from the results in Table 5, a large number of patterns assigned to class U are assigned to the correct class. This is particularly true of patterns belonging to classes 5, 6, and 7. The overall correct detection rate after grid refinement was observed to be between 82% and 95%.

7. DNAScan software

The DNAScan program was augmented with a user-friendly graphical user interface (GUI). The GUI allows the user to interactively perform the following tasks:

(a) Graphically place and adjust a rectangular grid over the hybridization image. This allows the user to manually translate, rotate, and scale the grid to best fit the underlying segmented image.
(b) Magnify/reduce portions of the segmented image or raw input image.

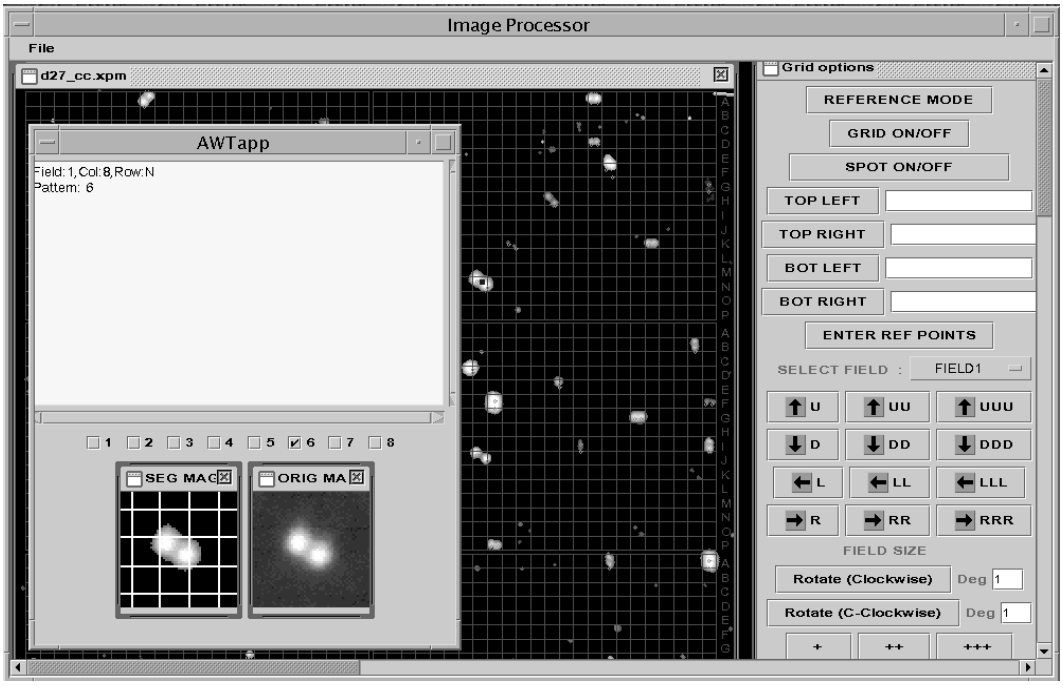


Fig. 14. GUI snapshot

Table 4. Classification results before grid refinement

Pattern class	U	1	2	3	4	5	6	7	8	M	WG
U	100%	0	0	0	0	0	0	0	0	0	0
1	0	93%	0	0	0	0	0	0	0	4%	3%
2	0	0	95%	0	0	0	0	0	0	0	5%
3	0	0	0	95%	0	0	0	0	0	2%	3%
4	0	0	0	0	94%	0	0	0	0	2%	4%
5	17%	0	0	0	0	75%	0	0	0	8%	0
6	14%	0	0	0	0	0	78%	0	0	6%	2%
7	25%	0	0	0	0	0	0	65%	0	8%	2%
8	25%	0	0	0	0	0	0	0	66%	7%	2%

Table 5. Classification results after grid refinement

Pattern class	U	1	2	3	4	5	6	7	8	M	WG
U	100%	0	0	0	0	0	0	0	0	0	0
1	0	94%	0	0	0	0	0	0	0	4%	2%
2	0	0	95%	0	0	0	0	0	0	0	5%
3	0	0	0	95%	0	0	0	0	0	2%	3%
4	0	0	0	0	96%	0	0	0	0	2%	2%
5	8%	0	0	0	0	84%	0	0	0	8%	0
6	6%	0	0	0	0	0	86%	0	0	6%	2%
7	8%	0	0	0	0	0	0	82%	0	8%	2%
8	7%	0	0	0	0	0	0	0	84%	7%	2%

(c) Visually monitor and observe the results of the segmentation and grouping processes.
(d) Query the classification results for each spot and manually edit/override the results of the classification procedure. The capability to manually override the results of the classification procedure is particularly useful in the case of spots that are assigned to the unknown class or spots that are wrongly grouped.

Figure 14 shows the snapshot image of the GUI.

8. Conclusions and future work

This paper presented a computer-vision-based system DNAScan for the analysis of DNA hybridization images. The image analysis was shown to consist of two stages, extraction

of patterns denoting positive hits followed by pattern classification. Extraction of positive hits was shown to involve recursive segmentation, grouping, and pattern decomposition for which algorithms were designed, implemented, and tested. In the second stage, which involved classifier design, a mathematical model for the hybridization protocol was proposed. The model was general and flexible enough to encompass a variety of experimental protocols. A two-stage Bayesian classifier based on the mathematical model was implemented and tested on several hybridization images with satisfactory results. The classification results were shown to improve significantly with the incorporation of a grid refinement procedure that corrected for errors in grid placement and grid orientation and for scale distortions.

Future work will involve the design and implementation of more powerful grouping algorithms. The current grouping algorithm does not exploit class-specific information in the dissimilarity function after the grid refinement procedure has been performed. The grouping algorithm will be extended to handle instances where class-specific information is available. The current model for the hybridization patterns is based on the statistical independence of the orientation, position, and shape features as well as the assumption that the features have an underlying Gaussian distribution. These assumptions will need to be relaxed in future versions of DNAScan. On the practical side, DNAScan will be extended to deal with fluorescently as well as radioactively tagged data.

Acknowledgements. This research was supported in part by a research grant (award number DBI-9872649-A000) from the National Science Foundation.

References

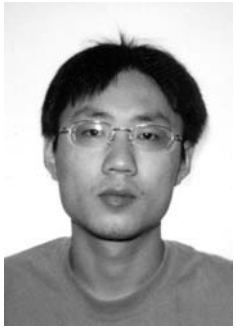
1. Affymetrix Inc (2002) GeneChip CYP 450 Assay. Santa Clara, CA
2. Agilent Technologies (2002) www.chem.agilent.com. Palo Alto, CA
3. Arnold J (1997) Editorial. Fungal Genet Biol 21:254–257
4. Audic S, Zanetti G (1995) Automatic reading of hybridization filter images. Comput Appl Biol Sci(5):489–495
5. Azuaje F (2002) A cluster validity framework for genome expression data. Bioinformatics 18:319–320
6. Ben-Dor A, Yakhini Z (1999) Clustering gene expression patterns. In: Proceedings of the ACM conference on research in comparative molecular biology (RECOMB), Lyon, France, April 1999, pp 33–42
7. Bennett JW (1997) White paper: genomics for filamentous fungi. Fungal Genet Biol 21:3–7
8. Bhalla US, Iyengar R (1999) Emergent properties of networks of biological signaling pathways. Science 283:381–387
9. Bhandarkar SM, Chirravuri S, Machaka S, Arnold J (1998) Parallel computing for chromosome reconstruction via ordering of DNA sequences. Parallel Comput 24(8):1177–1204
10. BioDiscovery Inc (2002) AutoGene v2.5. www.biodiscovery.com. Los Angeles
11. Brandle N, Chen H-Y, Bischof H, Lapp H (2000) Robust parametric and semi-parametric spot fitting for spot array images. In: Proceedings of the 8th international conference on intelligent systems for molecular biology, La Jolla, CA, 20–23 August 2000, pp 46–56
12. Brandle N, Bischof H, Lapp H (2001) A generic and robust approach for the analysis of spot array images. In: Proceedings of the SPIE conference on progress in biomedical optics and imaging: microarrays: optical technologies and informatics. San Jose, CA, 20–21 January 2001, 4266:1–12
13. Brown T (1999) Genomes. Wiley, New York
14. Brown CS, Goodwin PC, Sorger PK (2001) Image metrics in the statistical analysis of DNA microarray data. Proc Natl Acad Sci USA 98(16):8944–8949
15. Bouton CMLS, Pevsneri J (2002) DRAGON View: information visualization for annotated microarray data. Bioinformatics 18:323–324
16. Bumm K, Zhang M, Bailey C, Zhan F, Chiriva-Internati M, Eddlemon P, Terry J, Barlogie B, Shaughnessy Jr JD (2002) CGO: utilizing and integrating gene expression microarray data in clinical research and data management. Bioinformatics 18:327–328
17. Cai WW, Reneker J, Chow CW, Vaishnav M, Bradley A (1998) An anchored framework BAC map of mouse chromosome 11 assembled using multiplex oligonucleotide hybridization. Genomics 54:387–397
18. Chapman S, Schenk P, Kazan K, Manners J (2001) Using biplots to interpret gene expression patterns in plants. Bioinformatics 18(1):202–204
19. Chen T, He HL, Church GM (1999) Modeling gene expression with differential equations. In: Proceedings of the Pacific symposium on biocomputing, Big Island, HI, January 1999, pp 29–40
20. Chen H-Y, Brandle N, Bischof H, Lapp H (2000) Robust spot fitting for genetic spot array images. In: Proceedings of the international conference on image processing (ICIP), Vancouver, BC, Canada, 10–13 September 2000, pp 412–415
21. Chen T, Filkov V, Skiena SS (2001) Identifying gene regulatory networks from experimental data. Parallel Comput 27:141–162
22. Clemson University (2002) Clemson University Genomics Institute. www.genome.clemson.edu
23. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). J R Stat Soc B39:1–38
24. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 95:14863–14868
25. Evans GA, Lewis KA (1989) Physical mapping of complex genomes by cosmid multiplex analysis. Proc Natl Acad Sci USA 86:5030–5034
26. Filkov V, Skiena S, Zhi J (2001) Analysis techniques for microarray time-series data. In: Proceedings of the ACM conference on research in computational molecular biology (RECOMB), Montreal, pp 124–131
27. Friedman M, Kandel A (1999) Introduction to pattern recognition: statistical, structural, neural and fuzzy logic approaches. World Scientific, New York
28. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using Bayesian networks to analyze expression data. In: Proceedings of the ACM conference research in computational molecular biology (RECOMB), Tokyo, Japan, pp 127–135
29. Galil Z, Micali S, Gabow H (1986) An $O(EV \log V)$ algorithm for finding a maximal weighted matching in general graphs. SIAM J Comput 15(1):120–130
30. Garey MS, Johnson DS (1979) Computers and intractability: a guide to the theory of NP–completeness. Freeman, New York
31. Ghosh D, Chinnaiyan AL (2002) Mixture modelling of gene expression data from microarray experiments. Bioinformatics 18:275–286

32. Hall D, Bhandarkar SM, Arnold J, Jiang T (2001) Physical mapping with automatic capture of hybridization data. Bioinformatics 17(3):205–213
33. Hartuv E, Schmitt A, Lange J, Meier-Ewert S, Lehrach H, Shamir R (1999) An algorithm for clustering cDNAs for gene expression analysis. In: Proceedings of the ACM conference on research in computational molecular biology (RECOMB), Lyon, France, April 1999, pp 188–197
34. Hu MK (1962) Visual pattern recognition by moment invariants. IRE Trans Inf Theory IT-8:179–187
35. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Englewood Cliffs, NJ
36. Jain RC, Kasturi R, Schunk BG (1995) Machine vision. McGraw-Hill, New York
37. Jain AN, Tokuyasu TA, Snijders AM, Segraves R, Albertson DG, Pinkel D (2002) Fully automatic quantification of microarray image data. Genome Res 12:325–332
38. Jansen R, Greenbaum D, Gerstein M (2002) Relating whole-genome expression data with protein-protein interactions. Genome Res 12:37–46
39. Kass RE, Raftery JE (1995) Bayes factors. J Am Stat Assoc 90:773–795
40. Kececioglu JD, Myers EW (1995) Combinatorial algorithms for DNA sequence assembly. Algorithmica 13:7–51
41. Lashkari DA, DeRisi JL, McCusker JH, Namath AF, Gentile C, Hwang SY, Brown PO, Davis, RW (1997) Yeast microarrays for genome wide parallel genetic and gene expression analysis. Proc Natl Acad Sci USA 95:13057–13062
42. Leach S, Hunter L (2000) Compartive study of clustering techniques for gene expression microarray data. In: Miyano S, Shamir R, Takagi T (eds) Currents in computational molecular biology. Universal Academy Press, Tokyo, pp 1–2
43. Manduchi E, Grant GR, McKenzie SE, Overton GC, Surrey S, Stoeckert CJ (2000) Generations of patterns from gene expression data by assigning confidence to differentially expressed genes. Bioinformatics 16(8):685–698
44. McLachlan GJ, Bean RW, Peel D (2002) A mixture model-based approach to the clustering of microarray expression data. Bioinformatics 18:413–422
45. Michaels GS, Carr DB, Askenazi M, Fuhrman S, Wen X, Somogyi R (1998) Cluster analysis and data visualization of large scale gene expression data. In: Proceedings of the Pacific symposium on biocomputing, Big Island, HI, 3:42–53
46. Pan W (2002) A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. Bioinformatics 18(4):546–554
47. Pan W, Lin J, Le C (2002) Model-based cluster analysis of microarray gene expression data. Genome biology variables and stochastic processes. McGraw-Hill, New York
48. Papoulis A (1965) Probability, random variables and stochastic processes. McGraw-Hill, New York
49. Piper J, Rutovitz D, Sudar D, Kallioniemi A, Kallioniemi O, Waldman F, Gray J, Pinkel D (1995) Computer image analysis of comparative genomic hybridization. Cytometry 19:10–26; 3(2):research0009.1–research0009.8
50. Preparata FP, Shamos MI (1991) Computational geometry: an introduction. Springer, Berlin Heidelberg New York
51. Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1988) Numerical recipes in C. Cambridge University Press, Cambridge, UK
52. Roth K, Wolf G, Dietel M, Peterson I (1997) Image analysis for comparative genomic hybridization based on a karyotyping program for Windows. Anal Quantit Cytol Histol 19(6):461–473
53. Sahibsingh AD, Breeding KJ, McGhee RB (1977) Aircraft identification by moment invariants. IEEE Trans Comput 26(1):39–45
54. Samet H (1990) The design and analysis of spatial data structures. Addison-Wesley, Reading, MA
55. Scanalytics Inc(2002) Scanalytics Inc, Fairfax, VA. www.scanalytics.com
56. Schena M, Shalon D, Heller R, Chai A, Brown PO, Davis RW (1996) Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. Proc Natl Acad Sci USA 93:10614–10619
57. Shamir R, Sharan R (2000) CLICK: a clustering algorithm for gene expression analysis. In: Miyano S, Shamir R, Takagi T (eds) Currents in computational molecular biology. Universal Academy Press, Tokyo, pp 6–7
58. Sigma-Genosys Inc (2002) Sigma-Genosys, The Woodlands, TX. www.sigma-genosys.com
59. Spectral Genomics (2002) Spectral Genomics, Houston, TX. www.spectralgenomics.com
60. Stanford University (2002) Stanford University Genomic Resources. www-genome.stanford.edu
61. Steinfath M, Wruck W, Seidel H, Lehrach H, Radelof U, O'Brien J (2001) Automated image analysis for array hybridization experiments. Bioinformatics 17(7):634–641
62. Sturn A, Quackenbush J, Trajanoski Z (2002) Genesis: cluster analysis of microarray data. Bioinformatics 18:207–208
63. Theodoridis S, Koutroumbas K (1999) Pattern recognition. Academic, San Diego

Suchendra M. Bhandarkar received his B.Tech. in electrical engineering from the Indian Institute of Technology, Bombay, India in 1983 and his M.S. and Ph.D. in computer engineering from Syracuse University, Syracuse, NY in 1985 and 1989, respectively. He is currently a professor and director of the Visual and Parallel Computing Laboratory (VPCL) in the Department of Computer Science at the University of Georgia, Athens, GA, USA. He was a Syracuse University Fellow for the academic years 1986–1987 and 1987–1988. He is a member of the IEEE, AAAI, ACM, and SPIE and the honor societies Phi Kappa Phi and Phi Beta Delta. He is a coauthor of the book *3-D Object Recognition from Range Images* (Springer, 1992) and an associate editor of *The Computer Journal* and the *Journal of Applied Intelligence*. His research interests include computer vision, pattern recognition, image processing, artificial intelligence, and parallel algorithms and architectures for computer vision and pattern recognition. He has over 80 published research articles in these areas including over 40 articles in refereed archival journals. He has also served on the program committees of several international conferences in these areas.

Tongzhang Jiang received his B.S. in chemistry from the Qingdao University of Oceanography, P.R. China in 1992. He received his M.S. in computer science from the University of Georgia, Athens, GA, USA in 2000. Currently he is a software engineer at UpToDate Inc. in Wellesley, MA, USA. His professional and research interests are software engineering and human computer interface design.

Kunal Verma is a research assistant and Ph.D. student in the Computer Science Department at the University of Georgia. He completed his B.S. in engineering in electronics and telecommunications from the University of Bombay in August 1999. His research interests span Web processes, databases, and machine vision. He has been associated with the VPCL and LSDIS Labs at the University of Georgia. His current area of research is the role of semantics for creating Web processes with greater expressiveness and power. During summer 2003 he worked at the IBM T.J. Watson Research Center on adding dynamic binding to BPEL4WS using semantic Web technologies.



Nan Li graduated from Shanghai Jiao Tong University, Shanghai, P.R. China in 2000 with a B.Eng. degree in computer science and engineering. During his undergraduate study at Shanghai Jiao Tong University he received the Alcatel scholarship in 1999. He is currently pursuing his M.S. in computer science at the University of Georgia. He is also an educational programming specialist for the Learning and Performance Support Laboratory in the College of Education at the University of Georgia. He is a student member of ACM and USENIX. His current research interests are in computer systems, including compilers, operating systems, and networking. During the past year he has done research on energy-efficient compiler/linker techniques. He also has an interest in systems administration and advocates free software.