BIOINFORMATICS ORIGINAL PAPER
Vol. 23 no. 8 2007, pages 966–971
doi:10.1093/bioinformatics/btm043

Gene expression

Microarray blob-defect removal improves array analysis

Jun S. Song[1,†], Kaveh Maghsoudi[1,2,†], Wei Li[1], Edward Fox[3], John Quackenbush[1,4] and X. Shirley Liu[1,*]

[1]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Department of Biostatistics, Harvard School of Public Health, [2]Harvard-MIT Division of Health Sciences and Technology, Harvard Medical School, [3]Microarray Core Facility and [4]Department of Cancer Biology, Dana-Farber Cancer Institute, Boston, MA, USA

Received on October 13, 2006; revised on January 21, 2007; accepted on February 3, 2007

Advance Access publication March 1, 2007

Associate Editor: Joaquin Dopazo

ABSTRACT

Motivation: New generation Affymetrix oligonucleotide microarrays often have blob-like image defects that will require investigators to either repeat their hybridization assays or analyze their data with the defects left in place. We investigated the effect of analyzing a spike-in experiment on Affymetrix ENCODE tiling arrays in the presence of simulated blobs covering between 1 and 9% of the array area. Using two different ChIP-chip tiling array analysis programs (Affymetrix tiling array software, TAS, and model-based analysis of tiling arrays, MAT), we found that even the smallest blob defects significantly decreased the sensitivity and increased the false discovery rate (FDR) of the spike-in target prediction.

Results: We introduced a new software tool, the microarray blob remover (MBR), which allows rapid visualization, detection and removal of various blob defects from the .CEL files of different types of Affymetrix microarrays. It is shown that using MBR significantly improves the sensitivity and FDR of a tiling array analysis compared to leaving the affected probes in the analysis.

Availability: The MBR software and the sample array .CEL files used in this article are available at:
http://liulab.dfci.harvard.edu/Software/MBR/MBR.htm

Contact: xsliu@jimmy.harvard.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Data quality control is an essential first step in microarray analyses. Recent work has focused on examining quality control issues, including spot filtering for spotted microarrays (Sauer *et al.*, 2005) and visualizing and quantifying regional bias for spotted and Affymetrix microarrays (Reimers and Weinstein, 2005). However, no method has been able to detect and correct for different sized single chip image artifacts, leading to improved analysis results.

In this article, we focus our analysis on widely used Affymetrix microarrays, for which new generation arrays often have 'blob-like' image defects. We define these as large, spatially contiguous clusters of signal from high intensity distributions, presumably resulting from extrinsic sources independent of transcription levels. These mostly oval-shaped defects, possibly caused by bubbles formed during array manufacturing, essentially render useless the transcriptional information in the affected area.

The Affymetrix Microarray Core Facility at the Dana-Farber Cancer Institute is one of the leading academic microarray cores in the world, and processes roughly 4000 arrays annually. Blob-like defects at the core have been rare for the more mature expression and 10/100K SNP arrays (roughly 1–2%, although they were more frequent when these arrays were first introduced to the market). However, for the newest 5 μm resolution arrays such as 500K SNP arrays, exon arrays and genome tiling arrays, our early estimates for blob frequency are at 10–20%. For human or mouse genome tiling array chipsets which tile the whole genome with seven arrays, in 90% of the time one out of the seven arrays in the chipset will contain a blob defect.

The traditional workflow of array data process starts at the microarray core facilities, where the Affymetrix GCOS software is used to convert each array image .DAT file to the data .CEL file (Fig. 1). When blob defects occur on any array, core facility staff can readily visualize them on the GCOS image display. However, the corresponding probes are often not identified as outlier probes by GCOS (more details in Section 3.1).

For a defective array with a blob occupying more than 10% of the array area, Affymetrix recommends repeating the assay and sometimes replaces the array for free. For arrays with less than 10% defect, the investigator faces the decision to either repeat the assay at their own expense or to analyze the data with the defect in place. In many cases the former will require repeating more than just the affected arrays, especially if the chipset contains multiple arrays. This is necessary not only to preserve the integrity of the entire set of samples and reduce batch effect but also because single arrays in a chipset are usually not commercially available. The latter option of simply ignoring the defect assumes that those covering less than 10% of the total area will not significantly affect experimental results. A justification for this assumption is that the probe layout on Affymetrix arrays is based on probe sequence and not genomic position. Thus, if the affected area is relatively small,

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

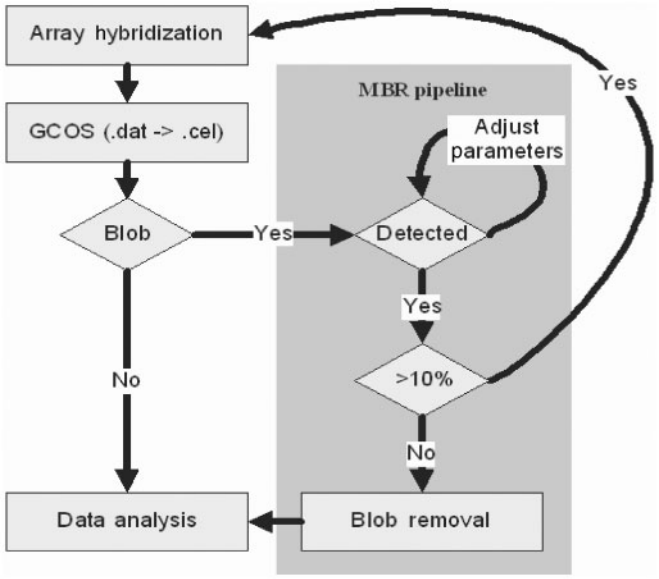

Fig. 1. Work flow for MBR usage: Following array hybridization, an Affymetrix core facility will scan the microarray and produce a .dat file. This .dat file is further processed by Affymetrix GCOS software to detect outlier probes (which are different from those produced by MBR) and create a .CEL file containing both the image intensity and an appended list of outlier probes. At this stage, MBR can use the .CEL file and blob defects may be noted by visual inspection. If there are no blobs, downstream analysis of the array may proceed as usual. If a defect is visualized, MBR can then be used to select the probes involved in the defect, adjusting detection threshold parameters until adequate overlap is achieved (default parameters do a good job for most blob defects). By using the histogram feature, the size and intensity distribution of blob defect is obtained. If the size of the blob is greater than 10%, of the array size, Affymetrix suggests performing the hybridization again on a new chip. However, we have shown that even for blob-defect sizes less than 10%, downstream analysis can be adversely affected. In these cases we recommend using MBR to remove the affected probes, and have shown the significantly better downstream analyses that result from this action.

the affected probes will sparsely map to diverse genomic regions with little final effect on the analysis results.

Here we investigate the consequences of using genome tiling arrays with blob defects and show that blobs even less than 10% in size may significantly affect the final results. We propose a simple and rapid solution, the microarray blob remover (MBR), which automatically detects and filters the affected probes from the data set. (A schematic diagram of MBR usage is summarized in Fig. 1.) We then analyze the remaining data in the absence of the defective probes and demonstrate that this provides a robust and vastly improved analysis result.

2 METHODS

2.1 Microarray blob remover (MBR)

MBR is a tool written in Java that displays the images of selected Affymetrix microarray .CEL files, detects blob defects and processes (removes) probes in those areas (Supplement 2.1).

2.1.1 Image display MBR constructs 3-byte images based on the probe intensities stored in binary .CEL files. The first two bytes interpolate between yellow and black colors, corresponding respectively to high and low probe intensities; and the last byte is used to add a blue hue to pixels identified as falling within detected blob regions. Because the dimensions of the images can be quite large (2450×450 probes in the newest generation arrays), we resize the images to 1/4 their original sizes and average the intensities of four neighboring probes to 1 pixel for visual display. Multiple images can be loaded and viewed using MBR provided the corresponding .CEL images have the same dimensions.

2.1.2 Blob-defect detection MBR adopts a two-step blob detection algorithm on the original .CEL image. Similar approach have been successfully used in 3D feature extraction in medical imaging (M.Albert *et al.*, sumitted for publication; Liu *et al.*, 2003; Tubic *et al.*, 2001). In the first step, MBR scans the chosen image with a 100×100 square, sliding in steps of 50 probes in both directions, and counts within each square the number of probes whose values are above the kth quantile of probe intensities. The default value of k is 90, but the 'Threshold for Blob Detection' slider allows user-defined thresholds between 60 and 100.

If the qualifying probes above the threshold cover more than half of the square, then MBR executes a second refining step. Otherwise, it stops in a few seconds without delineating any blobs. In the second step, MBR rescans the square with a circle of radius 20, sliding in steps of two probes. If more than p% of the probes in the circle have intensities above the $(k-5)$th quantile, then all probes inside the circle are flagged as being within blob regions and repainted in the display. The default value of p is 90 and can be adjusted between 80 and 100 using the 'Refinement Threshold' slider.

For arrays with user-discernable blobs on the MBR display, the user can adjust MBR parameters to ensure that the blob areas are correctly detected, although most often default parameters are sufficient for successful detection. MBR can process each array in a few seconds, with time dependent on array and blob sizes, and the user's computer power.

2.1.3 Blob-defect removal The Affymetrix GCOS algorithm can detect probes whose pixel intensities have high variances, and write the coordinates of these probes in an 'Outlier entries' section at the bottom of a .CEL file. These probes are often ignored by microarray analysis programs. MBR replaces the GCOS 'Outlier entries' section in the .CEL files with the locations of detected blobs. For arrays without visible blobs, MBR provides the option of eliminating the 'Outlier entries' section with the 'Remove Outliers' button. Following MBR processing, the .CEL files are then analyzed as required for the goals of the experiment and the particular array type used in the assay. The MBR processing of the .CEL files does not interfere with any analysis algorithm on Affymetrix microarrays. However, it is up to the downstream analysis algorithm to determine how to deal with probes detected as outliers.

2.2 Data

2.2.1 Examples with blob-defect Although most blob-defects in our experience are well-delineated oval or round regions, we used five heterogeneous examples to check whether MBR can work well with a variety of shapes and intensity profiles. They were arrays obtained from our own Affymetrix Microarray Core Facility at the Dana-Farber Cancer Institute, and included 1 expression, 1 SNP, 1 promoter tiling and 2 genome tiling arrays (Fig. 2).

2.2.2 ENCODE spike-in data The ENCODE consortium (The ENCODE Project Consortium, 2004) recently conducted a spike-in experiment to systematically compare different ChIP-chip protocols,

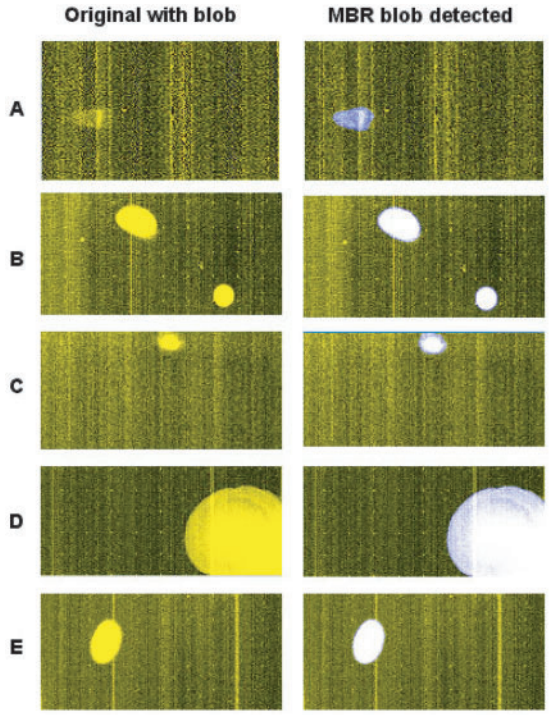

Fig. 2. MBR blob detection. *Left Column*: MBR's visualization tool shows various size and shape defects. *Right Column*: White–blue regions indicate area where MBR has automatically detected blob defects. *Rows*: (**A**) Expression array, (**B**) Genome tiling array, (**C**) SNP array, (**D**) Genome tiling array and (**E**) Promoter tiling array.

tiling microarray platforms and analysis methods (unpublished data). The spike-in samples (mock ChIP) are the mixture of the human genomic DNA and 96 ENCODE clones of ~500 bp, which are 2-, 4-, . . . , 256-fold enriched (12 clones at each concentration) in addition to the genomic DNA.

Genomic DNA without spike-in samples serve as the mock input controls. The samples were hybridized to different tiling array platforms including those from Affymetrix, NimbleGen and Agilent. Here, one spike-in and one input control sample on the Affymetrix ENCODE tiling arrays (GEO accession numbers GSM113413 and GSM113420, respectively) were randomly chosen for the simulation described below. Among the 96 spike-in regions, 10 were not tiled on the array due to repeat masking and 32 which overlapped with each other were merged. This left 70 unique and non-overlapping spike-in segments, ranging between 451 and 1476 bp in length. Although the spike-in concentrations relative to genomic DNA ranged from 2- to 256-fold, all spike-in segments were treated identically in the analysis.

2.2.3 Simulated data A typical blob on one array of the Affymetrix GeneChip® Human Tiling 2.0 Array Set was selected as a template blob to be superimposed on the spike-in ENCODE array. First, probe intensities on the Human Tiling 2.0 and ENCODE arrays were normalized by linear scaling to have the same mean. The ENCODE array was divided into nine equally sized squared regions, each containing a simulated blob of sizes 1–9%. The template blob had an oval shape and occupied 67 068 cells, which was about 1% the area of the Human Tiling 2.0 array and 4% of the ENCODE array. Therefore, for simulated blobs of sizes 1, 2 or 3% of the ENCODE array, the outer layers of the template blob were removed. For simulated blobs of size 5–9%, 30×30 squares within the template blob were randomly extracted and pasted to the ENCODE array. This generated a total of 81 (9 locations by 9 sizes) simulated arrays.

2.3 Microarray analysis

Each of the 81 simulated test arrays with and without blob removal was compared to the input control array to identify spike-in regions. To ensure the simulation results were not caused by analysis algorithm bias, two different tiling-array-analysis algorithms were used: Affymetrix tiling analysis software v1.1 (TAS) and model-based analysis for tiling arrays (MAT). TAS uses non-parametric quantile normalization and a Hodges-Lehmann estimator for fold enrichment (Affymetrix Tiling Array Software v1.1 Users Guide) (Cawley *et al.*, 2004) while MAT models baseline probe behavior from probe sequence and copy number in the genome (W.E. Johnson *et al.*, 2006).

3 RESULTS

3.1 Blob detection

For the range of blob defects on five real Affymetrix arrays, MBR was able to successfully delineate the blob on three arrays using default parameters. The larger, less uniform genome-tiling-array blob defect (Fig. 2D, defect size 9% of array area) required adjustment of the detection parameter k from 90 to 70% and the refinement parameter p from 90 to 85% for optimal image delineation. The more diffuse and low intensity expression array blob defect (Fig. 2A, defect size 2% of array area) was detected using k of 65% and p of 85%. With its current parameter ranges, MBR is not useful for large areas of faint smear or blobs smaller than 50×100 probes (almost 0.1% area of 5 µm resolution arrays). The former might require rehybridizing on a new array, and the latter might have minor impact on the analysis results.

For the five real arrays with blob defects, we also compared probes detected as outliers by the GCOS algorithm to those detected by MBR (Supplement 3.1). As a percent of the MBR generated outliers, the intersection of MBR with GCOS outliers ranged from just over 2% (for the 500K SNP, Fig. 2C) up to a maximum of just under 16% (for the expression array, Fig. 2A). Note that all the blobs on these five arrays were visually discernable (left column of Fig. 2 showing the array raw data with big yellow blobs) on the array image, and were successfully detected by MBR (right column of Fig. 2 showing MBR-detected blobs with white blue hue). This underscores the fact that GCOS outliers and MBR detect different probes, and that GCOS alone is not sufficient to detect probes in visually discernable blob defects.

For the 81 simulated ENCODE arrays with blob defects, where the blob regions are known ahead of time, MBR was able to identify and delineate the defect in all cases without any adjustment of the default parameters.

3.2 Tiling array analysis and spike-in detection

Both MAT and TAS were used to assess the downstream effect of analyzing tiling array data with relative blob sizes of 1–9%. Performance on detecting true signal from the ENCODE spike-in sample over the input control sample was based on measures of sensitivity (the number of correctly detected

regions divided by the total number of spike-in regions) and FDR (the number of falsely detected regions divided by the total number of detected regions). In ongoing efforts by ENCODE to assess analysis algorithms, correctly detected regions have been defined as those with any amount of overlap with the true regions. Here, however, we adopt the more stringent requirement that a correctly detected region must have at least 50% chromosomal coordinate overlap with the true spike-in region. Since there are nine samples of different blob locations for each blob size, the mean sensitivity and mean FDR are used as point estimates for each size group. Two-sided 95% confidence intervals are created for each mean.

The purpose of using both MAT and TAS was not to compare their relative performances, but rather to show that even different approaches to microarray standardization could be adversely affected by blob-like defects. As such, instead of using default parameters for MAT and TAS, we tried to ensure that the parameter values for a given algorithm yielded the best sensitivity and FDR measures for that algorithm. For each algorithm, the optimal *P*-value cutoffs that maintained FDR below 0.1 were determined by using an ROC-like curve (sensitivity versus FDR) for the spike-in array without blob defect (Supplement 3.2); these cutoffs were then used for spike-in arrays with blob defects.

3.2.1 Analysis with blob defect (prior to MBR use) Arrays with simulated blobs were first analyzed with both MAT and TAS. TAS sensitivities (Fig. 3A) appear to decrease while TAS FDR results (Fig. 3C) appear to increase with blob size. The MAT decrease in sensitivity (Fig. 3B) is less striking and the increase in FDR (Fig. 3D) is even milder across blob sizes. However, Figure 3 shows that for both MAT and TAS, there is a decrease in sensitivity and an increase in FDR for each blob size compared to the original non-defective array (blob size = 0%) sensitivity (MAT: 0.90, TAS: 0.414) and FDR (MAT: 0.033, TAS: 0.067). In fact, even for a defect occupying as little as 1% of array area, there is a statistically significant difference in sensitivity (MAT: 0.863 ± 0.007, TAS: 0.381 ± 0.015) and FDR (MAT: 0.306 ± 0.006, TAS: 0.137 ± 0.023) compared to the non-defective array sensitivities and FDRs (above).

3.2.2 Analysis after removing blob defect (after MBR use) Currently, MBR can be used with any software that addresses the outlier section of the Affymetrix .CEL data file. In the case of MAT, we implemented an algorithm to simply remove the outlier probes from further analysis. While other analysis software can readily take advantage of MBR by similarly dealing with outlier probes, currently they do not. Therefore we used only MAT to analyze all simulated arrays after MBR processing. Since MBR replaces the GCOS outlier section, we first verified that the regions detected by MAT did not vary significantly with and without GCOS outliers removed.

MBR removal of the defective probes from the simulated arrays slightly changes the point estimates for the sensitivities and FDRs across all defect sizes compared to those obtained from the original non-defective test array (Fig. 4A and 3B, dashed lines). However, these differences are not significantly different from either the non-defective array or from each other (sensitivity *P*-value = 0.157, FDR *P*-value = 0.233, Kruskal-Wallis Test). In contrast, for each given defect size, the sensitivity and FDR obtained after MBR blob removal are

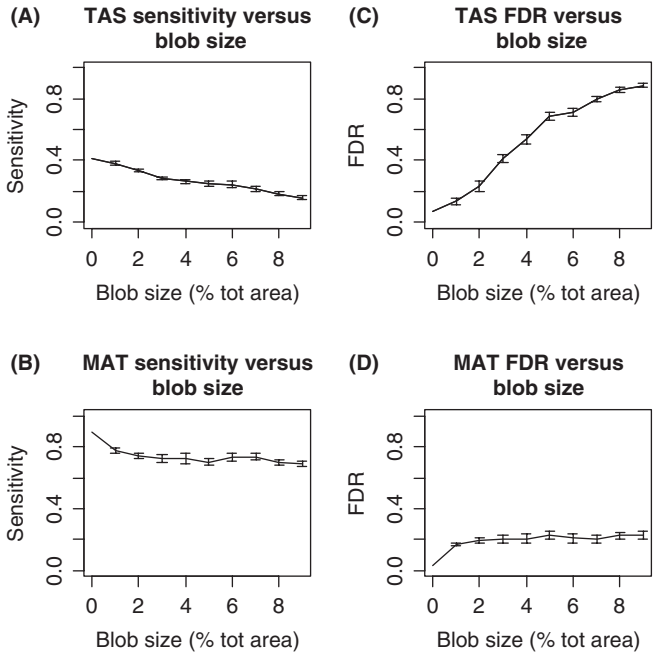

Fig. 3. The performance of TAS and MAT in detecting the spike-in regions with respect to the blob-defect size. (**A**) TAS and (**B**) MAT sensitivities; (**C**) TAS and (**D**) MAT FDRs. The error bars represent the two-sided 95% confidence interval of the mean estimates based on nine arrays at each simulated blob size with blob at different locations on the array. The sensitivity and FDR means obtained for the array with even the smallest size defect tested are significantly worst than those for the original array without defect.

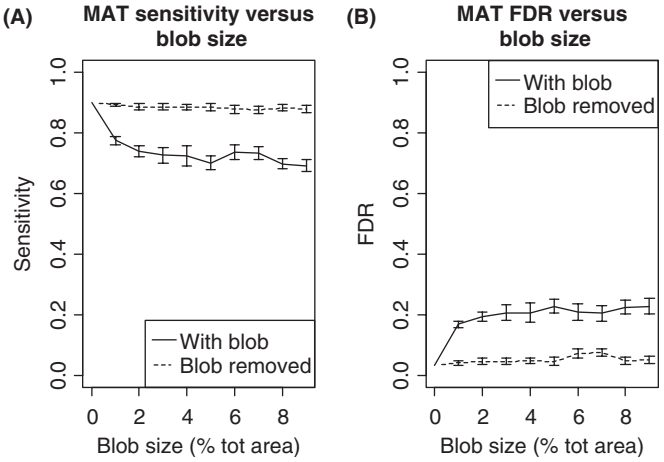

Fig. 4. MAT (**A**) sensitivity and (**B**) FDR at spike-in detection before and after blobs are detected and removed by MBR. For each blob size, the removal of the defect results in significantly improved values for both sensitivity and FDR.

significantly improved compared to those without using MBR (Fig. 4, *P*-value <0.0004). In fact, removal of even a 9% blob yields significantly improved results (sensitivity 0.925 ± 0.007, FDR 0.240 ± 0.010) compared to those (sensitivity 0.863 ± 0.007, FDR 0.306 ± 0.006) with a 1% blob left in place.

Another interesting observation is that for all blob sizes, the standard error for the sample mean estimates of sensitivity appear uniformly larger in the analysis without MBR correction compared to those with MBR processing (with the difference in standard error for a given blob size ranging from 0.0005 to 0.0082) (Fig. 4, see size of confidence intervals). The same holds for the FDR comparisons (with difference in standard error for a given blob size ranging from 0.0001 to 0.011), implying that detection properties (sensitivity and FDR) using MBR are more robust to varying array locations of the blob defect than those for which the defect is retained in the analysis.

4 CONCLUSIONS

We introduce an easy-to-use MBR tool that can rapidly and reliably detect and delineate blob-like defects ranging in size from 1 to 9% of microarray area using its default settings. Blobs of varying qualities and sizes can also be detected by adjusting the detection (k) and refinement (p) parameters. The usefulness of such a tool is underscored by analysis of simulated tiling array data. In our analysis, the sensitivity and FDR properties of both TAS and MAT were adversely affected by blobs of all sizes.

We believe the primary reason for this is that image defects can result in violations of distributional assumptions made by methods that are used to 'standardize' or 'normalize' array data so they can be compared to one another. For example one assumption of the widely used quantile normalization (which was used in TAS) is that the distributions of signal in all chips are similar. The addition of substantial amounts of high-intensity defect signal on one chip but not the others will violate this assumption (Fig. 5), leading to contaminated signal in all of the arrays, and subsequent spurious findings. Similarly, sequence-based standardization methods such as MAT can also be sensitive to regional defects of Affymetrix arrays. MAT standardizes the probe signal by checking probes with similar sequences. Since probes with similar sequences are often neighbors on Affymetrix arrays, a blob also adversely affect the standardization of neighboring probes in MAT. For these reasons, simply restoring the shape of the original distribution in a non-biased fashion by removing the affected probes helps improve array standardization and subsequent downstream tiling analysis results.

In support, our results suggest that analyzing arrays with defects even smaller than the 10% cutoff is at best suboptimal and has the potential to lead to spurious findings. Rather than

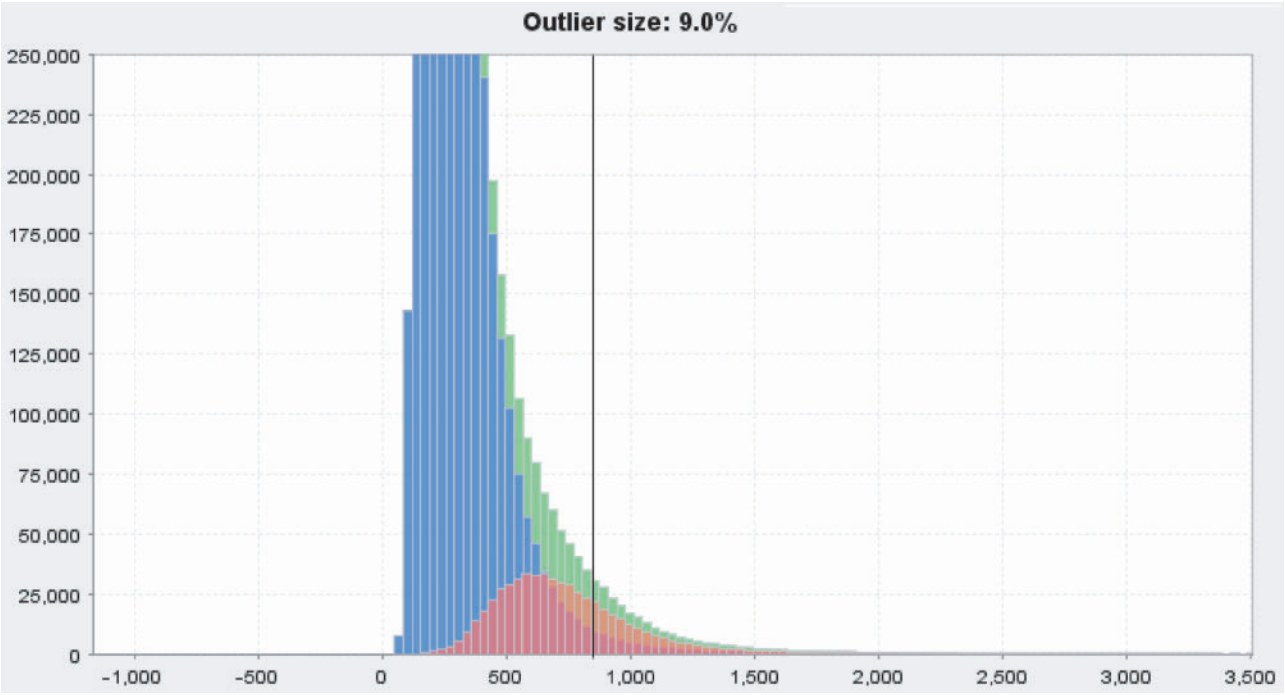

Fig. 5. MBR-generated histograms show three different distributions of probe signal intensity on a microarray: green for the overall array distribution including the blob defect, red for the blob-defect distribution and blue for the array after blob-defect removal (green minus red distributions). The *x*-axis corresponds to the raw signal intensity of a probe and the *y*-axis corresponds to the frequency of that intensity. The histogram is scaled to best show the blob-defect distribution and hence the top is not shown. A vertical line through the histogram indicates the 90th percentile of the original green distribution. The overall distribution of an array can be influenced by blob defects of different size and intensity distribution. This particular histogram example comes from defect D from Fig. 2.

repeating assays, which can be both costly and time-consuming, we tested the fast and simple strategy of removing the affected probes from the analysis. MAT analysis of tiling arrays from which blobs were removed demonstrated that this method is robust to defect size (from 1 to 9% area) and yields results far superior to including the bad probes in the analysis. We expect similar improvements with other downstream analysis software once MBR is implemented.

At present, MBR is not intended to be used in an entirely automated manner in a high-throughput setting. The concern of potential false positive and false negative detection is common for all image-processing algorithms, although theoretically the probability that MBR will falsely detect the smallest blob using the least stringent parameters is negligible (probability $\sim 10^{-55}$) (Supplement 4). Instead, the ideal use of MBR is at microarray core facilities to be coupled with GCOS (Fig. 1). If core technicians see blobs on the GCOS display after a .DAT file is converted into .CEL file (which often takes a few minutes), they could run MBR and remove the defective probes (which takes a few seconds). Since visualization first prompted the decision to use MBR, the parameters of MBR can be manipulated to optimize the encapsulation of the defect; however, in our experience, the default parameters do a good job for most blob defects.

ACKNOWLEGEMENTS

We would like to acknowledge Pamela J. Hollasch and Maura A. Berkeley at the DFCI Microarray Core facility for providing the background information and sample data for this manuscript.

Conflict of Interest: none declared.

REFERENCES

The ENCODE Project Consortium (2004). The ENCODE (ENCyclopedia Of DNA Elements) Project *Science*, **306**, 636–640.

Albert,M. *et al*. (Submitted for Publication) Defining the Rectal Dose Constraint for Permanent Radioactive Seed Implantation of the Prostate.

Cawley,S. *et al*. Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell*, **116**, 499–509.

Johnson,W.E. *et al*. (2006) Model-based analysis of tiling-arrays for ChIP-chip. *Proc. Natl. Acad. Sci. USA*, (In press).

Liu,H. *et al*. (2003) Automatic localization of im-planted seeds from post-implant CT images. *Phys. Med. Biol.*, **48**, 1191–203.

Reimers,M. and Weinstein,J.N. (2005) Quality assessment of microarrays: visualization of spatial artifacts and quantitation of regional biases. *BMC Bioinformatics*, **6**, 166.

Sauer,U. *et al*. (2005) Quick and simple: quality control of microarray data. *Bioinformatics*, **21**, 1572–1578.

Tubic,D. *et al*. (2001) Automated seed detection and three-dimensional reconstruction. I. Seed localization from fluoroscopic images or radiographs. *Med. Phys.*, **28**, 2265–2271.