Chapter 05

Automatic Spot Identification for High Throughput Microarray Analysis

Eunice Wu¹, Yan A Su², Eric Billings³, Bernard R Brooks³ and Xiongwu Wu³*

¹Thomas Jefferson High School for Science and Technology, USA
²GenProMarkers, Inc., USA
³Laboratory of Computational Biology, USA

*Corresponding Author: Xiongwu Wu, Laboratory of Computational Biology, 5635 Fishers Lane, T804, Bethesda, MD 20892, USA, Tel: 301-451-6251; Fax: 301-480-6496; Email: wuxw@nhlbi.nih.gov

First Published June 25, 2018


Copyright: © 2018 Xiongwu Wu, et al.

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source.
Abstract

High throughput microarray analysis has great potential in scientific research, disease diagnosis, and drug screening. The main problem that many microarray experiments must deal with is the time and effort spent accurately locating spots. An automatic microarray image processor will allow accurate and efficient determination of spot locations and sizes so that gene expression information can be reliably extracted in a high throughput manner. Current microarray image processing tools require manual effort in addition to the pre-processing by the program to correctly and accurately identify spots. This paper presents a method, herein called auto-spot, to automate the spot identification process. Through a series of correlation and convolution operations, as well as pixel manipulations, this method makes spot identification an automatic and accurate process. Testing with real microarray images has demonstrated that this method is capable of automatically extracting subgrids from microarray images and determining spot locations and sizes within each subgrid, regardless of variations in array patterns and background noise.

Introduction

Microarray analysis is a widely-used technology in genetic-related fields and has been applied in areas ranging from cancer research to pest control. Now scientists can design special microarrays for various purposes, such as cancer research [1], mitochondrial function [2], chromosomal abnormalities [3], and artery diseases [4]. As microarray technology progressed, high-throughput microarray analysis has become a powerful approach for scientific research and disease diagnosis [5]. For example, high-throughput microarray analysis has been developed and used in identifying molecular targets of brain disorders [6], drug discovery, toxicology, stem cell research [7], cancer research [8], molecular diagnosis [9], functional proteomics [10], and biological system analysis [11].
Many improvements have been devoted to achieving high-throughput microarray analysis, including the areas of sequencing, third dimension, experimental protocol, image capture, and image processing. Among these areas of improvement, microarray image processing is a key step for a successful high-throughput microarray analysis. This step determines the quality of microarray data, which is fundamental to all later analysis such as gene clustering and pathway derivation.

Scientists use microarrays to study gene expression levels and to sequence genomes. By putting DNA in an array on the microarray chip, an orderly formation of spots is created. Each spot contains thousands of identical DNA molecules, consisting of DNA, cDNA or oligonucleotides. Microarray image processing is performed to obtain the intensity information of these spots. Figure 1 shows typical difficulties in microarray image processing. These include: 1) spot misalignment, where spots are offset from their intended location; 2) image rotation, which is unwanted rotation of microarray grids; 3) unobservable spots, where large ranges of intensities make some spots hard to detect; 4) uneven background due to noise; 5) contamination on microarray slides; 6) high spot density; and 7) irregular spot shapes and sizes. In addition, microarrays from different vendors have different formats, often requiring special information from vendors to perform a successful image analysis.
Even with microarray format information from vendors, much human input is needed to accurately determine the locations and sizes of many spots in a microarray image. At present, image analysis software for microarray analysis often fails to accurately locate gene spots without manual adjusting. The circles enclosing spots may be too small, resulting in the loss of data. The circles could be too large and enclose more background and noise, resulting in larger errors. The circles could also be completely off of the spot, distracted by unwanted splotches on the chip. Consequently, human input is needed to correct these inaccuracies in spot detection and measurement. This takes up many hours of repetitive and tedious adjustments, making microarray technology, which is meant to expedite scientific investigations, much less efficient.

There are existing software tools available for microarray image analysis such as ScanAlyze (by M.B. Eisen, http://www.eisenlab.org/eisen/) and TIGR Spotfinder [12] (www.tigr.org/software/tm4/). However, these rely on human input to accurately identify spots and measure spot intensities.
There have been numerous attempts since the emergence of microarray technology at automating image processing procedures [13-32]. However, these methods usually rely on either specific array formatting information or human input, and are not available in most commonly used microarray analysis software.

The most recent and up to date work for automatic image processing is by Rueda and Rezaeian [32]. They use radon transform to determine rotation and use horizontal and vertical histograms to identify rows and columns. This discrete approach may be inaccurate in determining the rotation angle and does not address non-rectangular spot arrays.

The purpose of the auto-spot method presented herein is to automate the process of analyzing microarray images. The method needs to be able to locate subgrids within a microarray image and to identify spot regions regardless of array patterns, rotational angles, and margin information.

**Method**

Due to a variety of production sources, microarray chips are produced with different densities, different spot sizes, and different arrangement patterns. The spot intensities reflect quantity levels and have a wide range of values. Further complicating the analysis, a microarray image is often misaligned and rotated to a varying degree. Also, background due to noise and contamination can directly affect the accuracy of spot detection and measurement. An automatic processing method for microarray analysis needs to be able to separate subgrids in a microarray image from one another and be able to identify spots within each subgrid image.

This work presents a method herein called auto-spot to automatically process microarray images. Figure 2 shows the image processing procedures of this method. The auto-spot method processes a microarray image using two main procedures. First, it extracts a subgrid from an image; and second, it identifies spots within each subgrid image. Figure 3 shows a flow chart depicting steps for ex-
tracting subgrids, and Figure 4 shows a flow chart of the steps for identifying spots. In the following subsections, these two procedures are described separately.

**Figure 2:** The microarray image processing procedure of the auto-spot method.

**Figure 3:** Flow chart for subgrid extraction.
A microarray image is arranged to have subgrids in rows and columns. However, where the rows and columns are, and how much the image has been rotated, need be decided before subgrids can be separated from one another. The auto-spot method locates subgrids by determining the separation lines that separate subgrids into rows and columns. The subgrids are then extracted as the enclosed region of these separation lines. The steps in this subgrid location process are explained in detail below.
1. Calculate the auto-correlation image of a microarray image (Figure 5).

**Figure 5**: Calculate subgrid auto-correlation map of microarray images.

First, estimate spot distance, $d$, from the peaks in the direct auto-correlation map of the microarray image. The microarray image is filtered with a bandpass filter to remove structure features of the size $d$ or less. For the convenience of correlation function calculations, the microarray image may be resized to a square with dimensions of a power of two. The subgrid auto-correlation map of a microarray image contains the arrangement information, mainly the slopes of the border lines separating the subgrids, while suppressing many irregular effects such as noise and contamination.
2. Construct strip images of rows and columns (Figure 6)

![Slopes of maximums](image1)
![Row strip image](image2)
![Column strip image](image3)

**Figure 6:** Slopes of the rows and columns calculated from the maximums and the row strip image and the column strip image constructed from the slopes.

The slopes of the rows and columns can be calculated from the maximums in the auto-correlation image obtained above. Identify the near vertical maximums from above and below the origin and the near horizontal maximums from the right and left of the origin. The average distance between the near horizontal maximums is $\Delta w$ and the average distance between the near vertical maximums is $w$. The slopes are calculated in the following way:

\[
\begin{align*}
    k_w &= \frac{1}{n_w} \sum_{i=1}^{n_w} \frac{y_i}{x_i} \\
    k_h &= \frac{1}{n_h} \sum_{i=1}^{n_h} \frac{x_i}{y_i}
\end{align*}
\]
Here, \( n_w, n_h \) are the numbers of maximums along the row and column separation lines. The variables, \( k_w, k_h \), are the slopes of the row and column separation lines. Using the slopes, a row strip image and a column strip image can be created as shown in Figure 6. These strip images are images of the same size as the original image. Their pixel values are one in a strip region and zero otherwise. The strip has a defined slope and width. The width of the strip is set to \( \Delta w \) and \( \Delta h \) for the column strip and the row strip, respectively. The purpose of these strip images is to enhance the contrast between spot regions and non-spot regions through convolution, as explained in the next step.

3. Locate separation lines from the strip convolution images (Figure 7 and Figure 8)

**Figure 7:** Convolution with correlation-derived strips to enhance the contrast of separations between subgrids
Convolve the low-bandpass filtered image obtained in step 1 with the row and column strip images obtained in step 2 to make the separation region stand out (Figure 7). The strip convolution makes spot region overlap except the separation between subgrids, so that the separation between subgrids have minimum intensities, which are apparent in the intensity sums along the row or column slopes (Figure 8). By identifying the minimums, the equations of separation lines, eqs. (2a) and (2b) can be defined for each separation.

\[
y_j = k_w x + b_{wj} \tag{2a}
\]

\[
x_i = k_h y + b_{hi} \tag{2b}
\]
where \( b_{ji}, b_{wi} \) define the location of the separation lines. Between the separation lines are the subgrids to be extracted for spot identification.

4. Extract subgrids from the region enclosed by the separation lines (Figure 9).

Figure 9: Identified separation lines in a microarray image. Subgrids can be extracted from the regions enclosed by the separation lines.

Separation lines are scaled back to the original microarray image size to separate subgrids. Subgrids between these separation lines are extracted from the image for spot identification (Figure 9).

**Spot Identification**

Once subgrids are extracted, spots are to be identified within subgrids. The flow chart of the spot identification procedure has been shown in Figure 4. The concept is that within a subgrid, spots are arranged in a single pattern. Spot locations match the pattern everywhere, regardless of actual spot intensity. Therefore, spot locations can be determined once the pattern is known. Then spot intensities
are used to identify boundaries of the spot array. Within the boundary, spot sizes and locations are refined with pixel values. By following the pattern, no spot will be missed within the boundary. The details of the spot identification procedure are described as the following steps.

1. Array pattern extraction (Figure 10)

![Subgrid](image1) ![Auto-correlation map](image2) ![Decaying lattice map](image3)

**Figure 10**: Processing a subgrid image to obtain grid lattice map. The grid lattice map describes the spot distribution pattern.

For convenience, each subgrid is resized to a square with dimensions that are powers of two. Calculate the auto-correlation image and find all maximums in the correlation image (Figure 10). The auto-correlation image reflects the array pattern but has inconvenient maximum values. The center peak is extremely high and peaks away from origin are blurring due to noises and printing distortion. Construct a lattice image based on the locations of the maximums but decay the peak heights exponentially from the origin. The decay function of the following form is used to reduce effects of long-range distortion:

\[
p = e^{-\frac{x^2 + y^2}{r_{\text{decay}}^2}} \quad (3)
\]

The decay distance is set to a fraction of the subgrid size, \( r_{\text{decay}} = \frac{w+h}{8} \).
2. Spot location mapping (Figure 11)

![Diagram of subgrid, grid lattice map, and spot location map]

**Figure 11**: Convolve the subgrid with the lattice map to obtain the spot location map.

The subgrid image is convolved with the lattice image from step 1 to obtain a spot location map that shows the spot positions throughout the map (Figure 11). A convolution with the lattice image enhances the intensities of positions that are on the lattice pattern. All positions belong to the lattice pattern are maximums in the spot location map and are identified by finding maximums.

3. Boundary determination (Figure 12)

![Diagram of subgrid, spot location map, and data map]

**Figure 12**: Determine the boundary rows and columns based on intensity sums in rows and columns. The subgrid image within the boundary is the data map.

The intensities at the spot locations are sum up in rows and columns. The sums within the boundary are significantly higher than the sums outside the boundaries. Therefore, the boundary locations can be defined at the place where the sum increases the most. The spots within the boundaries are data spots (Figure 12).
4. Location refinement and size determination (Figure 13)

![Data map](image1) ![Spot identification map](image2)

**Figure 13:** Spot location refinement and size determination.

For each data spot, calculate the center of intensity around the initial location by the following equations,

\[
\begin{align*}
    x_{c} &= \frac{\sum_{i} x_{i} I_{i}}{\sum_{i} I_{i}}, \\
    y_{c} &= \frac{\sum_{i} y_{i} I_{i}}{\sum_{i} I_{i}},
\end{align*}
\]

The spot size can be fixed to be a certain percentage (e.g., 80%) of the grid size, which is defined by the distance to the neighboring spots. It can also be variable and determined by the number of pixels with intensity above the mean intensity,

\[
r_{s} = 1 + \sqrt{\frac{N(I \geq \bar{I})}{\pi}}.
\]

Here, \(N(I, \bar{I})\) is the number of pixels that have intensity higher than average intensity of the spot region, where \(r_{g}\) is the radius in pixels. The values of \(x_{g}, y_{g}\), and \(r_{g}\), represent the location and size of the spot to be identified.

Once the locations and sizes of spots are determined, it is straightforward to extract the intensity information of the spots. As to the processing of intensities and related them to gene functions are the topics of many studies and are not discussed here.

**Program Design**

The auto-spot method is developed based on a popular image processing program: ImageJ [33,34]. ImageJ provides many image processing related functions, such as band pass filter, correlation, convolution, access and manipulation of pixel values, etc., so this devel-
opment can focus on concepts and ideas. ImageJ also provides macro script writing and debugging utility that makes it very convenient for programming and testing. It is planned to implement the auto-spot method to program R (http://www.r-project.org/) so this method can be accessed by many people in the microarray analysis community.

Result and Discussion

The auto-spot method presented in this work is for high throughput microarray image processing. This method’s capability of automatically gridding can be demonstrated using real microarray images. For objective tests and demonstrations, it is recommended to use publicly available microarray images to examine the method. There are many such microarray images available from the microarray community. Here, the microarray images from the Stanford microarray database (SMD) at www.smd.stanford.edu were downloaded and used to examine this method and to compare it with other methods. Due to space limitations, only two images with different microarray spot patterns are examined in this report. One has a rectangular array pattern, and the other has a hexagonal array pattern.

As a comparison, the results from commonly used microarray image processing software tools, ScanAlyze version 2.51 (by M.B. Eisen, http://www.eisenlab.org/eisen/) and TIGR spotfinder version 3.2.1 [12], are presented. Both of the programs are freely available and well documented. Due to space limitations and availability, as well as cost, other software tools are not compared. Further comparisons with other software tools will be reported in future studies. ScanAlyze and TIGR spotfinder are valuable software tools and have been used in many microarray studies. They have many more functions in addition to gridding microarray images. In the comparison here, the only focus is on spot identification.

Global Transcriptional Profiling Microarray (SMD ID:20385)

This image represents a typical microarray spot layout with a rectangular array pattern. We chose this image is because it contains
most features of microarray images, such as tilted layout, contamination, high noises, etc.

Figure 14 shows the image and the results of subgridding and spot identification with the auto-spot method. Panel (a) shows the SMD 20385 image, which tilts toward its right, is noisy, and has contaminations. Panel (b) shows the separation lines obtained from the auto-spot method. It is clear that the separation lines are positioned perfectly to separate rows and columns. The separation lines are tilted the same way as the subgrid layout to correctly enclose subgrids. Panel (c) shows a subgrid extracted from the region enclosed by the separation lines. Panel (d) shows the result of spot identification. The spots are marked by circles, whose sizes correspond to the spot sizes. For clarity, an enlarged portion of the spot identification is shown in panel (e). The locations and sizes of the circles agree well with the spots in the image. Some weak spots has large circles, because spot size is determined by the distribution of intense pixels, not by the intensities.

Figure 14: (a) Microarray image of SMD20385 channel 1 downloaded from smd.stanford.edu. This image contains 4X12 subgrids. Each subgrid contains spots in a rectangular pattern. (b) The separation lines obtained with the auto-spot method are shown as white lines. (c) A subgrid extracted with the separation lines. (d) The spots identification result. White circles represent the locations and sizes of the spots. (e) An enlarged portion of the spot identification result.
As a comparison, we show the results from ScanAlyze and Spotfinder. Figure 15 shows the result of ScanAlyze. To use ScanAlyze to grid a microarray image, one need to create a new grid if an existing grid is not available. To create a new grid, one needs to input a series of microarray parameters, such as number of rows and columns, spot resolution in x and y direction, spot width and height, pin spaces in x and y directions, and the number of subgrids to generate. Table I lists the microarray parameters used to create initial grids.

Figure 15(a) shows a generated grid with parameters shown in Table I. Only 32 subgrids are created because that is the maximum number of subgrids allowed by ScanAlyze. Initial grids often mismatch the spots in microarray images due to a variation in actual parameters, printing errors, as well as image tilting. Figure 15(b) shows an enlarged initial grid, from which one can clearly see the significant mismatches between grid circles and the spots. The mismatch can be adjusted manually with mouse by dragging the grid to the location and size of actual subgrids. The match can be further improved by pressing repeatedly a refine button. Figure 15(c) shows the final gridding result. An enlarged portion is shown in Figure 15(d). As can be seen the circles match the spots very well.

Figure 15: Gridding result with ScanAlyze for SMD20385. (a) 32 generated subgrids with input parameters listed in Table I. (b) An enlarged view of the generated subgrid. (c) After resizing, moving, tilting, and refining, the grids match well with the spots. (d) An enlarged region of panel (c).