American Journal of Applied Sciences 3 (2): 1730-1734, 2006
ISSN 1546-9239

cDNA Microarray Genome Image Processing Using Fixed Spot Position

Basim Alhadidi, Hussam Nawwaf Fakhouri and Omar S. AlMousa
Information Technology Department, Al Balqa' Applied University, Salt, Jordan

Abstract: Complementary DNA (cDNA) microarray is one of the most recent and important technology for exploring the genome. cDNA microarray Image analysis aims to measure the intensity for each spot in the scanned image and this intensity represents the amount of a specific gene in the studied cell. This paper presents a new algorithm that achieves an automated way for applying grid in the cDNA microarray images through the determination of spots position in the cDNA microarray and calculating the mean of every raw and column in the microarray image and the result will be a fast automated system for applying addressing (grid) on the microarray image without any human interaction and this automated system starts from the step that follow the sub microarray image cropping.

Keywords: Microarray, spot position, DNA chip, cDNA grid

INTRODUCTION

cDNA microarray image analysis is a very important technology for genome mapping , it represent both biology and computers[1], microarray slide contain thousand of small spots each spot represent genetic material and these slides after they are scanned give images that contain colored spots each spot intensity represent the amount of the expression level of genes[2], these images because they contain large numbers of spots need to be analyzed in a very fast automated way and this analysis is very important in discovering the cause of diseases and the best drug to treat it (ex. Discovering the gene that cause the cancer in the cell and the drug that suppress that gene.)

Microarray image grid and spot position determination is a very important step in the analysis of microarray image because it is the first part you need to do for the analysis and making this part automated and fast is also important that's why we propose our algorithm for microarray image grid (addressing) and spot position determination.

Several researches were done in the past but all the method were not fully automated and required human interaction as Stefano Lonardi and Yu Luo from Department of Computer Science and Engineering in University of California tested in their paper[3] they wrote : Several methods have been proposed and software tools have been developed. However, all the software systems they tested require human intervention. At the minimum, these software require the user to specify the geometry of the array, such as the number of grids, number of rows and columns, etc. (see, for example, SPOT from UCSF[4], IMAGENE from BioDiscover[5] and DAPPLE from University of Washington[6]).

Dapple, a new spot finding implementation for microarrays on glass slides. Dapple finds spots using morphological information which is robust to both variation and artifacts. It achieves high spot finding throughput and accuracy by learning to evaluate the quality of candidate spots from examples supplied by the user. Dapple's techniques are useful for improving the accuracy of data acquisition from DNA microarrays[6].

Mathias, Franz & Gerhard proposed microarray technology and MRF model of microarray gridding that is designed to integrate different application specific constraints and heuristic criteria into a robust and flexible segmentation algorithm[7].

Also there were other researchers who tried to fined deferent ways for gridding for example. Lonardi and Yu Luo assumed that the distance between adjacent cells should be approximately equal in order to find the grid which is not equal in real microarray scanned image, Images from microarray experiments are highly structured since they are composed by high intensity spots located on a regular grid. The shape of the spots is roughly circular, although variations are possible[3].

The ideal microarray image has the following properties[8] : (1) all the subgrids are of the same size; (2) the spacing between subgrids is regular; (3) the location of the spots is centered on the intersections of the lines of the subgrid; (4) the size and shape of the spots are perfectly circular and it is the same for all the spots; (5) the location of the grids is fixed in images for a given type of slides; (6) no dust or contamination is on the slide; (7) there is minimal and uniform background intensity across the image. It goes without saying that almost all real microarray images don't have at least one of these conditions. In fact, there are frequently observed variations on the spot position, irregularities on the spot shape and size, contamination

Corresponding Author: Basim Alhadidi, Information Technology Department, Al Balqa Applied University, Salt, Jordan

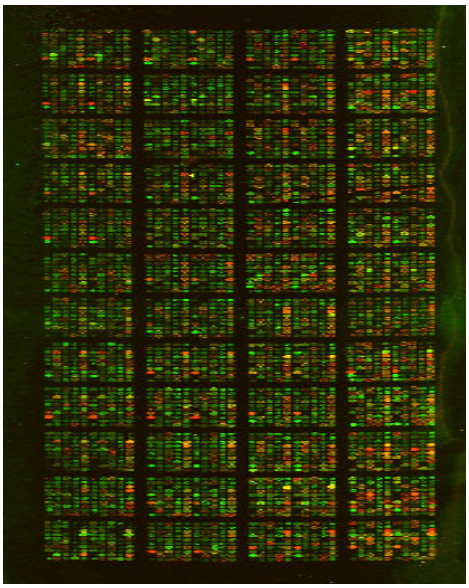

Fig. 1: Microarray Image

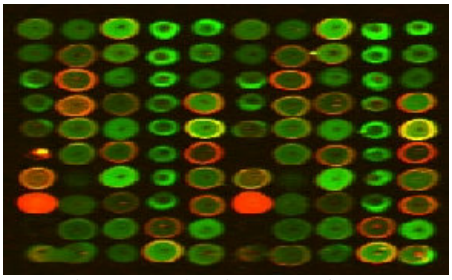

Fig. 2: Sub microarray image

and global problems that affect multiple spots, in general, the shape and the size of the spots may fluctuate significantly across the array[3], our new algorithm goes over the fact that real image don't have at least one of the above condition for the ideal conditions microarray image and determine the spot position and grid for the microarray image

MATERIALS AND METHODS

Microarray image that shown in Fig. 1 is stained using two dyes, these dyes are Cy5, red and Cy3, green and each Microarray image consist of large number of n sub microarrays and each sub microarray contains **A** columns and **B** rows of gene spots each spot is corresponding to the location of a cDNA probe to which mRNA from the cells of interest have been bound. In some of the experiments here, there are approximately 4800 spots on a slide, arranged in a 4 by 12 grid of sub microarrays, with each sub microarray being a 10 by 10 gene spots. The image of the slide gives a 3248 by 1248 array of grayscale pixel values. Each pixel is a 16-bit intensity value and we obtained these images and the necessary information regarding these images from[9,10]. We used Matlab software to implement the algorithm because Matlab is a high-performance language for education and research It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation and also it has toolboxes for signal processing , neural network, image processing , database … etc , Matlab Image Processing Toolbox is a collection of functions that extend the capability of the Matlab numeric computing environment. The toolbox supports a wide range of image processing operations, such as Image analysis and enhancement, Region of interest operations, Linear filtering and filter design[11].

PROPOSED ALGORITHM, RESULTS AND DISCUSSION

We will start discussion the algorithm from the step that follows resizing and cropping the sub determination since each pixel in the RGB image is determined by the combination of the red, green and blue intensities stored in each microarray that we want to analyze and in Fig. 3 we show the proposed algorithm flowchart. The first step is converting the cropped sub microarray that consists of A rows and B columns of genes that shown in Fig. 2 from RGB images to grayscale as shown in Fig. 4 by eliminating the hue and saturation information while retaining the luminance, this step mainly aims to enhance spots position color plane at the pixel's location with 8 bits for each color, giving a potential of 16 million colors and make it difficult to determine the position of the spot using all this combination; so we need to convert it to gray scale image.

The gray image can be represented by the function:

$$f: D_f \longrightarrow x \qquad (1)$$

Where D_f is a real number that represent an ordered set of gray levels, its values is from x_{min} to ,x_{max} . f is the gray value of the image at point x = (X , Y), The lighter the gray value of f at point x, the higher the altitude of the corresponding point {x , f } on the surface of t(e image. So the lower points and the zero values may represent the points the spaces between genes like the background points and the points that contain noise[12]; if the value of the function f = 0 then this point may represent the background and it location is at a point (X , Y) around the spot and by measuring the space between the highest point of two adjacent spots will give the mid point (X , Y) that lies in the line L that represent the center point between them in a more general form by taking 4 spots located in the positions (X1 , Y1) , (X2 , Y2) , (X3 , Y3) , (X4 , Y4) where the point (X1 , Y1) lies on the top left , the point (X2 , Y2) on the top right , point (X3 , Y3) on the lower left and the point (X4 , Y4) on the bottom right then by determining the mid point between each 2 spots

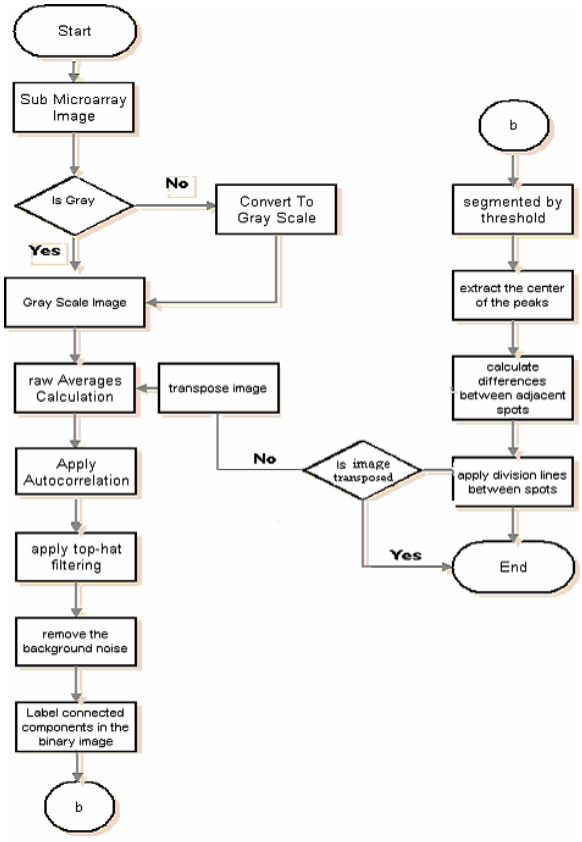

Fig. 3: Flow chart of the proposed algorithm

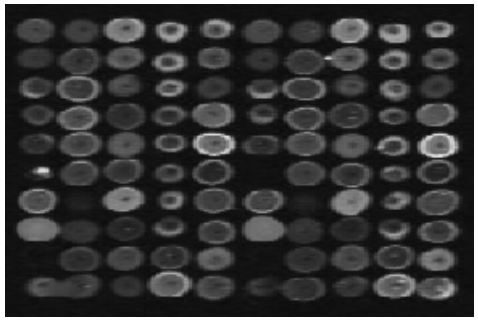

Fig.4: Gray sub microarray image

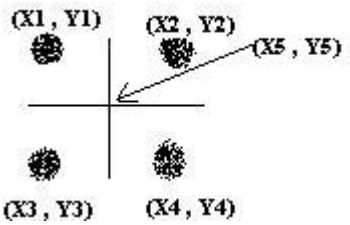

Fig. 5: Mid point between spots

(horizontal and vertical)and cross it with the line we can determine a point (x5,y5) that represent the coordinate of the rectangle that surround each spot as shown in the Fig. 5.

The second step of the algorithm is generalizing the previous 4 points by analyzing all the spots together so in this step calculates the average of the **A** * **B** sub

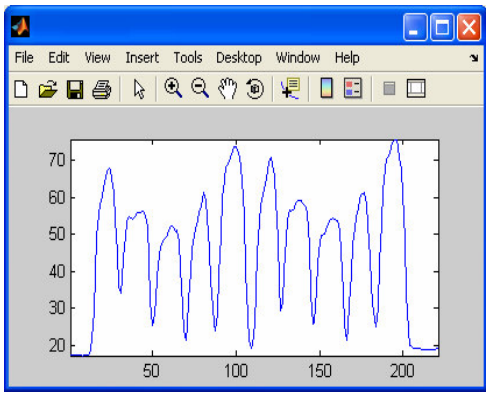

Fig. 6: Average of sub microarray image

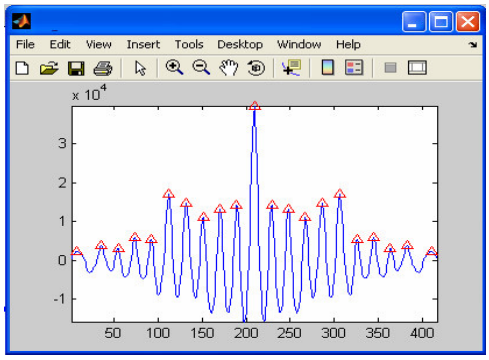

Fig. 7: Applying autocorrelation

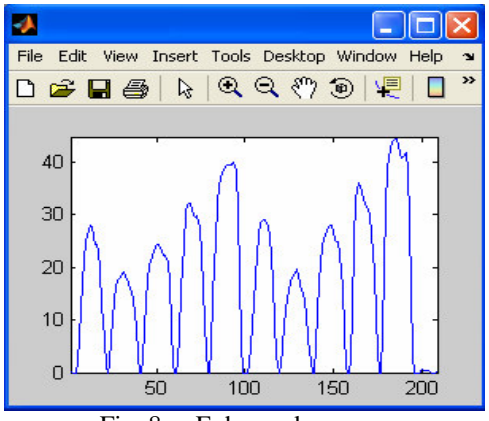

Fig. 8: Enhanced average

microarray. First we calculate the average of **A** columns in the image matrix then for B rows of the image matrix as shown in the Fig. 6.

The average of the column **A** and the raw **B** is irregular, because the intensity of the genes spots in the sub Microarray is irregular and because the size of each spot is different one from the other so we need to enhance it in order to be considered for determining the med points (X5 , Y5) and the coordinate of the rectangle that surround the spots.

Since spots have different sizes and intensities and both vertical and horizontal averages are irregular. So in step three of the algorithm we used autocorrelation to enhance the self similarity of the vertical and horizontal averages as shown in Fig. 7. The smooth result

promotes peak finding and estimation of spot spacing. The autocorrelation is applied by Estimating the cross-covariance where The Signal Processing Toolbox in Matlab allows easy computation of the autocorrelation function using the xcov command for the calculated mean (average) of the rows of the cDNA microarray image , xcov estimates the cross-covariance sequence of random processes and By default, xcov computes raw covariance with no normalization. For a length N vector[11] where vector is a raw or column in the array that represent the microarray image.

Figure 8 shows the average after the autocorrelation is applied as seen that the average now is more regular than the previous which is shown in Fig. 6, the peaks contain the maximum value has the (X_{max} and Y_{max}) values points which is the points that represent the mid line L_{max} that has most intensity expressed genes in the **A** columns or **B** raws, this step will help in the determination of the spaces between spots. The points (X, Y) that contain minimum values will mostly represent the noise between spots

Step four is eliminating the point (X ,Y) that contains the minimum points (X_{min} , Y_{min}) and located in the space points between the intensity values (X_{max} , Y_{max}) and these stedp remove the noise from the background .As a result of autocorrelation an enhanced average resulted; so we performed morphological top-hat filtering on the grayscale input image using the structuring element SE, where SE is a single structuring element object. After that we create line-shaped structuring element, then spacing estimate is applied to design a filter to remove the background noise from the intensity by creating a simple rectangular line shaped structuring element.

Step five aims to Label connected components in the binary image after removing the noise from the image , we used the algorithm of the general procedure outlined[13] to number every peak region with a label before finding its center as described in the following paragraph, the algorithm in can be summarized by first Run-length encode the input image. Then Scan the runs, assigning preliminary labels and recording label equivalences in a local equivalence table. After that resolve the equivalence classes. Create image by using threshold. Compute an appropriate threshold to use it to convert the intensity image to binary; in binary image, each pixel assumes one of only two discrete values. Essentially, these two values correspond to on and off. A binary image is stored as a logical array of 0's (off pixels) and 1's (on pixels). Then using threshold that will segment peck region to convert this grayscale image to binary. The output binary image has values of 0 (black) for all pixels in the input image with luminance less than level and 1 (white) for all other pixels.

We can extract the center of the peaks that correspond to the horizontal centers of the spots as shown in Fig. 9 by measuring a set of properties for

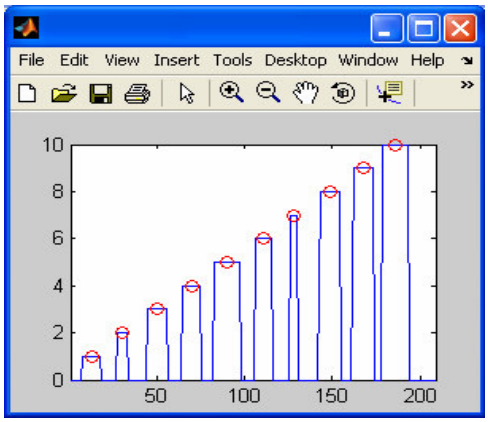

Fig. 9: Center of peaks

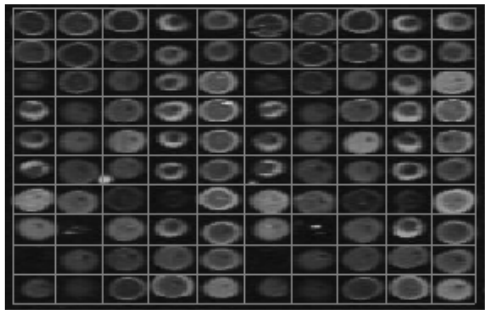

Fig. 10: Center of peaks

each labeled region in a labeled matrix L of the microarray image the Positive integer elements of L corresponded to different regions For example, the set of elements of L equal to 1 corresponds to region 1; the set of elements of L equal to 2 corresponds to region 2; and so on. L : is column or row at the center of each spot. Note that the first element is the horizontal coordinate (or x-coordinate) of the center of each spot and the second element is the vertical coordinate (or y-coordinate).

Now we calculate differences between adjacent spots of the sub microarray image. If X is a vector(row or column), then diff(X) returns a vector, one element shorter than X, of differences between adjacent elements: [X(2)-X(1) X(3)-X(2) ... X(n)-X(n-1)] , while X is a matrix, then diff(X) returns a matrix of row differences: [X(2:m,:)-X(1:m-1,:)]. In general, diff(X) returns the differences calculated along the first non-singleton (size(X,dim) > 1) dimension of X. Y = diff(X,n) applies diff recursively n times, resulting in the nth difference and this difference is the division between spots and the crossing points of the division lines will be the coordinate of the rectangle that surround each spot and this step of the algorithm is followed by repeating all the steps for the columns as for the rows. by transposing the image and repeat all the steps above and the result applied to the gray image as shown in Fig. 10.

The proposed algorithm depended on calculation based on all spots position distributed on the cDNA

Microarray images rather than trail and error techniques as used in the MD Anderson Cancer Center in the University of Texas, they wrote in[10]: we start by gridding the 10 by 10 dot setup, identifying good cut points. We do this by playing with the axis command, trimming off bits at a time. Good dividing values found by trial and error. While trail and error techniques results took to much time to get them and to analyze large microarray spot genes and also it is manual and slow while our proposed algorithm is automated and take only few second to analyze a sub microarray that consist of 100 gene spot (10 raw genes and 10 columns).

CONCLUSION AND FUTURE WORK

This paper described a new automated algorithm for determining spot position and grid for the cDNA the microarray image, it improves gene position determination by making it automated that requires no human interaction, also because it is very fast method that requires seconds for the determination of coordinates of the rectangle that surrounds the gene spot in the microarray image and apply the grid.

Microarray image analysis is very interesting research area that's why our future work intended to pursue the research for developing a new algorithm for extracting the spot from the background, enhance the microarray image and calculating the intensity for each spot.

ACKNOWLEDGMENTS

Authors would like to thank the presidency of Al-Balqa' Applied University for encouragement of research and researchers at the university, also authors would like to thank to the deanship of The Prince Abdullah Bin Gazi Faculty of Science and Information Technology. Finally thanks are due to the scientific research committee and deanship of The Graduate Studies and Scientific Research at Al-Balqa Applied University.

REFERENCES

1. Istepanian, R., 2003. Microarray image processing: Current status and future directions. IEEE Tran. Nanobioscience, 2: 4.
2. www.genome.gov
3. Lonardi, S. and Y. Luo, 2004. Gridding and compression of microarray images. IEEE Computational Systems Bioinformatics Conf. (CSB'04) Stanford, CA.
4. Jain, A.N., T.A. Tokuyasu, A.M. Snijders, R. Segraves, D.G. Albertson and D. Pinkel, 2002. Fully automatic quantification of microarray image data. Genome Research.
5. Kuklin, A., 2000, Laboratory automation in microarray image processing. American Laboratory.
6. Buhler, J., T. Ideker and D. Haynor, 2000. Dapple: Improved techniques for finding spots on DNA microarrays. Technical Report
7. Katzer, M., F. Kummert and G. Sagerer, 2003. A Markov random field model of microarray gridding. Proc. 18th ACM Symp. Applied Computing.
8. Kuklin, A., 2000. Laboratory automation In microarray image processing. American Laboratory.
9. Coombes, K.R., L. Zhang, C. Bueso-Ramos, S. Brisbay, C. Logothetis, J. Roth, M.J. Keating and T.J. McDonnell, 2002. TAD: A web interface and database for tissue microarrays. Appl. Bioinformatics.
   Coombes KR, Zhang L, Bueso-Ramos C, Brisbay S, Logothetis C, Roth J, Keating MJ, McDonnell TJ. TAD: a web interface and database for tissue microarrays. 2002 Appl. Bioinformatics.
10. Baggerly, K. The University of Texas M. D. Anderson Cancer Center: http://bioinformatics.mdanderson.org/ tut-image. html and Graphical Statistics.
11. Matlab 7 Image Processing Toolbox, Signal Processing Toolbox.
12. Angulo, J. and J. Serra, 2003. Automatic analysis of DNA microarray images using mathematical morphology. Bioinformatics.
13. Haralick, R.M. and L.G. Shapiro, 1992. Computer and Robot Vision. Vol. I. Addison-Wesley.