Processing of Microarray Images

Fernando Mastandrea and Álvaro Pardo

Department of Electrical Engineering - Universidad Católica del Uruguay
fmastand@ucu.edu.uy, apardo@ucu.edu.uy

Abstract. In this paper we present the results of a system for processing microarray images which includes the gridding and spot detection steps. The main goal of this work is to develop automatic methods to process microarray images including confidence measures on the results. The gridding step is based on the method proposed in [1] and improves it by the automatic determination of the grid parameters, and a more precise orientation detection. For spot detection the algorithm uses the Number of False Alarms methodology [2] which can be used to finely adjust the spot position and provides a confidence measure on the detection. We test the results obtained by our method with simulated images against existing microarray software.

1 Introduction

Microarray technology allows comparative experiments on gene expression. Each array consists on thousands of regularly placed spots which contain control and test samples. The samples are labeled with two different green (Cy3) and red (Cy5) fluorescent dyes. After the biological reaction takes place the digital image is obtained using a microarray scanner. The intensity of each pixel indicates the hybridization of the control and test samples. Once the image has been acquired, it is processed to extract features at each spot. After that, statistical processing is used to reveal the gene expression levels.

The analysis of microarray images involves the detection of the features that will later be used to infer the results on the experiment. The statistics that will be gathered for inference will be calculated from the pixel values of the detected spots. Therefore the correct detection of spots is crucial for data extraction.

The processing of microarray images can be divided into three steps: grid detection, spot detection and data extraction [4]. In the first step a template of the grid must be adjusted to the acquired image. In this step it is also useful to automatically learn some parameters of the grid, like spacing between spots, angles, etc. The result of this step provides candidate centers for each spot. Based on this information the detection of each spot is refined. Finally, in the last step, information from each spot is extracted for later analysis.

Although at first glance these problems may seem simple, microarrays may contain noise and distortions that deteriorate the results of these steps: small dots/speckles which can be confused with spots, artifacts contaminating the spot, missing spots (blanks), donut shaped spots and also entire columns (or rows) can be very dim, among others.

In order to assist the technician in the process of microarray analysis we need automatic and semiautomatic methods which provide a confidence measure on the detection

E. Bayro-Corrochano and J.-O. Eklundh (Eds.): CIARP 2009, LNCS 5856, pp. 962–969, 2009.

results. In order to reach that goal we use results from the Computational Gestalt Theory (see Section 2.2) which gives a confidence measure that can be used by the user.

As mentioned earlier, we based part of the grid detection in [1], but it has some differences. First, the minimum and maximum radii of a typical spot are estimated automatically. Second, the angles which represent the orientation of the grid are found by interpolation, giving better accuracy without being too computationally expensive. Third, we don't use a regular grid model nor MRF to refine spot center coordinates. However, our gridding method allows having different distances between rows and columns of spots and the spot center coordinates are refined by the segmentation algorithm. We also obtain the grid coordinates with subpixel accuracy. As for the segmentation, we added a method for spot detection which includes confidence estimation on the detection (see Section 2.2).

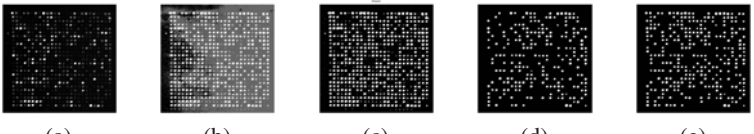
2 Proposed Method

2.1 Gridding

The gridding step consists in finding the approximate spot center coordinates. This information will be used later in the segmentation step. Usually, a microarray is composed of several grids arranged in matrix form in the image. The subgrids follow the same layout of spot rows and columns. Since the layout parameters are known beforehand here we concentrate on the extraction of spots on subgrids. The proposed algorithm has only two parameters: the number of rows and columns of the grid, sr and sc respectively. In Fig.5 we show the GUI of the developed software.

Radius estimation. In this step we estimate the mean spot radius using a heuristic procedure. For now on if the image is bi-channel, the average intensity image I is used (see Fig. 1). First we apply histogram equalization and stretching to I (see Fig. 1(b)). Next, we compute k different thresholds so that $t_i = \frac{i}{k} \cdot (I_h - I_l) + I_l$ with $i = 1 \ldots k$. Where I_h and I_l correspond to the maximum and minimum of I. Now for every iteration i, we apply the threshold t_i to the image obtaining U_t. We show in Fig. 1(c) for a threshold t_i with $i = k/2$. For every U_t we apply the following algorithm:

1: Remove isolated pixels and fill holes in U, store the result in R.
2: Remove all regions of R which area is less than $\pi * (R_{min})^2$.
3: Label remaining regions of R.
4: **if** there is more than one region left **then**



(a) (b) (c) (d) (e)

Fig. 1. (a) Input grid image. (b) Grid image followed by histogram equalization and stretching. (c) Thresholded grid image. (d) Regions passing constraints. (e) Final binary image.

5: Discard regions based on compactness[1] ($0.75 < c < 1.25$) and eccentricity[2] ($0 < e < 0.5$).
6: Generate a binary image B from all remaining regions.
7: Perform bitwise OR with I, so that: $I = I|B$
8: **end if**

After the k iterations the result is a binary image which contains most spots (see Fig. 1(e)). Finally we take all regions and calculate their median area $\bar{a}$ and estimate the radius as $\hat{r} = \sqrt{\frac{\bar{a}}{\pi}}$. The parameter k is the threshold granularity, we use $k = 10$. R_{min} is used for filtering out artifacts and a value of 2 pixels seemed satisfactory in our experiments.

Angle estimation. The first step in the gridding process is to identify the angles α and β that determine the orientation of the grid columns (sc) and rows (sr) of spots. This step is partly based in the procedure described in [1]. The following paragraphs are a brief explanation of the the orientation estimation procedure presented in [1].

Initially we apply the Orientation Matching (OM) transform to the grid image $I(x, y)$ obtaining $OM\{I\}(x, y)$. Given the percentage v of spot radius variability we define the following maximum and minimum radii for the OM transform: $R_M = r(1 + v/100)$ and $R_m = r(1 - v/100)$). The OM transform provides us with an image intensity values in the range $[-1, 1]$ which represent the match between the gradient of the image and the normals of an annulus of R_m and R_M radii centered on (x, y). The OM transform of the grid image of Fig.1 is shown on Fig.2. We filter the OM image with a median filter of size $[5 \times 5]$ since our experiments had shown the following steps benefit from an image with less noise. Next, we apply the Radon Transform (RT), obtaining $\mathcal{R}\{OM(I)\}(s, \phi)$. We then integrate the s variable obtaining $\Gamma(\phi) = \int_s \mathcal{R}^2\{OM(I)\}(s, \phi)\, ds$. Then we low-pass filter $\Gamma(\phi)$ (with the same parameters as specified in [1]) and take the two maximum values $m_1 = \Gamma(\phi_a)$ and $m_2 = \Gamma(\phi_b)$ corresponding to the principal orientations ϕ_a and ϕ_b (see Fig. 2(b)). A typical grid with little rotation will have it's maxima around $\phi \approx 90$ and $\phi \approx 180$. So we choose ϕ_a as the radon angle closer to $90°$, and ϕ_b closer to $180°$. Up to this point the method is the same as in [1].

Because the maxima are expected to be at $90°$ and $180°$, we calculate ϕ_i ranging from $45°$ to $225°$ instead of from $0°$ to $180°$. So ϕ is the vector $\phi = \{\phi_0, \phi_i, \ldots, \phi_n\}$, where $\phi_0 = 45$ and $\phi_n = 225$. We do that in order to avoid having a maximum of $\Gamma(\phi)$ which would correspond to the grid orientation at the end point. It follows the higher is n the more precision we have for $\Gamma(\phi)$. However, since the RT has to be calculated with a given angle step, increasing n to obtain more precise angles is more computationally demanding. As an alternative we can propose interpolation.

Angle interpolation. Given ϕ_a and ϕ_b we improve angle estimation via interpolation. We find a new interpolated angle α by taking the points $(\phi_{a-1}, \Gamma(\phi_{a-1}))$, $(\phi_a, \Gamma(\phi_a))$ and $(\phi_{a+1}, \Gamma(\phi_{a+1}))$ and solve the parabola $y = ax^2 + bx + c$ that passes through these

[1] $c = \frac{P^2}{4\pi A}$.
[2] Defined as the ratio of the distance between the foci of the ellipse and its major axis length. $0 \leq e < 1$.

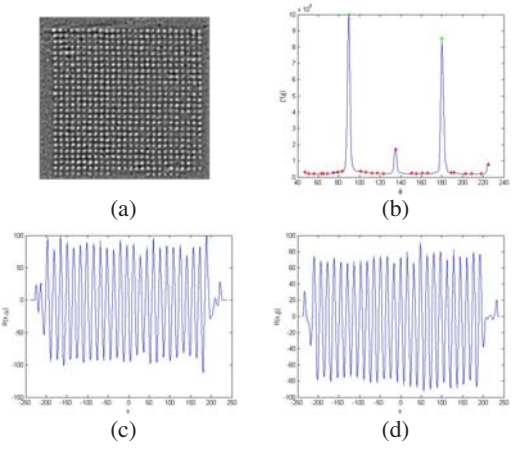


Fig. 2. (a) OM transformed image of the grid. (b) $\Gamma(\phi)$ graph. The peaks correspond to the angles ϕ_a and ϕ_b. Profiles (c):$\alpha = 89.616$ and (d):$\beta = 180.15$.

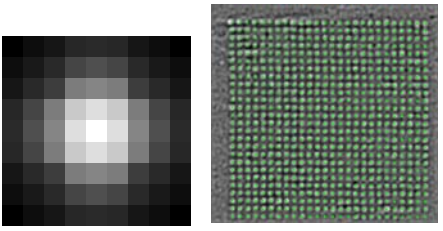
points. We then take the maximum value as the new $\alpha = -b/(2a)$. In the same way the anlge β is computed.

Obtaining grid rows and columns. Given the new interpolated angles, we find the profiles given by rotating (using interpolation) the grid image with α and β. This step provides us of two profiles, which are the radon profiles $\mathcal{R}\{OM(I)\}(s,\alpha)$ and $\mathcal{R}\{OM(I)\}(s,\beta)$ (see Fig. 2).

To simplify the notation we use $\mathcal{R}(s,\phi)$ instead of $\mathcal{R}\{OM(I)\}(s,\phi)$. Now, for every s, $\mathcal{R}(s,\alpha)$ represents the OM image projected with the direction α. The angle α is such that the local maxima of $\mathcal{R}(s,\alpha)$ indicate the spacing between the rows of the grid, as seen in Fig.2. Lets call the radon distances where the local maxima occur $\{s^\alpha\} = \{s_1^\alpha, s_2^\alpha, \ldots, s_{n_1}^\alpha\}$. In a similar way, the maxima of $\mathcal{R}(s,\beta)$ indicate the spacing of the columns, and we call it $\{s^\beta\} = \{s_1^\beta, s_2^\beta, \ldots, s_{n_2}^\beta\}$. So, we consider normal lines originating at the center of the image, with angle α and distance $\{s^\alpha\}$ and call them $\{l^\alpha\}$. We do the same with angle β and distance $\{s^\beta\}$, getting $\{l^\beta\}$. We are using the peaks in the profile as indicators of the location of the spots rows and columns. We are going to use this information to construct a non-regular grid (not the same distance between spot rows and columns).

Up to this point we have the profiles in function of s. Using the same ideas as in **Angle interpolation** we find the maximum values of the profiles. The process is the same, but instead of interpolating the angles vector we apply it on the distances vector s. We obtain a set of interpolated distances which we call $\{s^\alpha\}$ and $\{s^\beta\}$. These distances, in subpixel resolution, represent the spot rows and columns plus noise, as we will see next.

Filtering erroneous maxima. If the array image was ideal, we would expect to find the spot centers in the intersection of the two set of lines $\{C\} = \{l_\alpha\} \cap \{l_\beta\}$. Or equivalently, we would expect for $\{s^\alpha\}$ and $\{s^\beta\}$ to have $n_1 = sr$ and $n_2 = sc$ elements



(a) (b)

Fig. 3. (a) Spot template. (b) The output of the gridding method.

respectively. But, in a real grid image, noise takes part in erroneous detection of the maxima of $\mathcal{R}(s, \alpha)$ and $\mathcal{R}(s, \beta)$ as can be seen in Fig. 2. Note that, in the first profile we have 29 maxima and $sr = 26$. To filter out these false maxima we propose the following iterative procedure to remove elements in the set of distances $\{s^\alpha\}$ and $\{s^\beta\}$. We iterate through the distance differences vector d, and if some element falls below the threshold T, the algorithm removes the element of $\{s\}$ so that the new $\{s\}$ has differences closer to its median. In other words this filter tries to remove false spot rows or columns which are in between the real rows and columns. After this process we have the set of distances $\{s^\alpha\}$ and $\{s^\beta\}$ with some erroneous elements removed and tentative spot center coordinates at the intersection of corresponding lines.

Grid placement. At this point we have a set of spot centers but there can still be erroneous centers due to noise. In this step we build a grid, based on the known number of spot columns and rows and the coordinates we already have, to find the best match of that generated grid to the image. We start by making a spot template, see Fig. 3, as a disc with radius r obtained earlier. We OM transform this disc since the input image for this step is the OM transformed grid image. Then we generate a grid with $sr \times sc$ deltas centered at the intersection of previously found lines. Since wrong line detection generates false spot centers, we could have detected more spot centers than the ones present in image. To select the correct spot centers we generate several grids of deltas with the known number of rows and columns based on the number of spot center coordinates we found previously. Then we convolve each delta grid and the spot template to obtain a grid template. Finally, we find the correlation of the template grids with the OM transformed grid images, and select the template grid that best matches the image. After the steps presented above we have a grid that best matches the input image, its parameters and the spots parameters (center and radius). Next, we present a segmentation method to refine the spot center coordinates.

2.2 Spot Segmentation Using Computational Gestalt Theory

Computational Gestalt Theory was first presented by Desolneux, Moisan and Morel [2] as a way to obtain a quantitative theory of the Gestalt laws. Computational Gestalt uses the *Helmholtz Principle* to define a quantitative measure of a given gestalt [2].

Helmholtz Principle. The observation of a given configuration of objects in an image is meaningful if the probability of its occurrence by chance is very small. The Helmholtz

principle can be formalized by the definition of the *Number of False Alarms* and ϵ-*meaningful events:*

Number of false alarms - NFA. The number of false alarms (NFA) of an event E is defined as: $\mathrm{NFA}(E) = \mathcal{N} \cdot \mathrm{P}[\mathcal{E} \geq E|H_1]$ where $\mathcal{N}$ is the number of possible configurations of the event E and H_1 is the background or *a contrario* model. An event E is ϵ-meaningful if the NFA is less than ϵ: $\mathrm{NFA}(E) < \epsilon$.

Spot Segmentation. Using the center coordinates and radius estimated before we apply a threshold based segmentation. The optimal threshold which separates the spot from the background is estimated with an NFA approach.

This method is applied to a small, square image centered on the spot of size $2r + 1$. Given a threshold, t, we compute the number of pixels, k_o, outside the spot with grey level above t. If N_o is the total number of pixels outside the spots and p_s is the probability of a pixel being above the threshold t[3] we can estimate the probability of at least k_o pixels above the threshold among N_o using the binomial distribution. Therefore, in this case the NFA is computed using the binomial tail as: $N_T \times B(p_s, n_o, k_o)$ where N_T is the number of thresholds tested. Additionally, with this procedure the NFA is a confidence measure which tells us if there is a spot or not in this position. Spots with $NFA > \epsilon$ are not considered in following steps. We iterate this procedure in a small region around the spot center given by the gridding step. We settle with the coordinates that give the best NFA figure. Therefore obtaining a better estimate of the spot center coordinates.

Data extraction. Our software also includes this step. Due to lack of space we do not present or evaluate this step here.

3 Results

In this section we show a comparison between the results obtained by our method and the program UCSF spot [4]. We used simulated microarray images with noise and distortions generated by *mamodel* [3].

In the *mamodel* website there are three parameter sets to generate different images. Their description read: “High quality slide”, “Noisy slide” and “Disturbing noise”. We chose the last one and made the necessary adjustments to the parameters for generating one grid of size 40x25 spots providing us a grid of 1000 spots.

As can be seen in Fig. 5 each image channel is generated with the same spot intensities and some distortions are present in both channels (scratches, air bubbles). However, stains can be present in one or both channels.

We begin our tests by obtaining the true positives (TP), that is, spots that should be found. mamodel provides in its output the noise free intensity values for each spot. In a posterior process, mamodel generates the noise contaminated slide image from those values. So, a reasonable approach would be to flag a spot as negative if its original intensity falls below the background noise level. For this matter, we took a region of the

[3] The probability p_s is empirically estimated based on the values of the pixels of a square region around the spot.

[4] www.jainlab.org

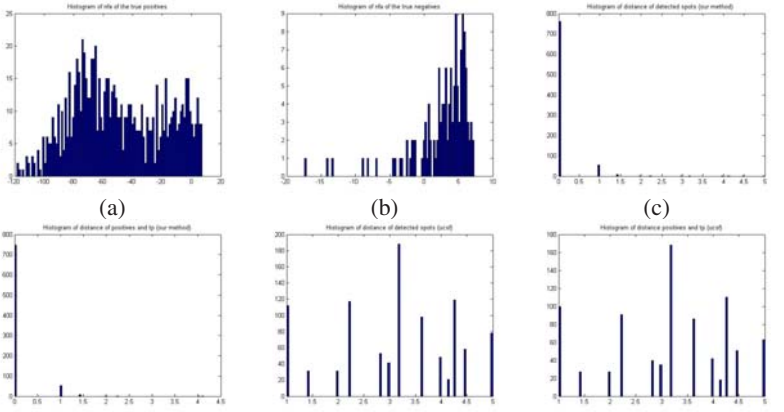


Fig. 4. (a) NFA of TP. (b) NFA of true negatives. (c-f) Histogram of distances from the reference center coordinates with: (c) Our method for TP, (d) Our method for all positives, (e) UCSF for positives, (f) UCSF for positives.

image containing only background noise and found its mean and obtain the true positives, as explained above. Although simple, this procedure has one obvious drawback: we are not taking into account any distortion, for example scratches, air bubbles, etc.

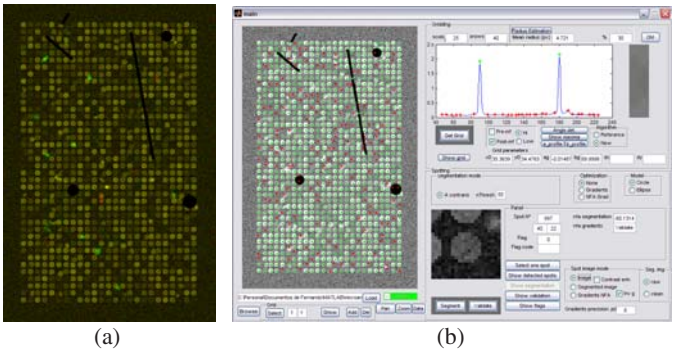
In Fig.4(a) and (b) we show the histogram of the NFA values for the TP and the true negatives (TN). Note that the NFA value can be used to flag a spot as found or missing as correctly discriminates between TP and TN. Also note that in Fig.4(a) there are still a significant amount of spots with NFA$>$0. Manual inspection of those spots confirmed that is caused by our imperfect way of flagging a spot as TP, without taking into account the distortions in the image as mentioned earlier.

If we consider a spot as positive if its NFA$<$0 we obtained the results in the following table with false positives (FP) and false negatives (FN). As we can see our method produces a more balanced pair sensitivity-specificity.

Value	Our method	UCSF Spot
Number of FN	50	5
Number of FP	22	137
Sensitivity	94.5%	99.4%
Specificity	86.2%	50%

Now we compare the accuracy on spot center detection for our method and UCSF. We take into account only the spots flagged as found by each program. We computed the distances to the reference spot center coordinates given by mamodel and plotted its histogram shown in Fig.4. As we can see for TP our method gives zero error for most of the spots while UCSF has errors ranging from 1 to 5. Regarding all detected spots (positives) our method gets some spots with larger errors but still the great majority has error cero as can be seen in the histograms. The statistics are presented in the following table.

Value	Our method (positives)	Our method (positives and TP)	UCSF Spot (positives)	UCSF Spot (positives and TP)
Mean	0.1267	0.0980	3.1488	3.1552
Median	0	0	3.1623	3.1623
Variance	0.2501	0.1456	1.3497	1.3554



(a) (b)

Fig. 5. (a) Simulated Image. (b) Screen of the developed software prototype showing the detected spots: green $NFA \leq 0$, red $NFA > 0$.

4 Conclusion

We developed a software prototype for the analysis of microarray images which includes the stages of gridding, segmentation and data extraction (not presented here). Starting from the method proposed in [2] we introduced several improvements to increase the accuracy. For the segmentation step we presented a method based on NFA which provides a confidence measure that can be used to flag spot and assist the user during manual inspection. We compared the results of our method with UCSF and outperformed it in sensitivity-specificity and spot center detection.

References

1. Ceccarelli, M., Antoniol, G.: A deformable grid-matching approach for microarray images. IEEE Transactions on Image Processing 15(10), 3178–3188 (2006)
2. Desolneux, A., Moisan, L., Morel, J.-M.: From Gestalt Theory to Image Analysis: A Probabilistic Approach. Springer, Heidelberg (2008)
3. Nykter, M., Aho, T., Ahdesmaki, M., Ruusuvuori, P., Lehmussola, A., Yli-Harja, O.: Simulation of microarray data with realistic characteristics. BMC Bioinformatics 7, 349 (2006)
4. Yang, Y.H., Buckley, M.J., Speed, T.P.: Analysis of cdna microarray images. Briefings in Bioinformatics 2(4), 341–349 (2001)