O'Neill, Paul; Magoulas, George and Liu, Xiaohui (2003) Improved processing of microarray data using image reconstruction techniques. *IEEE Transactions on Nanobioscience* **2** (4) 176-183.

Improved Processing of Microarray Data Using Image Reconstruction Techniques

Paul O'Neill, George D. Magoulas, *Member, IEEE*, and Xiaohui Liu*

***Abstract*—Spotted cDNA microarray data analysis suffers from various problems such as noise from a variety of sources, missing data, inconsistency, and, of course, the presence of outliers. This paper introduces a new method that dramatically reduces the noise when processing the original image data. The proposed approach recreates the microarray slide image, as it would have been with all the genes removed. By subtracting this background recreation from the original, the gene ratios can be calculated with more precision and less influence from outliers and other artifacts that would normally make the analysis of this data more difficult. The new technique is also beneficial, as it does not rely on the accurate fitting of a region to each gene, with its only requirement being an approximate coordinate. In experiments conducted, the new method was tested against one of the mainstream methods of processing spotted microarray images. Our method is shown to produce much less variation in gene measurements. This evidence is supported by clustering results that show a marked improvement in accuracy.**

***Index Terms*—Data quality, image, microarray, preprocessing, reconstruction.**

I. Introduction

MICROARRAY technology is progressing at an amazing rate; the number of genes that can be processed is ever increasing, along with a multitude of techniques that can be applied to analyze the various stages of data [16], [17]. No matter how carefully a microarray experiment is conducted, it is always certain that there will be sources of external errors within the slide. These errors can come in many forms, such as technical errors like the random variation in the scanning laser intensity, inaccurate measurement of gene expressions, and a wide range of artifacts such as hair, dust, or even fingerprints on the slide. Another main source of errors is more biological in nature, with possible contamination of the cDNA solution and inconsistent hybridization. If this did not cause enough problems, the technology is still very expensive, and so this often means that measurements cannot be repeated or they are at most duplicated. This is not an ideal situation for data analysis; however, it does emphasize that as much work as possible should go into the preprocessing of the data to obtain the best possible results for further analysis.

Currently, one of the mainstream uses of gene expression data is that of clustering [7]. Generally, this utilizes methods such as hierarchical clustering to group or order the genes based on their similarity or dissimilarity, an example of which can be seen in the work by Gasch *et al.* [8]. Based on this, biologists can make suppositions about the functions of previously unknown genes or possibly study the interactions between genes, such as the discovery of gene pathways. However, all these forms of analysis are very susceptible to noise, highlighting again the need to make sure that the preprocessing of the data is as reliable as possible. Although many people have focused on the postprocessing of the microarray data [4], [7], comparatively there has still been very little work done looking into improved methods of analyzing the original image data [14], [19].

In this paper, we will propose a novel method for analyzing the image data that inherently deals with slide variation and a large degree of slide artifacts, and then contrast this to GenePix Pro [3], a software package that is often used for analysis. We contrast them through a series of experiments designed to look at variation within genes, between repeated genes, and between clustering results. These tests will be conducted on a large dataset that contains a high number of repeated genes. The next section takes a closer look at methods used in processing microarrays and possible sources of error. Then, in Section III, we describe the new methods of processing microarray image data and give examples of its use. Section IV presents the real-world dataset that was used during this paper, along with various metrics and algorithms used throughout the testing process. Following this is a series of experiments designed to compare the two techniques. Finally, there is a summary of our findings and a brief look at how this work can be extended.

II. Existing Techniques

Although there are many varied techniques for analyzing microarray data, they generally have several similarities. They all require knowledge of a central point for the spot they are going to analyze and need to know how the slide was laid out originally. At this point, they then all define some sort of boundary around the gene; in GenePix [3], this is a circle of varying diameter, while other techniques use the partitioning of the pixels' values by use of a histogram [4], [9] and growing a region from the center [1]. For a comparison of these techniques and more details about their implementation, see Yang *et al.* [19], who have compared them all in detail.

Manuscript received December 20, 2002; revised June 16, 2003. This work was supported by the Engineering and Physical Sciences Research Council Grant GR/P00307/01 and the Biotechnology and Biological Sciences Research Council Grant 100/BIO14300;100/EGM17735. *Asterisk indicates corresponding author.*

P. O'Neill and G. D. Magoulas are with the Department of Information Systems and Computing, Brunel University, Uxbridge UB8 3PH, U.K. (e-mail: paul.oneill@brunel.ac.uk; George.Magoulas@brunel.ac.uk).

*X. Liu is with the Department of Information Systems and Computing at Brunel University, Uxbridge, Middlesex UB8 3PH, U.K. (e-mail: Xiaohui.Liu@brunel.ac.uk).

Digital Object Identifier 10.1109/TNB.2003.817022

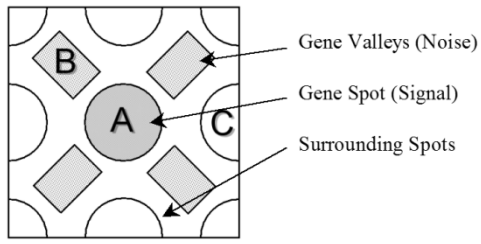


Fig. 1. GenePix sampling method.

For this paper, we will focus on comparing our method to that of GenePix, a software package commonly used by biologists. As shown in Fig. 1, GenePix first fits a circle to the gene itself using a simple threshold to calculate the diameter; then, the median value of these pixels is used as the intensity of spot signal (see area A in Fig. 1). The technique samples the surrounding noise by placing four rectangular regions (e.g., area B) in the diagonal space between this and adjacent spots (e.g., area C). Again, the median value of all the pixels within these regions is taken as the noise. The final stage is to subtract the noise from the signal measurement, and once this is done for the red and green channels of the image, the ratio of the spot can be calculated. This process makes the assumption that there is little variation both within the gene spot and in the surrounding background noise and that an exact region can be fitted to the gene. Unfortunately, this is not always the case, as various types of artifacts are commonly found. In the next section, we describe a technique that deals with this problem.

III. New Technique

In our attempt to develop a resilient technique that will be able to deal with the types of artifacts that are commonly found both in and around the gene spots, we decided to look at ways of recreating the noise that is found behind the spot itself. Rather than relying on a median sample of the surrounding pixels, we will recreate by a process of interpolation what we believe the noise behind the spot to have been. The image reconstruction technique (IRT) is unlike other methods that focus on the signal of the gene spots. Instead, we focus on the noise from start to finish. The result of this process is an image of the noise that can be subtracted from the original to leave us with the remaining signal. The genes can then be calculated in a normal fashion by measuring the intensity of the remaining pixels for each gene. This method is advantageous, as it inherently deals with slide variation and a large degree of slide artifacts. Due to its nature, it does not necessitate the exact fitting of a boundary around the spot, which is very important if this technique is to be automated.

A. Description

The algorithm we propose requires that we can successfully reconstruct the noise in the part of the image where the gene resides. The technique used is a simplified version of the method proposed by Efros *et al.* [5], a well-established technique for image tiling and hole filling. For example, if there is a hair on an image that we wish to remove, these techniques can take a mask of the artifact and then recreate the image as it would have been. Applying this to microarray images, we mask out all the genes so that we can recreate the noise. The original version of the method proposed in [5] is suitable for processing individual genes; however, in order to process the thousands of genes involved in a typical experiment, it proved to be too time consuming.

To this end, we tested several variations inspired by [5] and came up with the following process that should be applied to every gene in each channel of the slide. The main difference between this and the method proposed in [5] is that we do not use the Markov random field model in selecting pixels. Instead, for any given pixel that we need to fill, we compute and score candidate pixels based on a simple distance score of the surrounding pixels within a given window.

Let us first define the set of all pixels within a generic image window of size w, i.e., $\Omega^P(w) = \{p_{ij}\}$, where the relative coordinates, $i, j \in$ Z, of a pixel p take values in the range $[-W, +W]$, and the window is centered at p_{xy} with coordinates x and y, where $W = \lfloor (1/2)w \rfloor$. Next, we define a set of pixels in a window centered on the gene, G, with a width of SampleSize denoted by s. $\Omega^G(s)$ is the set of pixels centered on the gene spot coordinates G_{xy} with a window width of s. From this, we create a subset of $\Omega^G(s)$, $S^P = \{p_{ij}\}$, where all pixels fall within the spot boundary and need to be reconstructed. Next, we define another subset of $\Omega^G(s)$; $B^P(m) = \{p_{ij}\}$, where no pixel within the window $\Omega^P(m)$ with width MatchWindow, m, centered around p_{ij} belong to this gene or any adjacent genes. Therefore $S^P \cap B^P = \emptyset$ and $S^P \cup B^P \neq \Omega^G(s)$. If we now define $\Im(p)$ as the intensity of pixel p, the reconstruction process can be described as follows; for each pixel p in S^P, we find the best matching pixel b from the set of sample pixels, B^P. Then we set the intensity of the pixel p to that of b, $\Im(p) := \Im(b)$. In order to find the best match for pixel p, we compare it to all the pixels in the background set B^P. Therefore pixel b will be the pixel with the minimum match score M, defined by

$$L^{Dst} = C^p_{\Omega^p(m) - S^P} \tag{1}$$

$$L^{Src} = C^b_{\Omega^b(m)} \cup L^{Dst} \tag{2}$$

$$M = \frac{\sum_{n=1}^{N} \left| \Im^p(L_n^{Dst}) - \Im^b(L_n^{Src}) \right|}{N} \tag{3}$$

where $C^P_X = \{\{c_{ij}\} | \forall\, c_{ij}(p_{ij}, P_{xy}) \in X\}$ creates a set of pixel coordinates based on $c_{ij}(p_{x_1y_1}, p_{x_2y_2})$, the relative coordinates i and j of pixel $p_{x_1y_1}$ relative to pixel $p_{x_2y_2}$ ($i = x_1 - x_2$ and $j = y_1 - y_2$); $\Im^p(p_{ij})$ is the intensity of the pixel with relative coordinates p_{ij} in respect to pixel p; N is the total number of pixels being compared. This is in essence the average Manhattan distance of the surrounding pixels within the window m.

B. Example

We now use the example shown in Fig. 2 to explain the process previously outlined. The method first of all creates two lists, one with a set of surrounding background pixels (Background Noise) within a square (Sample Window) centered on the gene spot with dimensions SampleSize, and the other a set of pixels in the spot that need to be reconstructed. Then, the list of pixels to be filled is sorted in order, based on how many

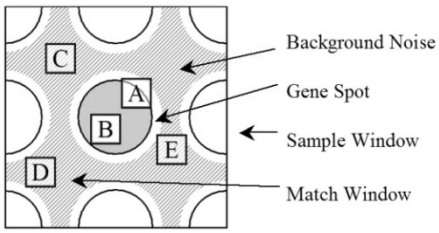


Fig. 2. IRT sampling method.

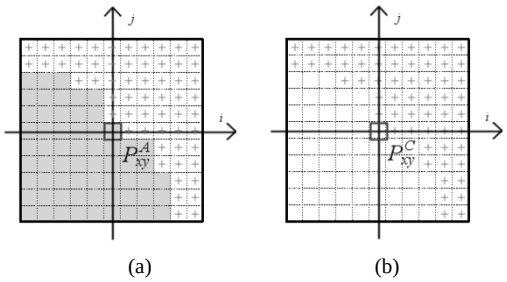


Fig. 3. An example MatchWindow. (a) Pixel to be filled (A). (b) Pixel being tested (C).

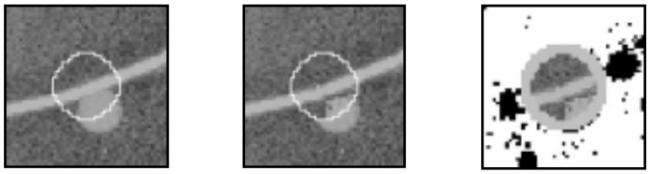


Fig. 4. An example reconstruction applied to a blank area of a slide with artifacts. (a) The original area. (b) The reconstructed area. (c) The sample locations used in the reconstruction.

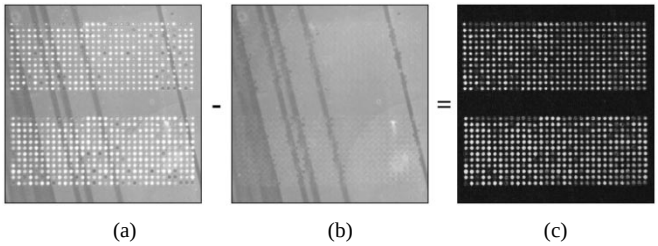


Fig. 5. An example reconstruction: (a) the original image minus (b) the reconstructed background image leaves (c) the final gene image.

surrounding pixels exist around each pixel within a square of dimensions MatchSize (e.g., Sample Windows A or B). The pixel with the most neighbors is selected (e.g., pixel p_A in window A). This is then scored against all of the pixels within the background area. Scoring is based on how closely the surrounding pixels present within a square (Match Window), defined by MatchSize, match one another.

For example, when matching pixel p^A_{ij}, the window of pixels around it would be compared to the corresponding background pixels (for example, those in windows C, D, E, and so on). This is illustrated in Fig. 3, where Fig. 3(a) shows the pixel which is being reconstructed and Fig. 3(b) shows the background pixel. The gray area shows pixels that have not yet been reconstructed; the pixels denoted by the "+" symbol are those that will be used in the comparison. In our techniques, this comparison consists of calculating the average Manhattan distance of all these pixels (i and j are the relative pixel coordinates from the pixel in question).

During initial testing of this method, we conducted numerous reconstructions on blank areas of the slides. These are areas that do not contain gene spots but do, however, contain various artifacts. Fig. 4 shows an example reconstruction; Fig. 4(a) is the original image with two easily distinguishable artifacts that intersect a circular region that we plan to restore. Without knowledge of this circular region, Fig. 4(b) shows the reconstruction, although not perfect it matches the original very closely. Fig. 4(c) gives an indication as to how this image was reconstructed. In the center you can see the reconstructed area. The gray circular region around this is a buffer zone placed around the reconstructed area; no pixels within this region are used in the reconstruction. Outside this buffer zone, black pixels represent pixels that were used to reconstruct the image. Here we can see that a lot of pixels from the hair-like artifact have been used within the reconstructed artifact within the central region.

Various measures such as the Euclidean or the Manhattan distance can be used for the comparisons. Following some preliminary testing, we have found that the average Manhattan distance between all noise pixel intensities p^X_{ij} gave the best performance (where i, j are the relative coordinates of the pixels within each window centered around pixel p). For example, this was calculated between all noise pixel intensities p^A_{ij} in a window A of width MatchSize and the corresponding pixel intensities of a background window of similar width, such as the pixels p^C_{ij} of window C. This calculation was repeated for all background pixels, and then the intensity value of the pixel with minimum average distance was used to reconstruct the intensity of p^A_{ij}. This procedure was then repeated for all pixels within the gene spot.

To get a glimpse at how this would work on full slide, Fig. 5 shows two blocks of genes. In total, there are 768 gene spots in two blocks. The image on the left [Fig. 5(a)] shows the original slide. In the middle [Fig. 5(b)] is an example reconstruction of this slide surface, and on the right [Fig. 5(c)] is the difference between the two; as you can see, only the gene spots remain with the noise removed.

IV. Supporting Material

A. Dataset

The slides used in this paper came from two experiments that were conducted using the human gen1 clone set slides [10]. The experiments were designed to contrast the effects of two cancer inhibiting drugs (PolyIC and LPS) on two different cell lines, one normal (Control) and one cancerous (HeLa), over a series of time points. Altogether, this gives us a total of 48 microarray slides for which we have the uncompressed image data and the initial GenePix results. Each slide in this dataset consists of 12 gene blocks that have 32 columns and 12 rows. The first row of each odd-numbered block is the Lucidea scorecard [15] consisting of a set of 32 predefined genes that can be used to test the accuracy of the experimental process. The remaining 11 rows of each block are the human genes.

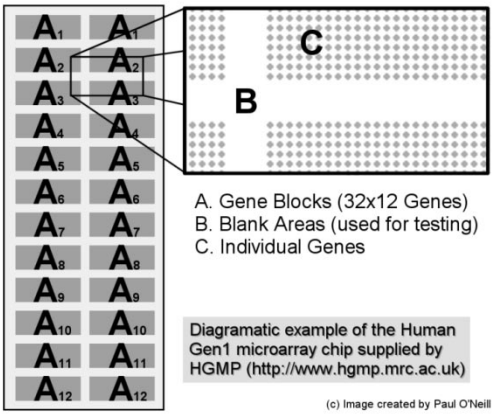


Fig. 6. Overview of slide layout.

This is depicted in Fig. 6, with each of the 12 gene blocks labeled "$\mathrm{A_n}$." As you can see from this diagram, each of the blocks on the left have biological repeats on the right. This is important, as it means that on each slide there are a large number of repeats, 24 repeats of each of the 32 scorecard genes and duplicates of each of the remaining 4224 ($32 \times 11 \times 12$) human genes. Another useful characteristic of this slide layout is the large area that is left between each of the gene blocks, an example of which is annotated "B." This is a useful area for testing background reconstruction techniques such as those used by GenePix and the method we propose.

The scanned images of the spotted microarray slides are stored in the standard TIFF format. Some scanning software will store the red and the green channels separately, while others, such as the GenePix software, store the two channels and preview images in a single composite file. In either case, there are essentially two TIFF images that we are interested in. These both have exactly the same properties and represent the two channels of the scanned image, Cy5 (red) and Cy3 (green). Each image is stored as a uncompressed 16-b gray-level bitmap, with each pixel taking a value between 0 and 65 535. It is generally accepted that any pixels at the extremes of this scale (i.e., 0 or 65 535) should be ignored, as these are beyond the range of the scanner to accurately measure.

Along with the scanned images, another useful resource is the GenePix result files that store the initial analysis from the software provided with the scanner. These files store a list of all of the genes along with detailed information about each spot. This information is in the form of spot coordinates, spot diameter, its ratio, and various statistical measures such as the standard deviation of the pixels within the spot. Although GenePix can work in an automated mode when locating all the spots, it also allows for manual intervention. In the case of the GenePix results we compare our method to, all spots on every slide have been visually inspected and manually corrected if necessary. This gives us a high degree of confidence that the GenePix results are a good representation of the program's best capabilities.

B. Weighted Kappa

A useful metric for scoring clustering results is something that can often be found in medical statistics, weighted kappa (WK). WK is used to rate the agreement between the classification decisions made by two observers [2]. In this case, observer 1 is the cluster output from the algorithm being tested, and observer 2 is the known arrangement in which the genes should be clustered. The formulas for calculating this metric are as follows:

$$K_w = \frac{p_{o(w)} - p_{e(w)}}{1 - p_{e(w)}} \tag{4}$$

$$p_{e(w)} = \frac{1}{N_k^2} \sum_{i=1}^{k} \sum_{j=1}^{k} w_{ij} \mathrm{Row}(i) \mathrm{Col}(j) \tag{5}$$

$$p_{o(w)} = \frac{1}{N_k} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \mathrm{Count}_{ij} \tag{6}$$

$$w_{ij} = 1 - \frac{|i-j|}{N-1} \qquad \text{where } 1 \le i,\, j \le k \tag{7}$$

$$N_k = \sum_{i=1}^{k} \sum_{j=1}^{k} \mathrm{Count}_{ij} = \sum_{i=1}^{k} \mathrm{Row}(i) = \sum_{i=1}^{k} \mathrm{Col}(i) \tag{8}$$

where K_w is the WK agreement; $P_{o(w)}$ and $P_{e(w)}$ are the observed weighted proportional agreement and the expected weighted proportional agreement, respectively; $\mathrm{Row}(i)$ and $\mathrm{Col}(i)$ are row and column totals respectively; the sum of all of the cells is denoted by N_k, and Count_{ij} is the count for a particular combination of classifications; and k is the number of classes (i.e., we have two classes: the algorithms either agree or disagree; thus, $k = 2$). This metric will score the output clusters against the expected clusters and give us a value between -1 and 1; the higher the value, the better the arrangement, with 1 being total agreement. Based on this metric, we can now score how well the results produced via the image reconstruction and GenePix compares to the original arrangement of genes.

C. Clustering Methods

1) Hierarchical Clustering: This method of clustering yields a dendrogram (binary tree) representing the nested clusters of patterns and similarity levels at which clusters change [18]. The dendrograms can be broken at different levels to yield different clusterings of the data. There are different variants of this algorithm, but most are agglomerative, and involve merging then adding nodes (which can be either single variables or clusters) to form larger clusters based on some minimum distance criteria. The algorithm for hierarchical clustering is described as follows.

```
Input:
D:    A distance matrix between n objects

Output:
H(0) ··· H(n)  A Dendrogram Representing
the Sequence of Merges

Algorithm:
Let H(0) = {{1}, {2}, ···, {n}}, i.e., the
n Objects in Their Own Clusters
```

```
For i = 1 to n
  Choose the Two Most Related Clusters in
H(i-1) According to D
    Update D
  Merge These Two Clusters Creating H(i)
End For
```

In order to extract a given number of clusters from the dendrogram, the Cutree technique used in the R statistical package[1] can be applied.

2) Partitioning Around Medoids (PAM): Partitioning around medoids (PAM) [12] works by first selecting m out of n total objects that are the closest (according to the distance matrix) to the remaining $(n-m)$ objects. The fitness of these medoids is calculated by placing the remaining $(n-m)$ objects in a group according to the nearest medoid and summing up all of the distances of the group members from this medoid. These m selected objects are the initial medoids. A swapping procedure is then applied until there is no improvement in fitness. Swapping involves generating all of the possible medoid and nonmedoid pairs, evaluating the fitness of each pair, and then performing the swap that improves fitness the most. PAM is described in the following pseudocode:

```
Input:
D:   A distance matrix
Iterations:
How long to run the algorithm for
n:   The number of Objects being clus-
tered
m:   The number of clusters required

Output:
z:   m clusters

Algorithm:

Construct m Initial Medoids, z_j that mini-
mize Σ_{i=1}^{n} D(i, z_j)
For i = 1 to Iterations
For all Object Pairs,(i,j), Where i
Medoids and j Medoids
Swap the i,j Pair Which Increases the Fit-
ness the Most
End For
End For
Allocate Each Medoid to a New Cluster
Allocate the Nonmedoids to Their Nearest
Medoid
```

D. Pearson's Correlation

In order to use clustering methods such as hierarchical, a method is needed to compare genes to one another. Pearson's correlation coefficient (r) measures linear relationships between two variables, x_1 and x_2, either discrete or continuous and is defined in (9)

$$r(x_1, x_2) = \frac{\sum_{i=1}^{T}(x_{1i}-\mu_1)(x_{2i}-\mu_2)}{\sqrt{\sum_{i=1}^{T}(x_{1i}-\mu_1)^2 \cdot \sum_{i=1}^{T}(x_{2i}-\mu_2)^2}} \quad (9)$$

where μ_1 and μ_2 is the mean of variables x_1 and x_2, respectively, and x_{1i} and x_{2i} are the ith observations of variables x_1 and x_2. Note that $-1 \leq r \leq 1$. This gives us a measure of correlation between the genes with 0 being no correlation, -1 exact inverse correlation, and 1 exact correlation (i.e., identical).

V. Experimental Study

In this section, we provide an empirical comparison between the two methods in the form of four tests all conducted on a large real-world gene expression dataset. The first two tests take a look at the performance of the methods based on absolute error and the standard deviation of biological repeats. The third test looks at the variation of 202 752 duplicate results, and the fourth test looks at scoring clustering results in an attempt to assess the impact of these techniques on later processing of the data.

A. Absolute Error

The method that has been proposed relies on the fact that it can reconstruct the area behind a spot. Therefore, the first line of testing was to see if this could be achieved. There is no way that you can truly know what the background would have been behind the gene spot; therefore, it is impossible to verify the technique on the gene spot locations. However, if we instead look at blank areas of the slide, then this is no longer the case, and we can easily test how well the reconstruction methods work. This can be achieved by locating an empty portion of the slide, usually between the 24 blocks of genes; then, by cutting a hole in the slide and recreating the background, we can then verify the recreated sample against what we know to have been there previously. From this, we can calculate the absolute error of the reconstruction and then compare this to the other processing techniques such as GenePix.

For this experiment, three slides were picked at random, and 1056 spots of random size were positioned within the blank regions for both the Cy5 and Cy3 channels. After reconstructing each spot, the difference between the recreated background and the true background was compared to the difference between the GenePix noise estimation and the true pixel values. If the image reconstruction was closer to the true background, then this was counted as a positive improvement.

On average, 73% of spots were improved, although it would be more interesting if we can see how those spots, where GenePix estimated the background better, compared to those which benefited from using the image reconstruction.

Fig. 7 is a graph that shows just this. The positive y axis of the graph shows how much each pixel within all the spots was improved when using the IRT. This is contrasted to the negative portion of the y axis that shows the improvement that was

[1]http://www.r-project.org

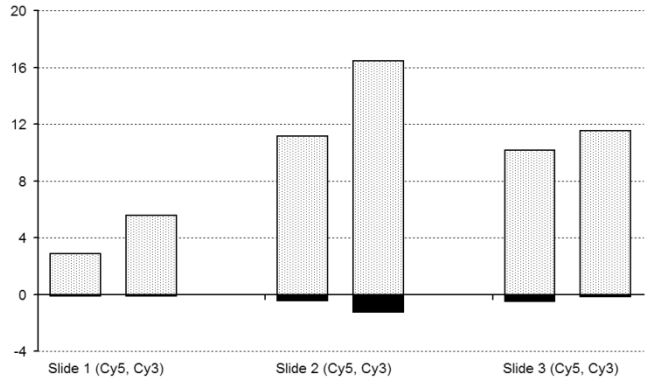
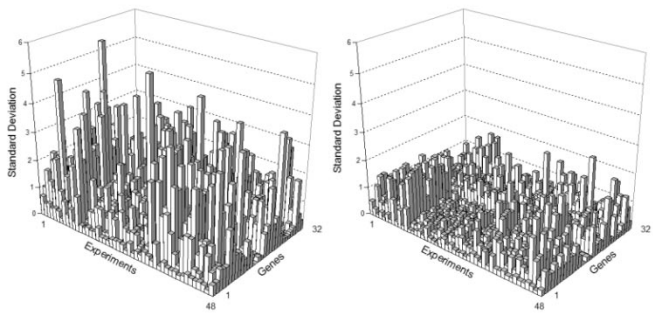


Fig. 7. The gained improvement in pixel values (positive axis) against the loss in erroneous pixels (negative axis).



(a) (b)

Fig. 8. The standard deviation of 32 genes repeated 24 times across 48 experiments processed using the IRT. (a) GenePix. (b) IRT.

gained when using GenePix for those spots that showed negative results. It is clear from the graph that image reconstruction is having a significant effect on how the noise is estimated for any given spot value. Using the IRT not only improves nearly three quarters of the spots processed, but also this improvement is in orders of magnitude higher than the negative effects which are seen in just over one quarter of the spots processed.

B. Variation in Scorecard Gene

As we stated before, the dataset we are working on contains a high number of repeats for each slide. First of all, there are 24 repeats of 32 control genes on each slide, followed by duplicates of 4224 human genes. For this section, we will only be using the control data, as we have the most repeats for these. This experiment analyzes the standard deviation between the 24 repeats of the scorecard genes on each of the 48 slides. This is based on the assumption that if a gene sample is repeated on a slide, then it is reasonable to expect its ratio to be the same. Therefore, if a given technique is producing a lower standard deviation between repeated measures of the same gene, it must be analyzing that gene more successfully. Fig. 8 shows the standard deviation of each of the 32 genes for the 48 experiments. It is clear from this that the image processing technique we present in Fig. 8(b) has significantly reduced the standard deviation. It should be noted that the IRT also allowed us to measure the ratios of an extra 8% of the genes (2908 genes); this could be due to the IRT method correcting for these noise in these genes sufficiently enough to be able to measure their expression, whereas GenePix was unable

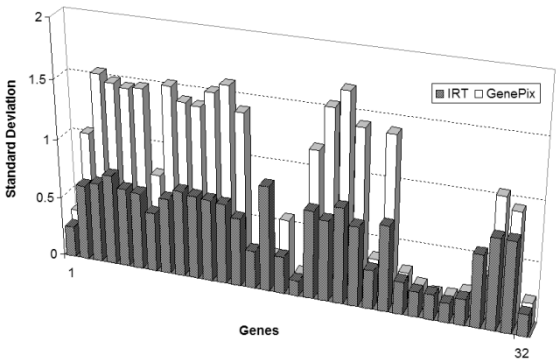


Fig. 9. Overview of the standard deviation across the 32 repeated genes.

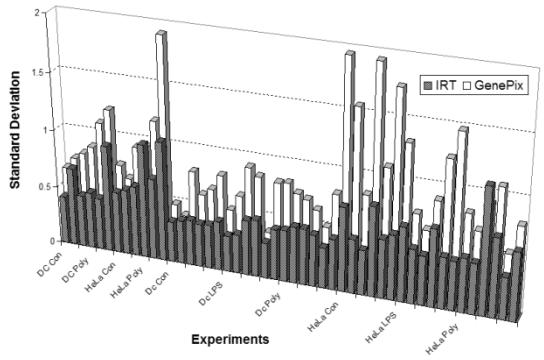


Fig. 10. Overview of the standard deviation across the 48 experiments.

to remove the noise. To summarize these results, the mean standard deviation is reduced from 0.91 to 0.53, and the maximum standard deviation is reduced from 6.23 to 2.43.

Fig. 9 is an average cross section of Fig. 8, plotting the mean of all 32 repeated genes over the 48 experiments. The white bars show how GenePix performed against the image reconstruction in gray. Altogether, 28 of the 32 genes showed a reduction in standard deviation or no change.

Another interesting way of looking at this data is to average all the genes across each of the 48 experiments as shown in Fig. 10. Here again, we can see that the image reconstruction significantly reduces the standard deviation, with 46 out of the 48 experiments showing an overall reduction in standard deviation.

Looking at the previous two summaries of results, it seemed that certain experiments benefited while others remained unchanged, but we do not know exactly which slides or genes. To get an overview of the entire experiment, we produced Fig. 11.

This shows those genes and experiments that were improved when using the IRT (white), those that were better using GenePix (black), and those that were unchanged between techniques (gray). Here we can see that genes 5 to 8 remained largely unchanged; this is likely to be as these four genes were very highly expressed. However, to draw any real conclusions from this, we believe that it is not enough to just look at the results of one experiment; maybe in the future, after processing multiple experiments, it will become clear what types of spots different methods process most accurately. This could lead to being able to classify spots before processing and then selecting the best technique, allowing simple techniques to be used on

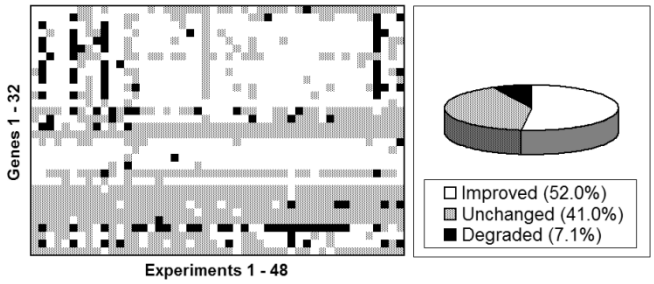


Fig. 11. An overview of which genes in which experiment showed improvement, those that did not and those that remained unchanged ($\leq \pm 0.1$ difference).

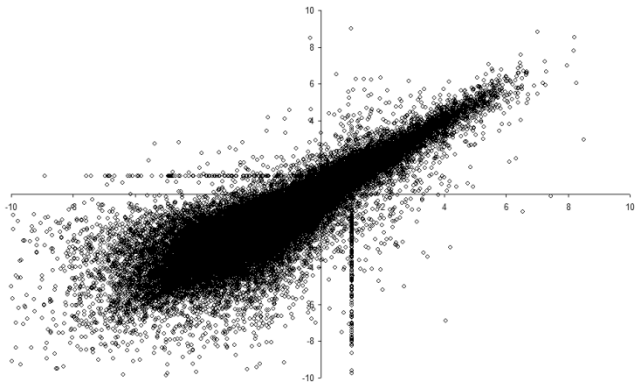


Fig. 12. Scatter plot of all 202 752 duplicate gene $\log_2$ ratios processed using GenePix.

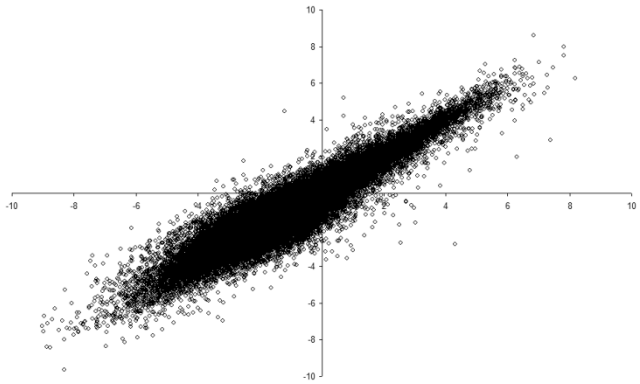


Fig. 13. Scatter plot of all 202 752 duplicate gene $\log_2$ ratios processed using the IRT.

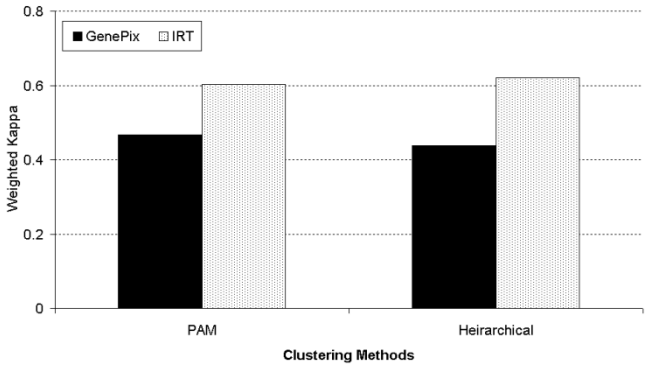


Fig. 14. WK of resultant clusters using hierarchical clustering.

relatively clean areas of the slide while using more robust but time-consuming techniques such as image reconstruction on the noisy sections or those affected by artifacts.

C. Variation in Human Genes

The previous experiments have focused on the scorecard genes that are present in all of the slides. This is because these genes had the largest number of repeats. However, this still leaves duplicates of the 4224 human genes from the actual experiment. Taking the $\log_2$ ratios of these genes and plotting them in scatter graph should show a linear relationship. The closer to the $y = x$ line the points are, the less variation there is. In Fig. 12 is the plot for all 4224 genes across the 48 slides for GenePix; Fig. 13 is the same for the IRT. From these graphs, it is clear that the IRT shows much less variation, especially for the negative ratios.

D. Clustering

The final method of testing we propose is the validation of clustering results. Normally, this would be difficult to achieve, as there is not enough substantial domain knowledge to be able to calculate a score for clusters. However, in this dataset we have a large number of repeats for a subset of the genes. These genes are the 32 control genes that make up the Lucidea scorecard found in the first row of every block. Using the 48 experimental slides, we now have enough data to enable us to use standard clustering algorithms to blindly group these genes together. Then, once we have the resulting clusters, we would expect to find all 24 repeats of the same gene clustered together. As each of the 32 genes in the scorecard are meant to be unique, we can also expect to find 32 unique clusters. As we now know how to expect the genes to form clusters, it is possible to score the results using the WK [2] metric. As explained before, WK scores agreement between the resultant clusters and known true clusters, with 0 being a random arrangement and 1 the perfect result. To verify the results, we chose to use hierarchical [18] clustering and partitioning around medoids [12]. For this technique, we used the implementation in the freeware statistical package R,[2] and the results are shown in Fig. 14. Here, we have only used those genes for which all the samples were present, and it is clear that there is a remarkable improvement when using the image reconstruction, with both algorithms showing nearly a 0.2 increase in their WK scores. This shows that the clustering results are much more accurate in terms of both an increase in the correct clustering of genes and a decrease in the incorrect clustering of genes.

VI. Conclusion

This paper proposed a new method of approaching microarray image analysis. Instead of focusing on the signal as with other techniques, we looked toward being able to reconstruct the noise accurately in a way that we could demonstrate to be more robust. To this end, we presented the results of several experiments run on real microarray experimental data

[2]http://www.r-project.com

to demonstrate the performance of this technique at various processing stages. The results obtained have shown a great improvement over a software package that is commonly used in the field. The technique greatly reduces the variance between repeats on a given slide; also, in looking at clustering results, we have shown that not only is the biological information left intact but it appears to be more accurate. In the future, we would like to extend our testing to include other mainstream methods that exist such as Spot [19], ScanAlyze [6], and QuantArray [9]. We plan to test the technique on other datasets and include the analysis of other clustering techniques along with verification against biological domain knowledge. This technique has already been shown to be more resilient to spot boundary fitting, and this may allow for more automated processing of microarrays in the future. To this end, it will be interesting to analyze exactly how it compares to other techniques in terms of robustness and whether it can be used automatically without loss of accuracy.

ACKNOWLEDGMENT

The authors would like to thank P. Kellam and A. Kwan from the Department of Immunology and Molecular Pathology, University College, London, U.K, for providing the datasets used in this study.

REFERENCES

[1] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 641–647, June 1994.
[2] D. G. Altman, *Practical Statistics for Medical Research*. London, U.K.: Chapman & Hall, 1997.
[3] Axon Instruments Inc.. GenePix Pro Array analysis software. [Online] Available: http://www.axon.com
[4] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *J. Biomed. Optics*, vol. 2, pp. 364–374, 1997.
[5] A. A. Efros and T. K. Leung, "Texture synthesis by nonparametric sampling," in *Proc. IEEE Int. Conf. Computer Vision*, 1999, pp. 1033–1038.
[6] M. B. Eisen. (1999) ScanAlyze. [Online] Available: http://rana.lbl.gov/index.htm?software
[7] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sci.*, vol. 95, pp. 14 863–14 868, 1998.
[8] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown, "Genomic expression programs in the response of yeast cells to environmental changes," *Mol. Biol. Cell*, vol. 11, pp. 4241–4257, 2000.
[9] GSI Lumonics. QuantArray analysis software. [Online] Available: http://www.gsilumonics.com
[10] Human Genome Project Mapping Resource Centre. Human gen1 clone set array. [Online] Available: http://www.hgmp.mrc.ac.uk/Research/Microarray/HGMP-RC_Microarrays/description_of_arrays.jsp
[11] C. Kooperberg, T. G. Fazzio, J. J. Delrow, and T. Tsukiyama, "Improved background correction for spotted DNA microarrays," *J. Comput. Biol.*, vol. 9, pp. 55–66, 2002.
[12] L. Kaufman and P. J. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L1-Norm*, Y. Dodge, Ed. Amsterdam, The Netherlands: North-Holland, 1987, pp. 405–416.
[13] M. A. Newton, C. M. Kendziorski, C. S. Richmond, F. R. Blattner, and K. W. Tsui, "On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data," *J. Comput. Biol.*, vol. 8, pp. 37–52, 2000.
[14] J. Quackenbush, "Computational analysis of microarray data," *Natl. Rev. Genet.*, vol. 2, no. 6, pp. 418–27, 2001.
[15] H. Samartzidou, L. Turner, T. Houts, M. Frome, J. Worley, and H. Albertsen. (2001) Lucidea Microarray ScoreCard: An integrated analysis tool for microarray experiments. *Life Sci. News* [Online] Available: http://www1.amershambiosciences.com/aptrix/upp01077.nsf/Content/lsn_online_issue_7_13
[16] (1999, Jan) The chipping forcast I. *Nature Genetics Suppl.* [Online] Available: http://www.nature.com/cgi-taf/DynaPage.taf?file=/ng/journal/v21/n1s/index.html
[17] (2002, Dec.) The chipping forcast II. *Nature Genetics Suppl.* [Online] Available: http://www.nature.com/cgi-taf/dynapage.taf?file=/ng/journal/v32/n4s/index.html
[18] J. H. Ward, "Hierarchical grouping to optimize an objective function," *J. Amer. Stat. Assoc.*, vol. 58, pp. 236–44, 1963.
[19] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," *J. Comput. Graphical Stat.*, vol. 11, pp. 108–136, 2002.

Paul O'Neill is currently working toward the Ph.D. degree in intelligent data analysis at Brunel University, London, U.K., focusing primarily on the pre- and postprocessing of microarray image data.

George D. Magoulas (M'02) is a Lecturer in the Department of Information Systems and Computing at Brunel University. His research is focused on learning and evolution algorithms for intelligent systems with applications to adaptive systems and bioinformatics.

Dr. Magoulas is a Member of the Operational Research Society, the Technical Chamber of Greece, and the Hellenic Artificial Intelligence Society.

Xiaohui Liu is Professor of Computing at Brunel University, London, U.K., where he leads the Intelligent Data Analysis (IDA) Group, performing cutting-edge interdisciplinary research involving artificial intelligence, dynamic systems, signal processing, and statistics, particularly for biomedical and engineering applications. He serves on the editorial boards of four computing journals, founded the biennial international conference series on IDA in 1995, and has given numerous invited talks, including a keynote at the International Conference of the Royal Statistical Society in 2002.