Copasetic analysis: a framework for the blind analysis of microarray imagery

K. Fraser, P. O'Neill, Z. Wang and X. Liu

Abstract: From its conception, bioinformatics has been a multidisciplinary field which blends domain expert knowledge with new and existing processing techniques, all of which are focused on a common goal. Typically, these techniques have focused on the direct analysis of raw microarray image data. Unfortunately, this fails to utilise the image's full potential and in practice, this results in the lab technician having to guide the analysis algorithms. This paper presents a dynamic framework that aims to automate the process of microarray image analysis using a variety of techniques. An overview of the entire framework process is presented, the robustness of which is challenged throughout with a selection of real examples containing varying degrees of noise. The results show the potential of the proposed framework in its ability to determine slide layout accurately and perform analysis without prior structural knowledge. The algorithm achieves approximately, a 1 to 3 dB improved peak signal-to-noise ratio compared to conventional processing techniques like those implemented in GenePix® when used by a trained operator. As far as the authors are aware, this is the first time such a comprehensive framework concept has been directly applied to the area of microarray image analysis.

1 Introduction

Microarrays allow biologists to simultaneously analyse the expression level of many thousands of genes. This is accomplished by a technique called *competitive hybridisation*, which is conducted on a microscopic scale [1]. Here, we provide a brief review of relevant background material; for a more detailed explanation readers may find references [2] and [3] of interest. Typically, on a slide surface area of less than 24 cm^2, receptors for as many as 30 000 genes can be printed and analysed. To process this so-called chip, the slide is digitised using a dual-laser scanning device, producing a two-channel 16-bit grey-scale image. The gene receptor locations in this image (typically 16 to 20 pixels in diameter) are identified, their median intensity value is measured and then summarised as $\log_2$ ratios across both channels.

When using the aforementioned technology, one of the largest expenses involved is the lab technician's time in the preparation and post-processing of these cDNA microarray chips. In current microarray image analysis systems there is still too much reliance on operator intervention. Therefore, more autonomous stages of processing are required, allowing valuable laboratory time to be spent working on the biology rather than wasted in front of a terminal aligning the gene spots. Hence, there is an urgent need for developing an effective technique in order to reduce the time and effort for the technicians, and also maximise the potential use of available microarray images. In this paper we present a framework for a technique that allows both standard and custom microarray imagery to be processed with no prior knowledge of the slide; in fact, the only assumption we make about the input image is that it will have some sort of regular structure. The framework is designed to work with both single- and multi-channel data. Therefore, the microarray channels can be supplied either individually or together, so that extra information can be gained by comparing similarities and differences across the channels. The support for multi-channel analysis is not restricted to the processing of microarrays. For example, the processing of a colour image would also require that separate red, green and blue images be analysed. The framework is supported at all stages with real experimental results from a variety of images that have been processed and is found to perform better than conventional techniques such as GenePix® when used by an experienced operator.

2 Background

Feature detection in cDNA microarray image analysis is the process whereby either an algorithm or an operator categorises the pixels in the image as belonging to either a specific gene spot or the background. This consists of two distinct stages: the first is 'spotting', such as the Bayesian approach proposed by Hartelius and Carstensen [4] which divides the imagery into manageable blocks; the second involves segmentation [5, 6], which classifies pixels in a region immediately surrounding a gene as belonging to either the foreground or background domains. Once the pixels for each spot have been identified, they can then be summarised as $\log_2$ gene ratios. For example, Yang *et al.* [7] present a detailed comparison of many traditional techniques used in this area.

The large amount of time that has to be spent on manually processing the microarrays has lead to the recent interest in fully automating the process. Bozinov and Rahnenfuhrer [8] proposed clustering the full image area in one step; however this is not computationally feasible with current processing power. To overcome these issues, an abstraction of the

Systems Biology online no. 20045002

doi: 10.1049/sb:20045002

Paper first received 10th March and in revised form 5th April 2004

The authors are with the Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, UK, email: xiaohui.liu@brunel.ac.uk

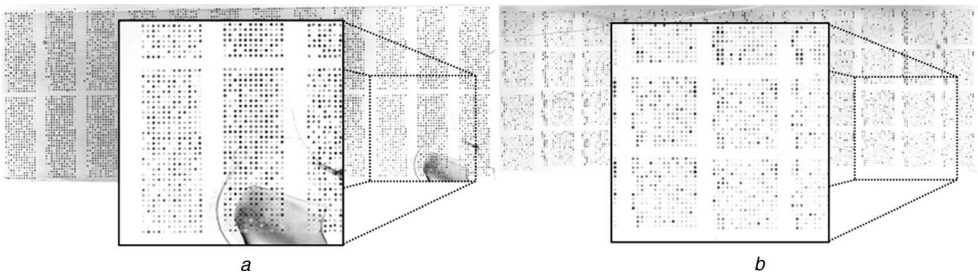

Fig. 1 *Example slides from test dataset illustrating varying structure and noise elements*

a Set 1: 12 × 2 Blocks, each 12 × 32 genes
b Set 2: 16 × 4 Blocks, each 15 × 16 genes
Note: *a* and *b* have been filtered to overcome printing difficulties

k-means [9] technique was proposed [10], whereby predefined centroids were chosen for both foreground and background, to which all pixel intensities could be assigned. Unfortunately, although traditional *k*-means is able to choose centroids according to the dataset's characteristics, this approach is inherently biased towards outlying values (saturated pixels for example), and not the true region of interest (the foreground pixels). Other methods, such as the application of wavelets [11] and Markov random fields [12] showed great promise. However, at this time they have only been attempted on what would be classified as 'good slides' (whereby there is noise but not of an extreme nature). If these techniques fail to determine the location of just one gene, the system would fail, thus having to fall back on user intervention in order to recover.

Overall our work has been based on slides representing two underlying structures with varying degrees of noise. Combined, these consist of ten images, which were selected because they contained varying anomalies both in the background intensities and in the printed spot structure. In Fig. 1, an image from each of these sets is displayed to highlight various issues. These example slides are also representative of the problems which are associated with the processing of this type of data, such as background artefacts and gene block misalignment. In the following Section, we present a framework which has been established to facilitate the processing of these images. This is a challenging and important problem and as far as we are aware, it is the first time such a comprehensive framework has been applied to microarray image analysis.

3 Copasetic analysis overview

Copasetic analysis (CA) is a framework in which automated blind microarray image analysis can be conducted. Unlike other techniques that have been proposed to this effect, it is not a rigid framework; in fact, it is its modularity and adaptability that give it its robustness. In Fig. 2, a skeletal structure of this process is presented showing the required stages, from the original input images through to the final stage of calculating the gene spot $\log_2$ ratios. In this diagram we can see there are four key parts which make up the CA process and these will be described in the remainder of this Section. Following this, the paper will focus in detail on two of these components. Importantly, each of these components are goal-orientated, which means the components, internal processes can be 'swapped out' but the computational task as a whole will remain unchanged. Some stages are composed of combinations of existing and new techniques, such as the *data services* stage, while others are novel algorithms like *Copasetic Clustering* which facilitates the application of existing clustering techniques to a dataset which previously would have been unfeasible. Another interesting point is the adaptability of the framework when things do not quite go as planned. For example, if a stage fails there is always the ability to backtrack. If the result of the current process is insufficient, at any point a stage can request a different view of the data from the *Image Transformation Engine* (ITE) so that the existing processing can be combined with results from a new perspective on the dataset.

This framework is designed to process images which have some form of regular structure like that found in microarray imagery and, as such, the input for the process is always going to be the raw image data. The ITE is the only component that has direct access to this raw data, its function being to supply this data both unaltered and in various transformed and filtered views to the components as requested. For example, the view requested could be a simple summary, such as providing the mean pixel intensity of the image, or a more complex image transformation and filtering technique. It is conceived that in this way components will not be restricted to one view of the data as is typical, but can benefit from a multitude of perspectives.

After the ITE has acquired the raw imagery, the first components in the framework to be executed are those that make up the *structure extrapolation* stage, which are designed to discover both the structure and composition of the image. The *Image Layout* component uses cross-sectional profiles of the image in order to ascertain the general layout of the image surface; this constitutes the discovery of the gene blocks. With microarrays, we know that the slide should have a regularly repeating structure in each gene block and therefore this information can be used to help guide the block structure discovery. With these gene blocks now defined, the *Image Structure* component then uses a similar process (conducted internally within each of these blocks) to define the individual gene spots. Alongside the discovery of the image structure, the image composition can also be analysed with segmentation techniques such as clustering. This compositional stage can either utilise the raw data or more beneficially, one of the alternate views as provided by the ITE service. In later Sections, we will describe in more detail the copasetic clustering (CC) technique which allows the application of these methods to images which would normally be too large to cluster.

From the structural information that has been determined, we can now start to identify objects of interest within the image, which constitutes grouping together all the pixels that form a gene spot. This is achieved by *Spatial Binding* which uses both the estimated gene centre position and the clustering results to search and combine groups of pixels that fall within close proximity to each other. The process can be completed for the majority of the genes that were well defined and the information gained can be used

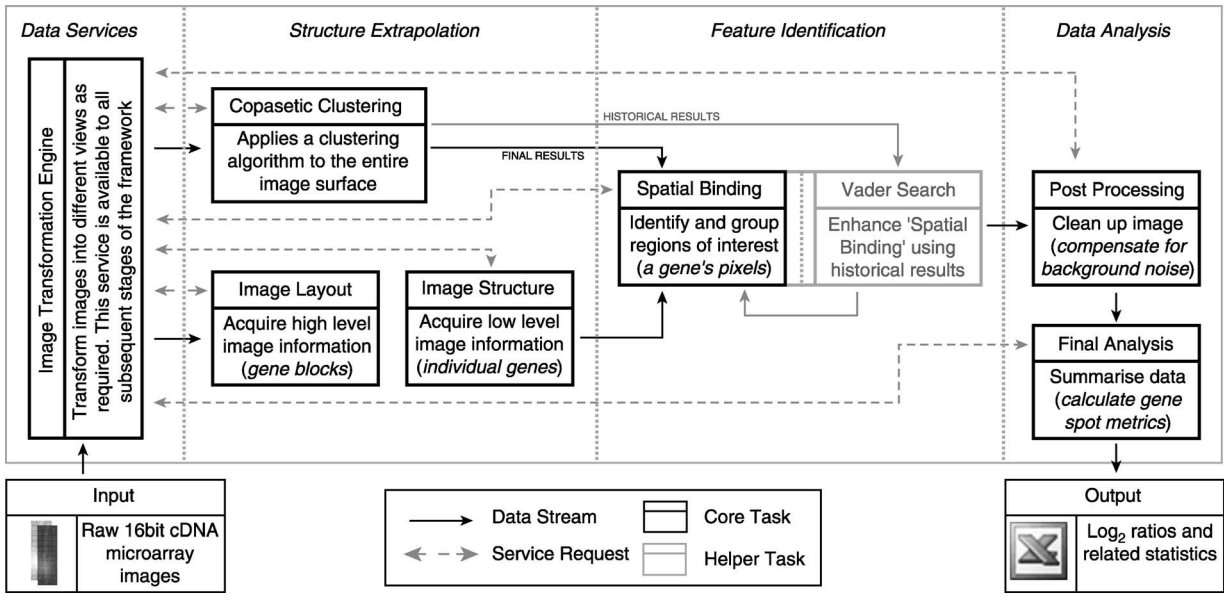

Fig. 2 *Copasetic analysis workflow diagram*

to generate an average gene spot. This mask construct is required, as clustering will never guarantee that all pixels belonging to gene spots will be identified correctly. One of the main strengths of CC is its transparency into the intermediate layers. Using this knowledge (when coupled with the average gene spot characteristics), we can move back through the historical clustering results until the criteria for the average gene spot have been satisfied.

The final stage consists of components that could be thought of as the more conventional stages of microarray analysis. Here, using the structural information that has been identified and the raw image data, the genes on the microarray are analysed. Generally this consists of two stages, one of *Post Processing* (such as background correction) and a second of *Final Analysis*, a data reduction stage (such as converting the values from all the pixels that make up a gene spot into one representative value). Background correction can take the gene locations that have been previously determined and perform local, global or combined correction for background noise. This could be accomplished using simple techniques such as subtracting a median sample of local background pixels or more robust approaches such as background reconstruction [13] which produce an estimate of the noise behind a gene spot. The $\log_2$ ratios themselves can be calculated using the spot masks created by the earlier analysis stages; this ensures that noise and other unwanted pixel values will not be measured, unlike common techniques such as fitting a fixed circular region around the gene spot.

Next we plan to present a more detailed explanation, focusing on the use of *Data Services* and how they facilitate the processing of imagery within this framework. This will be illustrated using some of the components that make up the *Structure Extrapolation* stage to highlight how multiple views of data, as provided by the ITE service, can facilitate improved processing of the imagery. Following this, a detailed explanation of one of the key components will be presented.

3.1 Data services

Before processing any type of data, it is normal to perform various stages of pre-processing which are designed to transform the data in such a way as to: reduce the amount of noise, subdue the effect of outlying data points, remove undue biases and reduce the overall volume of data to be processed. Normally this would constitute a one-time pre-processing stage of the data flow in which the data is cleaned or prepared in some way. This restricts the potential of the later steps, as they have no control over how this manipulation affects the data. What if, instead of just one view of the data, multiple views are constructed and used in unison? Each view of the data is tailor made to the task at hand and then propagated to later stages.

This idea is captured in the ITE service which essentially acts as a data warehouse, providing not just the original input data to those stages that need it, but also a variety of transformed views of the data. If a particular view of the data proves to be insufficient for the task at hand then the stage can request an alternative view. It is in fact, because of this concept of taking multiple views of the data, that the framework is potentially so powerful. It allows flexibility in the process pipeline where normally a series of predefined steps would take place. It also facilitates adaptability whereby the system does not simply succeed or fail at a given task, but instead can step back and thus try with a different view. In order to better understand this process, the remainder of this Section will focus on two of the components found in the *Structural Extrapolation* stage of the framework. The first example illustrates how it can be beneficial to use an alternative filtered view of the data in the context of clustering an image, and the second will highlight how a transformed view of the data can be used to greatly simplify the task of structure extrapolation.

3.1.1 Example 1: Clustering an image:

Clustering algorithms [14] have many uses and have already been shown to be beneficial in the processing of microarray imagery [15]. These traditional techniques could be applied to the original pixel intensities, but this is not really necessary in order to retrieve the structural data we require. In fact in some cases, such as a lowly expressed slide with sparse points of saturated noise, these techniques can be detrimental. However, the transformed data can be thought of as a compression (by use of a response curve) whereby we aim to reduce the unusable ranges of data as much as possible, while keeping the finer detail, thus helping to distinguish spot boundaries (similar to that of contrast and brightness adjustment). The output from this process would be of the same magnitude as the input, which in this case

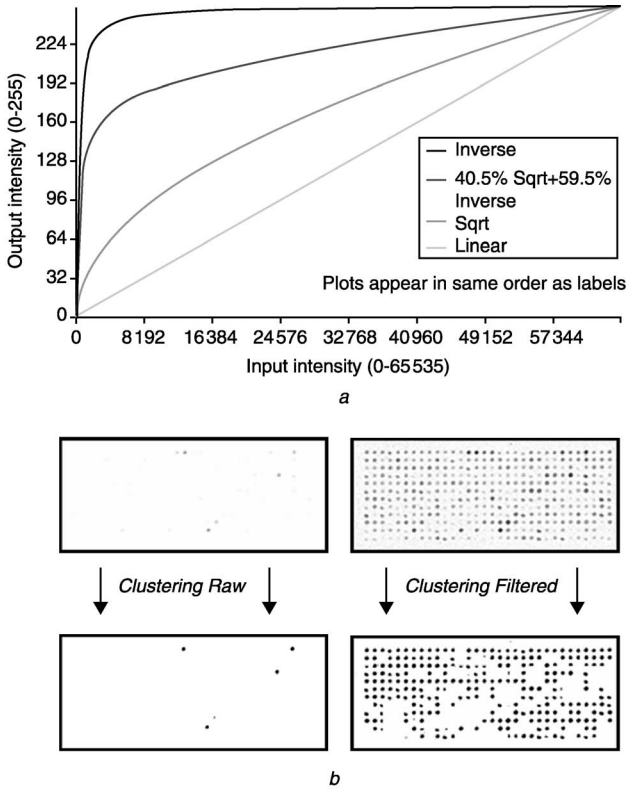

Fig. 3

a A plot of the response curves
b Their effect when used in conjunction with clustering

is 16-bit (0 to 65 535). However, it can also be rescaled to make it easier to process by other algorithms. The components discussed focused on an 8-bit scale (0 to 255) as this reduces the memory load significantly while still allowing a large range of variation between pixel intensities.

Figure 3*a* shows the differing methods of mapping the input and output pixel intensity, including (from top to bottom): an inverse, a summed inverse and square root, a square root and a linear function. Although the linear function sounds the most logical, in a dataset where the difference between normal genes and saturated ones is already heavily biased this can be problematic. Illustrated in Fig. 3*b*, fuzzy *c*-means clustering has been applied to an area of a typical microarray image. In the left-hand images labelled *Clustering Raw*, the top image shows the transformed input data and below the Boolean output of what is clustered as foreground (black) and background (white) pixels. Here we can see that the algorithm has done very poorly, identifying only four gene spots. However, if this same area of the image is first filtered using the inverse function to emphasise the genes, the *Clustering Filtered* image shows that there is a dramatic improvement. In the same way that a photographic polarising filter allows details normally hidden behind reflections to be captured, filtering the image in this way allows us to analyse details that would otherwise be lost.

3.1.2 Example 2: Discovering slide structure:

Before any image processing can occur, it is necessary to have an understanding of the data that is to be processed. The framework proposed in this paper is built on the understanding that no prior knowledge about the slide is utilised, and therefore all information needs to be extrapolated from the image itself. To facilitate this task, it is necessary to take a view of the data which emphasises the structure rather than the raw pixel intensities. One possible solution for this is to take an averaged cross-sectional view of the slide surface in the appropriate horizontal or vertical direction. Figure 4 shows two images with their corresponding horizontal profiles, where the light grey represents the image profile and the black line is generated using a combination of moving filters. This is a good example of how low pass filters can be applied in an attempt to improve the (subjectively measured) quality of the data for human or machine interpretability [16]. Figure 4*a* shows the profile for what would be classified as a 'good' slide, and from this it is relatively easy to distinguish the 12 block rows that exist in the slide (the peaks) and the inter-block gaps in-between (the valleys).

Figure 4*b* shows a slide which contains severe amounts of noise and irregularities in the gene layout structure. These slides were part of an early calibration run and were deemed inadequately hybridised to warrant further processing, but with this filtering method the information on the slide is still usable. With this transformed view, the blocks can be readily defined; it is also interesting to note the 'high slim peaks' on either side of each block, which are caused by corner guide spots present within each block structure.

3.2 Copasetic clustering

A useful method of analysing a slide's structure would be to cluster all the pixels into groups of either foreground or background. However, with current clustering techniques,

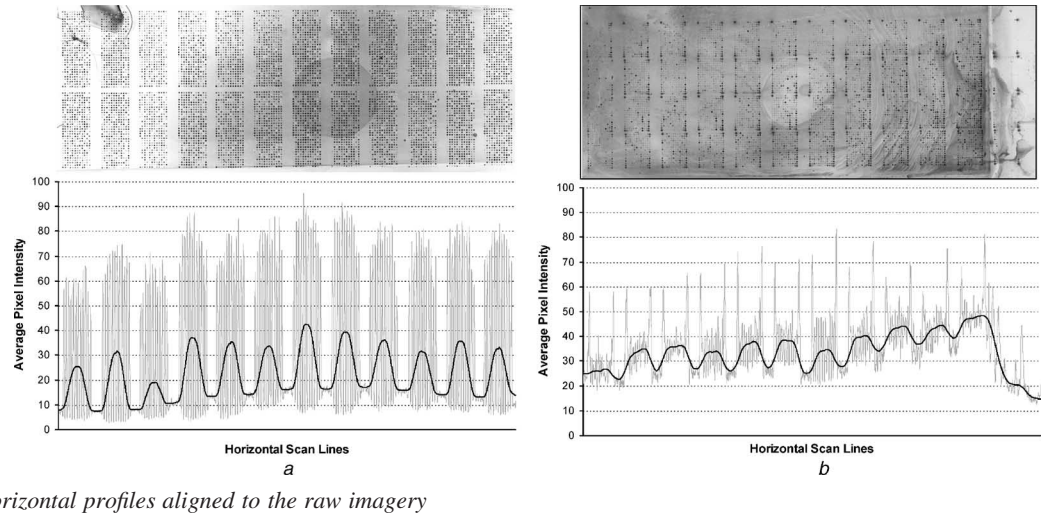

Fig. 4 *The horizontal profiles aligned to the raw imagery*

a Clean slide
b Corrupted slide

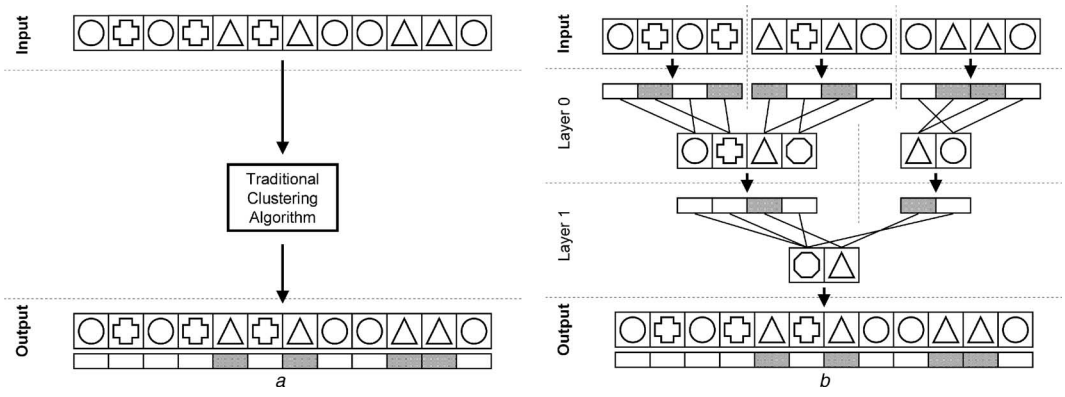

Fig. 5 *Conceptual diagram of clustering techniques applied to simplified data sets*
a Traditional clustering
b Copasetic clustering

this is unfeasible on such a large image. One suggested solution is to divide the slide into manageable blocks, often into the genes themselves [15]. Without prior knowledge of the slide structure this is again unfeasible. Therefore, the remainder of this Section describes a process whereby existing techniques can be scaled to larger datasets, not only making their application feasible but also providing extra information about intermediate steps which would traditionally be unavailable.

The CC method is a technique which facilitates the application of traditional clustering algorithms to large-scale data sets; it also has the additional ability of being able to capture spatial information allowing the refinement of groupings if so desired. Initially it arbitrarily divides the image into spatially related areas (normally very small grid squares). Each of these areas is then clustered using a traditional technique such as *k*-means [10] or fuzzy *c*-means [17], and the result is stored. Then representatives are calculated for each of the clusters that exist and these are then further clustered in the next generation. Such a process is repeated until all sub-clustered groups have been merged. The final result is that every pixel will have been clustered into one of *n* groups and on small data sets the output is comparable with traditional techniques. The aforementioned idea can be illustrated using the conceptual diagram shown in Fig. 5. In the *Input*, we can see 12 items which are to be clustered. Creating two groups with traditional clustering methods would compare all 12 shapes and is likely to output one group of triangles and one containing a mixture of circles and crosses as shown in Fig. 5*a*. CC, on the other hand, would divide this set into sub-groups of a given size, in this example, into three sets of four as in Fig. 5*b*. Each of these sub-groups would be clustered into one of two classes (represented by the checkerboard pattern or lack thereof, within the layer 0 stream located below the shapes). In layer 1, the previously generated representatives for each of these aforementioned groups have been clustered together. The final result can now be calculated by re-traversing the pyramid structure and tracking the shaded members.

Studying the three groups in this example closely, it can be seen that the shapes were chosen to illustrate how this sub-clustering can provide useful contextual information. The first group of circles and crosses can easily be clustered as two separate shape classes with no problems (as shown in the layer 0 stream). With the introduction of triangles in the second group, we see that the circle and cross are now clustered together; here this is the optimum arrangement due to shape similarity. By the layer 1 stream, the groups are the same as a traditional clustering algorithm, with all circles and crosses in one group and triangles in the other. Unlike a traditional clustering algorithm, information from previous layers can now be used to ascertain the fact that in some situations certain items should not have been grouped together. Here the circles and crosses could have formed two distinctive groups if it were not for the presence of the triangles. Note that this information is context specific, and relies on a spatial data set, an order dependent vector or image, for example.

To put this into context, think of the processing of a microarray image where the background is not consistent across the slide surface. The criteria for what will be classified as a background and foreground pixel will vary depending upon its local context. For example, if there are a group of pixels in the top of the slide which have a mean signal value of 100 and their local background has a mean value of 50 then this is a perfectly valid gene spot. However, in the context of another spot with a mean signal value of

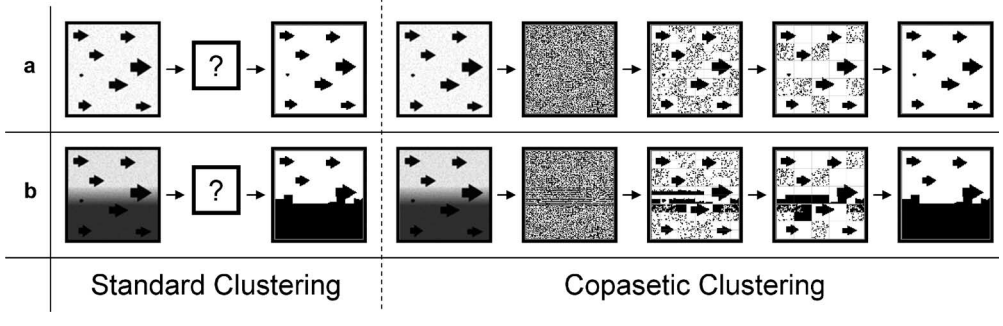

Fig. 6 *Results highlighting how clustering is performed locally and then propagated to neighbouring areas*
a Uniform noise example
b Realistic gradient noise example

5000 and a background of 4500, all the pixels of the first spot, both signal and noise, would be grouped with the background of the latter.

In Fig. 6*a* shows a comparison between the traditional clustering (left) and CC (right) techniques. Here, the intermediate steps have also been displayed showing the extra information that is available. The CC algorithm allows the processing of very large datasets, the drawback of which is that items that are spatially isolated are never compared to each other directly; instead, only their representatives will be compared. Normally such a bias would make the use of a technique like this questionable, but with microarray images we have a special case whereby it is actually beneficial.

This can be exploited by making use of 'historical data', essentially the intermediate stages taken from CC; Fig. 6*b* illustrates the point well. A conventional clustering method would take the input image which has a lot of background variation and cluster it into two groups with ease, as shown on the left. Almost 50% of the information in the original image was lost because a pixel which should be classified as noise in the bottom half is actually closer to the signal found in the top half of the image and as illustrated by the question mark; we have no idea how it determined this solution. CC suffers from the same limitations, producing an output with similar problems to that of a traditional clustering method. An important difference, however, is that we can restore this information from the previous layers. In the penultimate layer, all the arrows were clearly defined and could easily be used for further processing. It is this ability to jump inside the clustering routine that makes CC so flexible.

4 Results

In this Section, we compare the overall performance of the framework with that as determined by a commercially viable process as seen with the GenePix® package. First of all we will look at CA's success in discovering the structural composition of the slides, including overall block structure and gene spot locations. Then we will present a measure of accuracy for the entire process which will allow us to compare our automated system with that of an expert human operator.

It was initially envisaged that we would demonstrate the framework's capability using two sets of disparately structured microarray images with drastically varying quality (see Fig. 1). Ten images taken from previously analysed experiments were tested. Overall, CA successfully processed the block layout of all slides, with no prior knowledge of their structure (as shown by Fig. 7*a*). The CA process was able to successfully determine the underlying structure of the previously determined blocks, even when these blocks contained large artefacts (Fig. 7*b*) or partial gene spot information (Fig. 7*c*).

In order to quantify the performance capabilities of the automated CA framework against that of the human operator, a quality measure is also required which will allow the judgment of how well the calculated template fits the gene's spot position. Both techniques produce a mask that classifies the pixels as belonging to either signal (the gene spots) or noise (the local background). By overlaying the mask with the original image a metric can be utilised to quantify the disparity that exists between the two groups of pixels. If the masks fit the genes closely there will be high separation between these groups, and any misalignment between will lead to a diminished separation value.

There are many alternative metrics that can be used here. Typically, a preferred algorithm is the mean square error (MSE) which is defined as

$$MSE = \frac{\sum [f(i,j) - F(i,j)]^2}{N^2} \qquad (1)$$

where $f(i,j)$ represents the source or original imagery that contains $N \times N$ pixels and a mask image $F(i,j)$. Error metrics are computed on the luminance signal such that pixel values $f(i,j)$ range between black (0) and white (1). There are, for [0, 255] grey-scale images, two disadvantages of the MSE percentage as defined in (1). Firstly, the denominator is usually very large compared to the numerator, meaning that the improvement of the reconstruction process reduces this numerator value, but this might not be observable. Second, the MSE metric is sensitive to the brightness of the original image. Therefore, a more objective image quality measurement is known as the peak signal-to-noise ratio (PSNR) [18]. This metric is defined for $N \times N$ images with a [0, 1] or [0, 255] grey-scale range, in dB as

$$PSNR = 20 \log_{10}\left(\frac{1}{RMSE}\right) \qquad (2)$$

where the root mean squared error (RMSE) represents the norm of the difference between the original signal and the mask. The PSNR is the ratio of the mean squared difference between two images and the maximum mean squared difference that can exist between these. Therefore, the higher the PSNR value, the more accurately the mask fits the raw imagery. For all images present the proposed framework gave more accurate results.

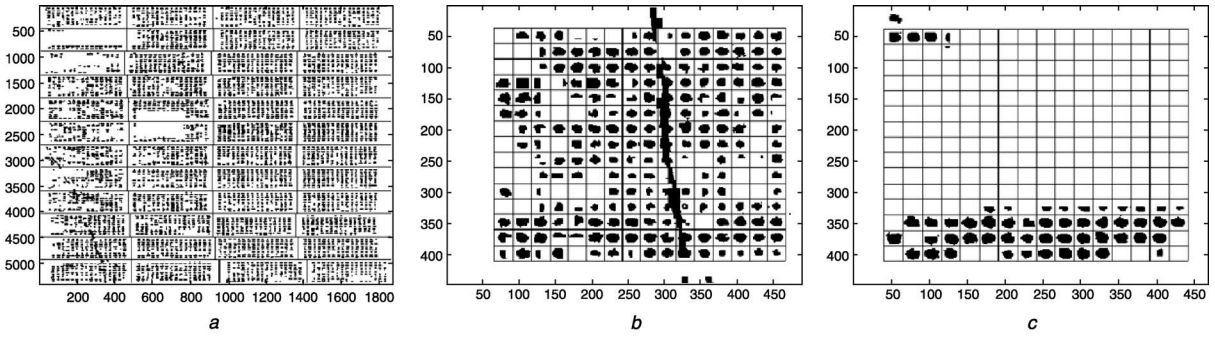

Fig. 7 *Examples of master block (a) and sub-block discovery (b, c)*

a Image Layout Example
b Image Structure Example 1, Row 11 × Col. 1 of *a*, *b* image structure Example 1, Row 11 × Col.1 of *a*
c Image Structure Example 2, Row 2 × Col. 1 of *a*, *c* image structure Example 2, Row 2 × Col. 1 of *a*

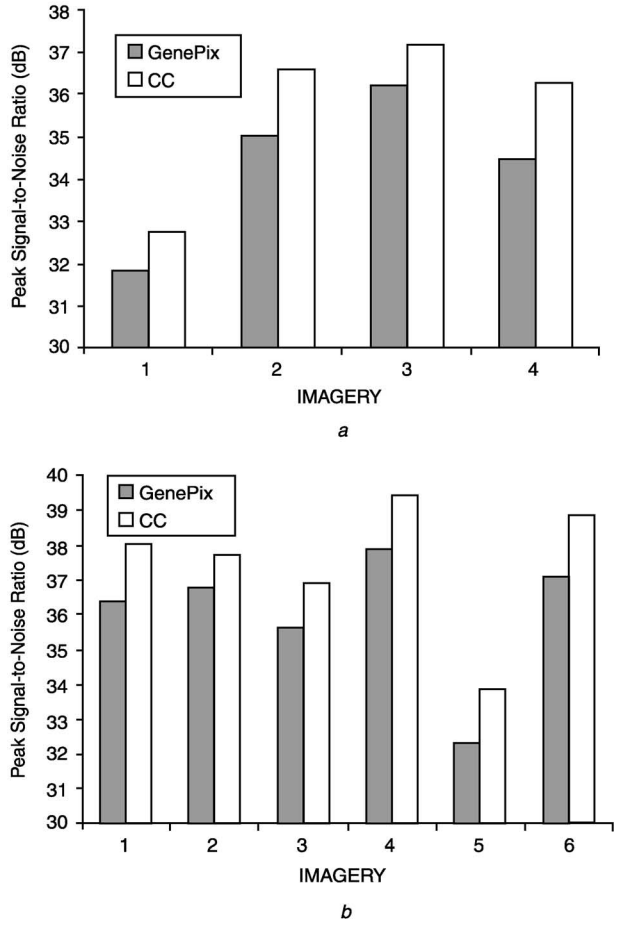

Fig. 8 *PSNR comparison between GenePix® and CA results*
a Image Set 1
b Image Set 2

From Fig. 8, we directly compare PSNR values determined by GenePix® and CA for the individual images and on average, CA has shown a marked 1 to 3 dB improvement. Essentially the CA process has consistently outperformed the human expert using GenePix® in terms of gene spot identification.

5 Conclusions and future work

We have presented a novel data-driven framework that attempts to improve the full workflow processing of microarray image analysis. Specifically, the framework consists of several components that process a microarray image from its raw 16-bit scanned representation to the final $\log_2$ ratios and related statistics without human intervention. *Copasetic Analysis* as detailed in Fig. 2 offers the following advantages over current implementations: *Copasetic Clustering* not only generates historical information allowing accurate image prediction, but also has the computational benefits of processing previously unfeasible datasets; *Image Layout* and *Image Structure* perform blind grid alignment on the imagery; *Spatial Binding* reconstructs the determined grid cell positions with accurate spot profiles; *Post Processing* corrects for the background noise and *Final Analysis* computes final microarray statistics. In the experiment Section of the paper we demonstrated the potential of CA using direct comparisons between our proposed approach and a commercially accepted process (GenePix®) over the dataset.

In future, we would like to focus on enhancing the current implementations of the framework's component parts. For example, the *Image Transformation Engine's* multi-view approach has proved to be beneficial in this initial testing; we are interested in exploring this component's potential in greater detail. Along with this, we intend to develop more sophisticated methods of slide structure reconstruction to further enhance the speed and reliability when processing particularly noisy slides. For example, Markovian analysis methods have previously been shown to produce good results when applied to discovering microarray structure [12]. However, they have a low tolerance to noise and common artefacts. A comparative study could be conducted, assessing the accuracy of these methods as individual components and the benefits when utilised in the CA framework. In this paper, we focused on two of the key components that make up the framework; it goes without saying that we plan to dedicate further publications to the study of each of the components. Finally, an important step will be the biological validation of these results; to this end we plan to analyse images containing control spots and a high number of biological repeats.

6 Acknowledgments

For kindly providing the datasets used, the authors would like to thank both Paul Kellam from the Dept. of Immunology and Molecular Pathology, University College London, and Dr Su Ling Li from the Institute for Cancer Genetics and Pharmacogenomics, Brunel University, London.

7 References

1 Moore, S.K.: 'Making chips', *IEEE Spectr.*, 2001, **38**, (3), pp. 54–60
2 Orengo, C.A., Jones, D.T., and Thorton, J.M.: 'Bioinformatics: genes, proteins & computers' (BIOS Scientific Publishers, Oxford, UK, 2003, 1st edn.), pp. 217–244
3 The Chipping Forcast II, Nature genetics supplement, 2002, pp. 461–552, http://www.nature.com/cgi-taf/dynapage.taf?file=/ng/journal/v32/n4s/index.html, accessed 20 May 2004
4 Hartelius, K., and Carstensen, J.M.: 'Bayesian grid matching', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, **25**, (2), pp. 162–173
5 Otsu, N.: 'A threshold selection method from grey level histograms', *IEEE Trans. Syst., Man. Cybern.*, 1978, **8**, pp. 62–66
6 Cheriet, M., Said, J.N., and Suen, C.Y.: 'A recursive thresholding technique for image segmentation', *IEEE Trans. Image Process.*, 1998, **7**, (6), pp. 918–920
7 Yang, Y.H., Buckley, M.J., Dudoit, S., and Speed, T.P.: 'Comparison of methods for image analysis on cDNA Microarray data', *J. Comput. Graphical Stat.*, 2002, **11**, pp. 108–136
8 Bozinov, D., and Rahnenfuhrer, J.: 'Unsupervised technique for robust target separation and analysis of DNA microarray spots through adaptive pixel clustering', *Bioinformatics*, 2002, **18**, (5), pp. 747–756
9 McQueen, J.: 'Some methods for classification and analysis of multivariate observations'. Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability (University of California Press, Berkeley, 1967), pp. 281–297
10 Bozinov, D.: 'Autonomous system for web based microarray image analysis', *IEEE Trans. Nanobiosci.*, 2003, **2**, (4), pp. 215–220
11 Wang, X.H., Istepanian, R.S.H., and Song, Y.H.: 'Application of wavelet modulus maxima in microarray spots recognition', *IEEE Trans. Nanobiosci.*, 2003, **2**, (4), pp. 190–192
12 Katzer, M., Kummert, F., and Sagerer, G.: 'Methods for automatic microarray image segmentation', *IEEE Trans. Nanobiosci.*, 2003, **2**, (4), pp. 202–214
13 O'Neill, P., Magoulas, G.D., and Liu, X.: 'Improved processing of microarray data using image reconstruction techniques', *IEEE Trans. Nanobiosci.*, 2003, **2**, (4), pp. 176–183
14 Berkhin, P.: 'Survey of clustering data mining techniques'. Accrue Software, San Jose, CA, 2002, www.accrue.com/products/rp_cluster_review.pdf, accessed 20 May 2004
15 Nagarajan, R., and Peterson, C.A.: 'Identifying genes in microarray images', *IEEE Trans. Nanobiosci.*, 2002, **1**, (2), pp. 78–84
16 Wayne Niblack: 'An Introduction to digital image processing' (Prentice-Hall, London, 1986)
17 Dunn, J.C.: 'A fuzzy relative of ISODATA process and its use in detecting compact well-separated clusters', *Cybern.*, 1974, **3**, (3), pp. 32–57
18 Netravali, A.N., and Haskell, B.G.: 'Digital Pictures: Representation, Compression and Standards' (Plenum Press, New York, NY, 1995, 2nd edn.)