A Biomimetic Algorithm for the Improved Detection of Microarray Features

Dan V. Nicolau, Jr[1], Dan V. Nicolau[2], Philip K. Maini[1]

[1] Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Oxford, OX1 3LB, UK

[2] Department of Electrical and Electronic Engineering, Liverpool University, Brownlow Hill L69 3GJ, Liverpool, UK

ABSTRACT

One the major difficulties of microarray technology relate to the processing of large and – importantly – error-loaded images of the dots on the chip surface. Whatever the source of these errors, those obtained in the first stage of data acquisition – segmentation – are passed down to the subsequent processes, with deleterious results. As it has been demonstrated recently that biological systems have evolved algorithms that are mathematically efficient, this contribution attempts to test an algorithm that mimics a bacterial-"patented" algorithm for the search of available space and nutrients to find, "zero-in" and eventually delimitate the features existent on the microarray surface.

1. INTRODUCTION

The introduction of microarray technology in the late 1990s has been the workhorse of genomics revolution and incoming proteomics and cellomics revolutions as well as being relevant to other high throughput screening techniques. Microarray technology is based on the large parallel testing of biomolecular interactions of pair biomolecules, of which one (probe biomolecule) is immobilized on the surface of the microarray chip and its pair (target biomolecule) is presented to the microarray surface from a solution. The deposition of the probe biomolecules on the microarray surface can be achieved by a variety of methods, either based on direct mechanical contact (as in the initial microarray technology), or thorough more evolved technologies derived from ink-jet printing. Whatever the case, the spots produced on the microarray surface have approximate circular shapes with uneven distribution of the concentration of the probe biomolecules inside the spot. The microarrays are then read through the biomolecular-specific recognition of the regions where probe biomolecules are immobilized and subsequent attachment of the target biomolecules. The geometrical imperfections of the dots comprising probes can be -at times- greatly amplified by this process, which has its own imperfections, e.g., non-specific attachment. Furthermore, uncertainties regarding the relationship between the intensity and the emission wavelength of the fluorescence –the classical method for readout- versus the concentration of the target biomolecules can also conspire towards the deterioration of the images of the features on the microarray surface.

The interpretation of the microarray data starts with the integration of the signal compared with the background on the area of individual dots – segmentation-, the results being further used for elaborate clustering methods. It follows that the incorrect demarcation of the circular dot features will propagate throughout the whole microarray data processing and will add to the other sources of variability of microarray data, e.g., biological variability, technical variability and labeling [1]. Several methods have been proposed to denoise data [2], adaptative split and merge algorithm [3], polynomial-hyperbolic spot shape model in combination with the Box–Cox transformation [4], spectral embedding [5], noise-resistant algorithms [6], background extraction [7], just to name few recent contributions.

It has been demonstrated recently that biological systems have evolved algorithms that are mathematically efficient, inter alia, for the exploration of available space [8] and finding nutrients [9]. This contribution attempts to test

Imaging, Manipulation, and Analysis of Biomolecules, Cells, and Tissues V, edited by Daniel L. Farkas, Robert C. Leif, Dan V. Nicolau, Proc. of SPIE Vol. 6441, 64411C, (2007) · 1605-7422/07/$18 · doi: 10.1117/12.725769

an algorithm that mimics a bacterial-"patented" algorithm for the search of available space and nutrients to find, "zero-in" and eventually delimitate the features existent on the features on the microarray surface.

2. METHODS

We begin by describing the swimming behavior of Escherichia coli. E coli is a single-celled organism that lives in the gut. Each bacterium possesses a set of around 6 rotary motors each roughly 45 nm wide. Each motor drives a thin, helical filament that extends several cell body lengths out from the bacterium; the motor-filament ensemble is called a flagellum. E coli uses the motion of the flagella in concert to swim towards favorable areas (for example, regions of high nutrient concentration). If all the flagella rotate counterclockwise, they form a spinning bundle and the cell swims in a roughly straight line (the trajectory is perturbed slightly by Brownian forces) – a "run". If some or all of the flagella spin clockwise, the bundle flies apart and the cell turns in place in a random fashion – a "tumble". By combining these two modes, the cell can alternate between swimming towards favorable areas and randomizing its new direction of travel, thus staying in these regions or keeping away from other, less favorable areas (Berg, 2000).

The cell can find areas of high nutrient concentration by biasing its random walk. The biochemical network responsible for chemotaxis compares the last second of the bacterium's life with the previous 3 seconds (Berg, 2000). If conditions have improved then the mean run length is increased. If, to the contrary, conditions have become less favorable, then the mean run length is decreased. Both the run and tumble times are exponentially distributed. In a gradient-free medium, the mean run time is 1.0 s and the mean tumble time is 0.1 s. Examples of typical E. coli trajectories in such a medium are shown in Figure 1.

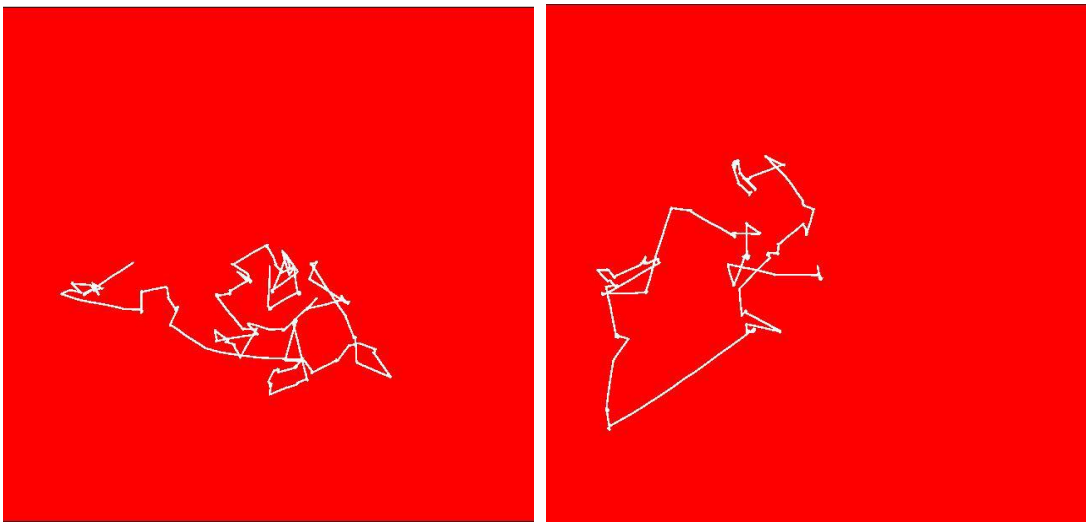

Figure 1. Typical E coli trajectories in a gradient-free medium.

In order to mimic the swimming behavior of E coli, we implemented the following Monte Carlo algorithm. The medium in which the organisms evolve in our simulations is a two-dimensional lattice (representing a microarray image) of dimensions $n \times n$ voxels. Initially, a fixed number of model organisms is randomly distributed on the lattice. An attractant $\chi(x,y)$ is distributed on this surface, such that the maximum concentration is found in the centre of the simulation area and the minimum at the edges.

Besides its position and direction vector θ_i, each bacterium i is endowed with the following properties: (1) a state – either "swimming" or "turning", (2) a simple comparative memory function M defined as

$$M(\tau) = \begin{cases} 1/w_r, & 0 < \tau \le w_r \\ -1/w_d, & w_r < \tau \le w_d \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where w_r and w_d are called the recent and distant memory windows, respectively and (3) two response functions, S and T (described below).

The algorithm visits each organism. We calculate the comparison function $A(t)$ as follows:

$$A(t) = \sum_{\tau=0}^{\min(w_r+w_d,t)} C(t-\tau)M(\tau) \tag{2}$$

where $C(t)$ is the concentration of attractant seen by the organism at time t, i.e. the difference is calculated between the average attractant seen over the last w_r steps and that seen over the preceding w_d steps (a simple comparison of the recent and distant past). The algorithm visits each organism in turn, incrementing its position and direction, updating its state, computing the comparison function A and recording the amount of attractant seen at each step. After every organism is visited, the simulation time is incremented by 1 and the cycle is repeated. The position and state of the organism are changed at each step according to the following rules:

- If the organism is in the "swimming" state:
  - Its direction θ_i is perturbed by ηD_{rot}, where η is a normally distributed random number with mean 0 and standard deviation 1 and D_{rot} is a rotational diffusion coefficient, known to be 0.15 rad^2/sec for *E. coli* (Strong et al., 1998).
  - Its position is incremented by a vector of length v (set to 1 in our simulations) pointing in the current direction.
  - A random number r between 0 and 1 is generated and compared with the swimming function $S(A(t))$, described below. If $r<S(A(t))$ then the organism stays in the "swimming" state, otherwise it switches to the "turning" state.
- If the organism is in the "turning" state:
  - Its direction is perturbed by $\eta(D_{rot} + D_{turn})$ similarly to above, where D_{turn} is the diffusion coefficient of "turning" and is set to $\pi/2$ in our simulations.
  - A random number r between 0 and 1 is generated and compared with the turning function $T(A(t))$, described below. If $r<T(A(t)$ then the organism stays in the "turning" state, otherwise it switches to the "swimming" state.

We now describe the meaning of each of the functions. Informally, $A(t)$ is a comparison of the recent and distant past using the memory function M. Note that in order to calculate $A(t)$, the organism only uses, at each step, the current concentration of attractant. The functions S and T are then used by the organism to probabilistically decide, based on $A(t)$, whether to remain in the current state (if swimming, using S; if turning, using T), or to switch to the other state: a Markov chain model. S and T could in principle be continuous in $A(t)$, so that the probability of remaining in the current state depends continuously on the comparison of recent and distant past. However, *in silico*, we discretise these functions so that the $A(t)$ space is partitioned into m bins each of size $2A_{max}/m$ where A_{max} is the maximum value of $A(t)$ we expect (which can be estimated using the attractant distribution). We find, at each step, the appropriate bin j such that

$$-A_{\max} + \frac{2(j-1)A_{\max}}{m} \le j < -A_{\max} + \frac{2jA_{\max}}{m}. \tag{3}$$

S and T can then be determined using a lookup table. For simplicity we restrict ourselves to the $m = 2$ case, in which the organism can only tell whether life is getting 'better' (positive $A(t)$, j=1) or 'worse' (negative A(t), j=2) but cannot quantify the improvement or deterioration. Larger values m would correspond to a higher resolution and allow the organism to employ increasingly sophisticated strategies to find regions of high attractant concentration (in all cases, $j=m/2$ is the point between life getting better and worse).

Thinking about the intensity of an image as different levels of nutrient concentration in space allows us to use this algorithm to identify regions of high intensity. Clearly, the form of $\chi(x,y)$ will affect the motion of the model bacteria. We initially suppose for simplicity that the intensity of a spot can be well described by a Gaussian function.

To locate the centre of a spot that is located randomly inside the simulation area, we can compute, at periodic intervals, the "centre of mass" of the model bacteria as follows:

$$c_x = \frac{\sum_{i=1}^{n} x_i}{n}$$
$$c_x = \frac{\sum_{i=1}^{n} y_i}{n} \tag{4}$$

where x_i and y_i are the x and y coordinates of bacterium i. As bacteria aggregate around the intensity centre of the image (the centre of the spot), the point (c_x,c_y) approaches the true centre of the intensity spot. We can also get a measure of the dimensions of the spot by computing the standard deviation of the bacterial positions:

$$\sigma_{spot} = \frac{\sqrt{\sum_{i=1}^{n}(x_i - c_x)^2 + \sum_{i=1}^{n}(y_i - c_y)^2}}{n}. \tag{5}$$

3. RESULTS

We first tried our method on detecting the centre and size of a single spot located somewhere at random inside a rectangular area. We let the spot have a Gaussian intensity distribution, so that the intensity of the image is given by

$$\chi(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \tag{6}$$

where (x_0, y_0) is the centre of the spot and σ is the standard deviation of the Gaussian and is the characteristic spot size. We seeded the lattice with 200 model bacteria, initially randomly located and heading in randomly chosen directions. Figure 2 shows the computed centre of the image (using Eq. 4) and the dimensions of the spot, determined using Eq. 5.

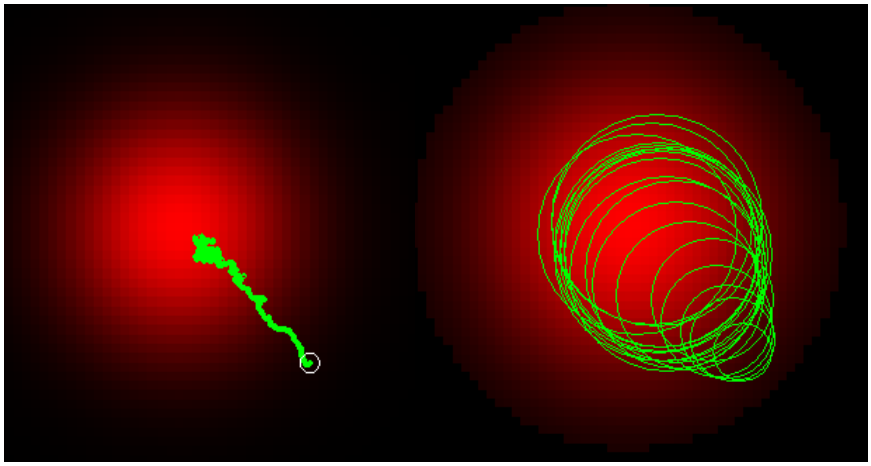

Figure 2. Determining the centre and width of a spot using model bacteria. Regions of high intensity are shown in red. Left: iterates of the spot centre, computed using Eq 4. Right: iterates of the spot size, computed using Equation 5.

Gaussian intensity distributions leads to good performance of the algorithm, possibly because in the natural habitat of bacteria, the nutrient is dispersed through diffusive processes, which are typically characterised by Gaussian or Gaussian-like distributions (since the fundamental solution of Fick's equation takes this form). However, this form may not be realistic for a microarray spot since, for example, we expect the edges of the spot to be much sharper than the long tail of a Gaussian distribution. We can simulate this, however, by truncating the Gaussians below a given threshold of intensity θ so that for all values of $\chi(x,y)<\theta$ we simply set $\chi=0$. Equation 4 can then be modified so that only the model bacteria located on voxels whose intensity is non-zero will be taken into account. Equation 5 can be similarly modified.

Furthermore, microarray images consist of many spots and the algorithm must deal with this additional constraint. Suppose for simplicity that the microarray is already segmented into square areas, each of which may or may not contain a spot – segmentation algorithms for this problem exist and perform well [3, 6]. We tested the bacterial algorithm using these two additional constraints. Typical results are shown in Figure 3 on a model 2×2 microarray with different (randomly generated) spot sizes, located at random points inside each array location.

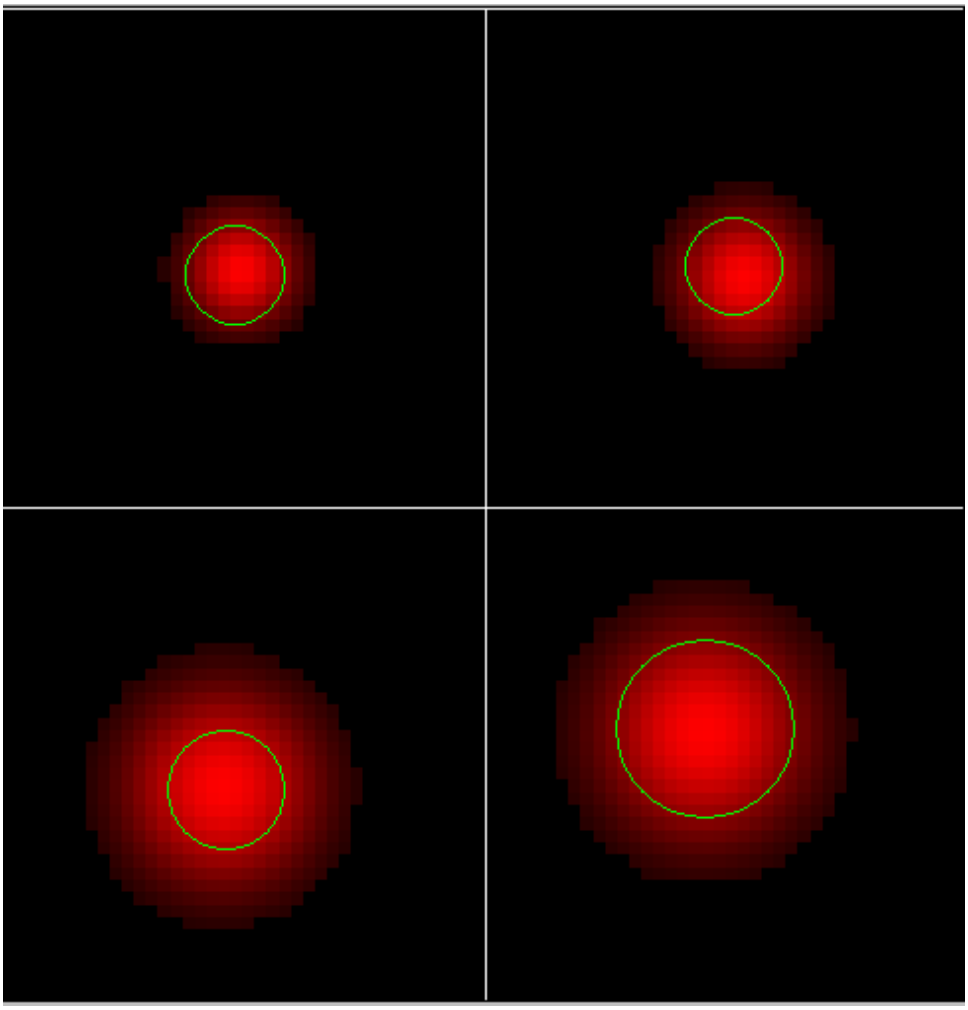

Figure 3. Spot sizes and centres on a 2×2 model microarray located using the bacterial algorithm. Each spot is a truncated Gaussian with $\theta=0.15$ (see text).

Finally, we measured the performance of our algorithm on a real microarray image chosen from [11]. Figure 4 shows the determined spot sizes and centres.

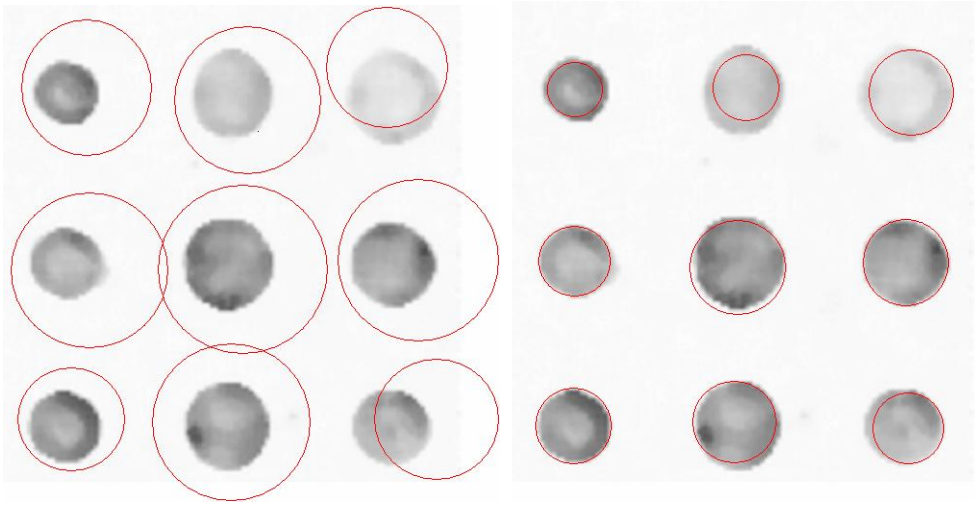

Figure 4. Performance of the algorithm on a real 3×3 microarray image after 100 (left) and 1000 iterations (right) (determined spot areas shown by solid circles).

4. DISCUSSION AND FUTURE WORK

Importantly, the algorithm presented here has potential even if the image is not segmented, as we have assumed it to be. In this more general case, the model bacteria would still converge on the spots present in the image, but it would not be possible to identify the number of spots or set bounds on their locations *a priori*. To solve this problem, one would need to detect when a cluster of bacteria has formed, this in turn requiring an algorithm dedicated to this task. One possibility is to mimic "quorum sensing", a phenomenon whereby bacteria not only consume but also release a chemoattractant into the environment. Other bacteria are then attracted to this region and in such a way a stable cluster is formed. It would then be possible to identify a cluster formed by model bacteria when the mean "chemoattractant" released on a set of voxels exceeds a known threshold. This will form the subject of future work.

One key step is to compare this algorithm with more classical approaches such as those presented in [2, 4, 5] and other contributions. We have not done this here and it will also form the subject of future work. However, we note that in general, biomimetic algorithms are expected to be useful in areas where classical algorithms encounter difficulties, such as image analysis, object identification and pattern recognition and therefore, that there is some evidence to suggest that bacterial-inspired algorithms could be useful in image analysis in general and in microarray analysis in particular.

5. ACKNOWLEDGEMENTS

The authors would like to thank Alex Oshmyansky for useful discussions. D.V.N. Jr is supported by the Clarendon Scholarship, the United Kingdom ORS Award Scheme and the Devorguilla Scholarship at Balliol College, Oxford University.

6. REFERENCES

[1] Zakharkin, S.O., Kim, K., Mehta, T., Chen, L., Barnes, S., Scheirer, K.E., Parrish, R.S., Allison, D.B., Page, G.P. Sources of variation in Affymetrix microarray experiments. *BMC Bioinformatics*, 6:214-225, 2005.

[2] Adjeroh, D.A., Zhang, Y., Parthe, R. On denoising and compression ofDNAmicroarray images. *Pattern Recognition* 39:2478–2493, 2006.

[3] Barra, V. Robust segmentation and analysis of DNA microarray spots using an adaptative split and merge algorithm. *Computer methods and programs in biomedicine*, 8:174–180, 2006.
[4] Ekstrøm, C.T., Bak, S., Kristensen, C., Rudemo, M. Spot shape modelling and data transformations for microarrays. *Bioinformatics*, 20:2270–2278, 2004.
[5] Higgs, B.W., Jennifer Weller, J., Solka, J.L. Spectral embedding finds meaningful (relevant) structure in image and microarray data. *BMC Bioinformatics*, 7:74-87, 2006.
[6] Novikov, E. Barillot, E. A noise-resistant algorithm for grid finding in microarray image analysis. *Machine Vision and Applications*. 17:337–345, 2006.
[7] O'Neill, P., Magoulas, G.D. Improved processing of microarray data using image reconstruction techniques *IEEE Transtactions on Nanobioscience*, 2, 176-183, 2003.
[8] Hanson, K.L., Nicolau, D.V., Jr, Filipponi, F., Wang, L., Lee, A.P. Nicolau, D.V. Fungi use efficient algorithms for the exploration of microfluidics networks. *Small*, 2, 1212-1220, 2006.
[9] Nakagaki, T., Yamada, H., Tóth, Á. Maze solving by an amoeboid organism, *Nature*, 407:470, 2000.
[10] Berg, H.C., Motive behavior of bacteria. *Physics Today*, 53(1):24–29, 2000.
[11] Rhodes, P., Enhancement of DNA and Microarray Analysis using Image Processing Practices (presentation), 2005.