BMC Bioinformatics

Methodology article

Hybrid clustering for microarray image analysis combining intensity and shape features

Jörg Rahnenführer*[1] and Daniel Bozinov[2]

Address: [1]Max-Planck-Institute for Informatics, Stuhlsatzenhausweg 85, D-66123 Saarbrücken, Germany and [2]Center for Human Molecular Genetics, Nebraska Medical Center, Omaha, NE 68198, USA

Email: Jörg Rahnenführer* - rahnenfj@mpi-sb.mpg.de; Daniel Bozinov - dbozinov@unmc.edu

* Corresponding author

Published: 29 April 2004

BMC Bioinformatics 2004, **5**:47

Received: 29 October 2003
Accepted: 29 April 2004

This article is available from: http://www.biomedcentral.com/1471-2105/5/47

Abstract

Background: Image analysis is the first crucial step to obtain reliable results from microarray experiments. First, areas in the image belonging to single spots have to be identified. Then, those target areas have to be partitioned into foreground and background. Finally, two scalar values for the intensities have to be extracted. These goals have been tackled either by spot shape methods or intensity histogram methods, but it would be desirable to have hybrid algorithms which combine the advantages of both approaches.

Results: A new robust and adaptive histogram type method is pixel clustering, which has been successfully applied for detecting and quantifying microarray spots. This paper demonstrates how the spot shape can be effectively integrated in this approach. Based on the clustering results, a bivalence mask is constructed. It estimates the expected spot shape and is used to filter the data, improving the results of the cluster algorithm. The quality measure 'stability' is defined and evaluated on a real data set. The improved clustering method is compared with the established Spot software on a data set with replicates.

Conclusion: The new method presents a successful hybrid microarray image analysis solution. It incorporates both shape and histogram features and is specifically adapted to deal with typical microarray image characteristics. As a consequence of the filtering step pixels are divided into three groups, namely foreground, background and deletions. This allows a separate treatment of artifacts and their elimination from the further analysis.

Background

In DNA microarray experiments, genetic probes with known identity are affixed to a glass slide or another substrate at discrete spots. The probes are prepared for binding with cDNA or mRNA samples. Typically, the genetic composition of two such samples is compared. The two samples are labeled with red-fluorescent and green-fluorescent dye, respectively, mixed and competitively hybridized to the microarray containing the complementary probes. For early references of this technology see Schena et al. [1] and Shalon et al. [2].

Using a laser scanner, TIFF images of the microarray are obtained. The relative abundance of one or the other sample is represented by a red or green signal at the spot location. The two major objectives of microarray image analysis are therefore to find the discrete spot locations and to quantify the spot intensities. Many available tools provide algorithms to solve these problems; among these,

GenePix (Axon Instruments [3]), Imagene (Biodiscovery, Inc. [4]), QuantArray (GSI Lumonics [5]) and ScanAlyze (Eisen [6]) are widely used. Most methods assume circular spot shapes and require manual alignment of the grid locations. Therefore, automated grid and spot finding as well as robust intensity quantification are highly desirable. For oligonucleotide fingerprinting and complex hybridizations, automated array processing has for example been presented by Steinfath et al. [7]. The intensities are calculated based on a normal distribution model for every single spot. In cDNA microarray images, the assumption of a circular spot shape is usually not justifiable due to artifacts caused by the printing process and the hybridization technique. Generally, two main concepts dealing with this obstacle have been presented, namely pixel intensity histogram methods and shape detection methods. Histogram methods are widely used, for example by Imagene or QuantArray, see Chen et al. [8] for an early reference. The first effective *shape* method is implemented in Spot (Buckley [9]) and uses the 'seeded region growing' algorithm (Adams and Bischof [10]), see also Yang et al. [11] for further details. The Spot software is currently one of the most competitive software tools for microarray image analysis, as it successfully deals with different spot shapes and artifacts.

A new robust method is pixel clustering. The basic algorithm, introduced in Bozinov and Rah-nenführer [12], was developed for spot identification and intensity quantification. Subsequently, the clustering approach was extended for grid finding. This idea was first tested in Bozinov et al. [13], where it was applied to a single high-quality array. The *Gridclus* algorithm described in this paper represents an improvement that leads to satisfying results also for low-quality arrays.

The original pixel clustering algorithm is an adaptive histogram method without direct attention towards the spot shape. In the present paper, we describe the further development and improvements of this algorithm. The detailed algorithms are given in the Methods section. The two main features of the new approach are *repeated clustering* and *mask matching*. *Repeated clustering* is applied, if the clustering selects only very few pixels as foreground region, mainly in case of low foreground intensity and small bright artifacts. In such a case the outlier pixels are removed and clustering is repeated on the background pixels, until at least m pixels (e.g. $m = 50$) constitute for the spot foreground area. *Mask matching* integrates the spot shape into the algorithm. Based on the cluster results of all spots, a bivalence mask is constructed to estimate the average spot shape. The mask is used as a template to filter out low-quality parts of single microarray spots. This yields a genuine combination of the two central features 'histogram' and 'shape' and thus a favorable hybrid image analysis solution. An advantage of this method is the partitioning of pixels into foreground, background and discarded pixels. This allows the elimination of artifacts from the further analysis.

In the Methods section below, first the pixel clustering approaches for grid and spot detection are reviewed. Then the hybrid approach and the associated quality measure 'stability' are introduced. The stability is the proportion of pixels in the foreground area that are not deleted by the mask matching step. The complete spot detection algorithm is called HYBRID PX_{KMEANS}.

In the next chapter, we report on the application of the new algorithms to real microarray images. The ability to produce reliable values in an experiment with replicates is compared, between the new hybrid solution and one of the most successful alternatives, the Spot software (Buckley [9]). The results show the competitiveness of the pixel clustering approach.

Results and discussion

HYBRID PX_{KMEANS} was applied to a real microarray image from the Microarray Core Facility of the University of Nebraska Medical Center. The array consisted of 24 rows and 36 columns of gene spots, in total 864 spots. With *Gridclus* the target areas could be perfectly identified.

Multiple clustering

Figure 1 proves the positive effect of the repetitive clustering in step 3 of HYBRID PX_{KMEANS}. The number of necessary pixels in the foreground was set to $m = 50$. The grey histogram bars capture the sizes of the foreground areas after single clustering, the shaded histogram bars after repeated clustering.

After single clustering, the foreground area size was smaller than 25 pixels for 118 spots, approximately 14% of all 864 spots. For all others spots the area was bigger than 50 pixels. Clearly, in the 118 problematic cases, the algorithm identified just very few outlier pixels as foreground. After repeated clustering, all area sizes were bigger than 50 pixels, by construction of the algorithm. It is striking, that the distribution of the sizes of these 118 reorganized spot areas resembles the distribution of the spot areas identified after the first clustering. For 115 out of those 118 spots, the algorithm terminated immediately after the second clustering; for the other 3 spots, after the third clustering. These observations indicate, that the spot shape is already detected after the second clustering, and the foreground area size is not just continually increased by some random pixels. More evidence for this hypothesis is obtained by the distribution of the stability values. This will be explained in more detail below. The choice of the number m is not critical, since the only purpose of this

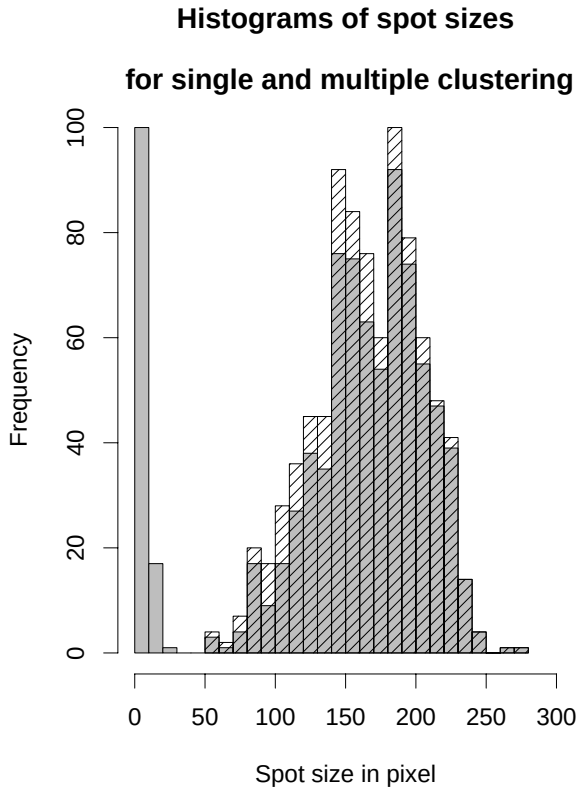

Figure 1
Distribution of spot sizes. Distribution of foreground area sizes after single clustering (grey) and after repeated clustering (shaded), applied to a microarray slide with 864 spots.

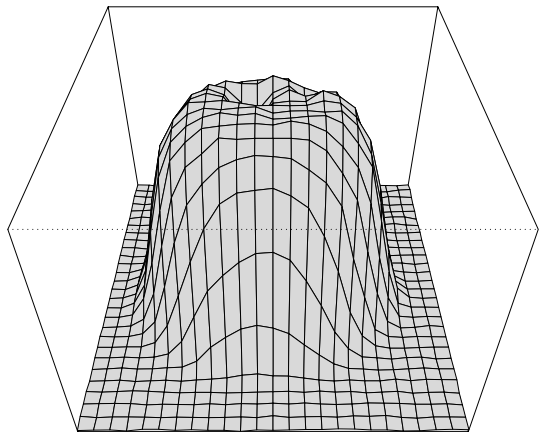

Figure 2
Accumulated foreground assignments of pixels. Plot of assignments of pixels to the foreground area, according to the clustering results, accumulated separately for every pixel over all 864 spots.

step is to eliminate the influence of bright pixels belonging to small artifacts. Such pixels often constitute for the whole foreground area after initial clustering. In the example presented, any number between 25 and 50 would have led to identical results.

Mask matching

After single clustering of all spots in the present image, the number of foreground assignments was determined for every pixel as described in step 4 of HYBRID PX_{KMEANS}. The average of these numbers over pixels was $\bar{f}$ = 146.98. According to step 4 of HYBRID PX_{KMEANS}, every pixel with $f > \bar{f}$ is assigned to foreground in the bivalence mask. In other words, if in more than 147 out of 864 spots a pixel at a fixed position belonged to foreground, it was also set to foreground in the bivalence mask. The other pixels were assigned to background.

Figure 2 shows the number of foreground assignments for all pixels, summed up over spots. One can see, that the accumulated information accurately defines an average spot. Figure 3 shows the bivalence mask that represents the average spot shape. Foreground pixels are plotted as white squares and background pixels are plotted as black squares. Although most single spots didn't have a nice circular shape, the aggregation of information led to an almost perfect circle.

Figure 4 shows the number of deletions due to mask matching, summed up over all spots. A pixel was deleted if it was assigned to foreground in the spot and to background in the mask, or vice versa. In the background area of the mask, only few deletions occurred, whereas in the foreground area of the mask, deletions were more and more likely for pixels closer to the margin. In the center of the spot, the deletion frequency was also increased. This reflects that some spots had a so-called doughnut shape with significantly lower intensity in the center. This artifact appears when the microarray pin detaches improperly during the spotting process, and not enough probe material is attached to the microarray slide. As desired, such areas were deleted by mask matching and not considered as background.

Figure 5 shows the 'stability' values of all 864 spots. The sizes of the foreground areas before mask matching are plotted against single 'stabilities', the relative numbers of deletions due to the mask. The 'median stability' over all

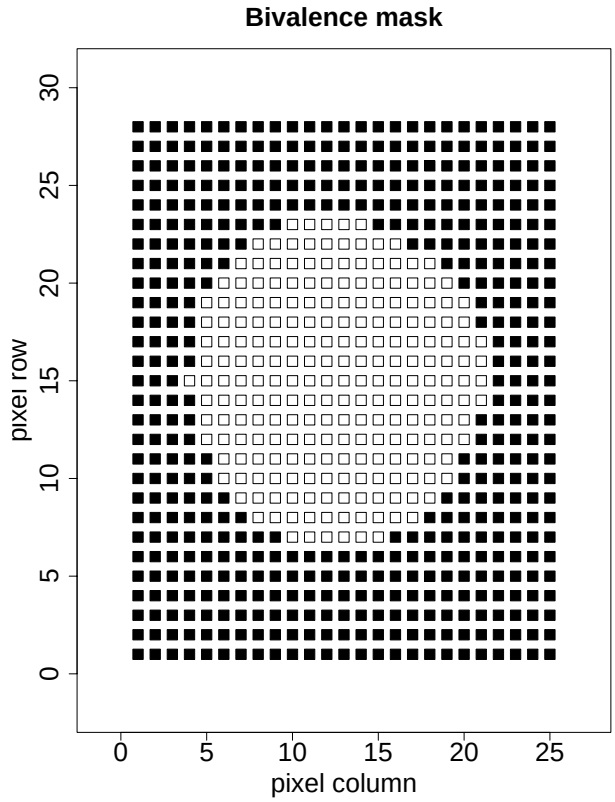

Figure 3
Bivalence mask. Typical bivalence mask for a real microarray image.

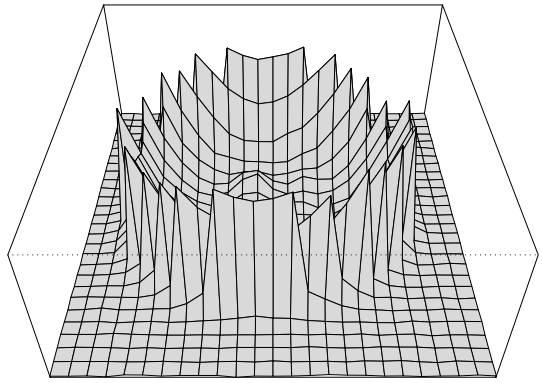

Figure 4
Accumulated pixel deletions. Plot of pixel deletions caused by mask matching, accumulated separately for every pixel over all 864 spots.

spots was 0.945, very close to 1. The lowest 'stability' was 0.5. Apparently, spots with smaller area sizes were more jeopardized of having a serious relative number of deletions, check for example the 13 spots with critical 'stability' values below 0.7.

In Figure 5, the points belonging to spots 402 and 788 are encircled (left 402, right 788). Spot 402 is a rather large green spot with a big yellow artifact on top of it. Figure 6 shows the target area of that spot, as a respective cut out of the original image. Due to mask matching, 42 of 227 foreground pixels were deleted. This included the whole artifact, see Figure 7 for the bivalence plot after mask matching. Here again, pixels in foreground and background are plotted as white and black squares, respectively. The deletion of the huge artifact was desirable, but decreased the 'stability' to 0.815. Only 59 other spots had even lower values. Out of all spots with larger foreground area size before mask matching, only spot 788 had a smaller 'stability'. Figure 8 shows the bivalence plot. Obviously the identified spot area was large compared to the bivalence mask. The analysis of the two spots demonstrates the success of the algorithm. It also makes clear that a 'stability' value around 0.8 is not critical, if the foreground area size is rather large.

In Figures 7 and 8 we also see that parts of the deleted areas resemble almost ring-shaped regions between foreground and background. This is a designed property of many other microarray image analysis tools. Here, it is an unforced consequence of the algorithm, which adds to the confidence in the strength of this approach.

Comparative study

On a real microarray image with 864 genes and 2 replicates for every gene (1728 spots in total, Figure 9 shows the top left corner of the image), the extended pixel clustering approach was compared to the Spot image analysis software (Buckley [9]). The algorithms implemented in Spot are based on a background estimation method called 'morphological opening' and on 'seeded region growing' by Adams and Bischof [10], for details see Buckley [9] and Yang et al. [11].

From the image analysis, two final intensity measurements for red and green (R and G) are obtained for every single spot. In an MA plot, the log ratio $M = \log_2(R/G)$ is plotted against the average log intensity $A = \log_2(RG)/2$. In the present study, two replicates are available for every transcript. The reproducibility of values is used to judge

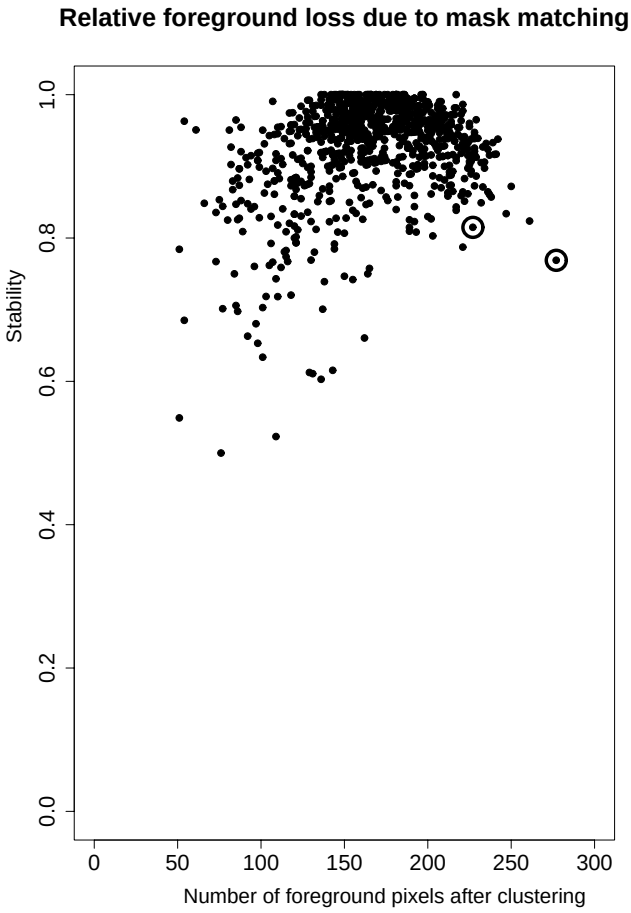

Figure 5
Distribution of 'stability' values. 'Stability' values for a microarray with 864 spots. For every spot, the foreground area size is plotted against the 'stability'. The points that belong to spots 402 and 788 are encircled.

Figure 6
Spot with large artifact on top. Green spot with a large yellow artifact on top that impairs results of the image analysis.

the performance of the image analysis algorithms. Figure 10 shows MA plots obtained with HYBRID PX_{KMEANS} (top) and with Spot (bottom), where the two measurements for every gene are connected by a line. Short lines indicate good stability of the procedure. A short horizontal distance stands for small intensity differences between the two replicates, and a short vertical distance means that both replicates produce similar estimates for the log ratio *M*.

It turns out that both methods lead to a satisfying reproducibility for *M* (log ratio) estimates, but for *A* (log intensity) estimates the differences are huge, at least for a small portion of the genes. This is partly due to the bad quality of the microarray image that was chosen on purpose to challenge both algorithms. Other reasons are biological and experimental variance, and last not least the quality of the algorithms. To directly compare both approaches, we calculated the absolute difference between the two *M* values, since the log ratio *M* is the most interesting measurement for investigators analyzing microarray data. The results are shown in Figure 11. Both for HYBRID PX_{KMEANS} (top) and for Spot (bottom), the absolute differences are plotted on a $\log_2$-scale against the intensities *A*. For both data sets, a lowess fit is calculated. This fit is a scatter-plot smoother that is popular also for microarray data normalization. The two lowess fits are plotted together in Figure 12 for better comparability. The solid line belongs to pixel clustering and the dashed line to Spot.

Figure 12 underlines the differences of the two approaches. The Spot algorithm produces only values with higher intensities, due to the treatment of the background, whereas HYBRID PX_{KMEANS} leads to intensity estimates in the whole potential intensity range. This is an advantage of the pixel clustering method, since for the low-quality array used for the comparison, the true expression of many genes is in fact rather low. The estimated relative expression of such genes is regressed towards 1 by the Spot software. The clustering method thus leads to a finer resolution for low expression values, whereas Spot is less sensitive to systematic errors. Naturally, in the low intensity range we observe a worse

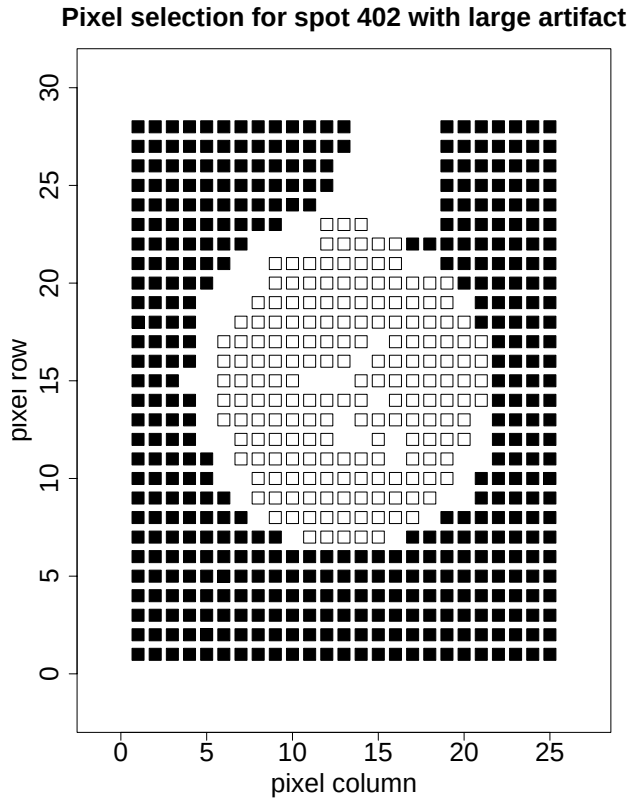

Figure 7
Classification result for spot 402. Selected foreground (white) and background (black) pixels for green spot 402 with a large artifact on top.

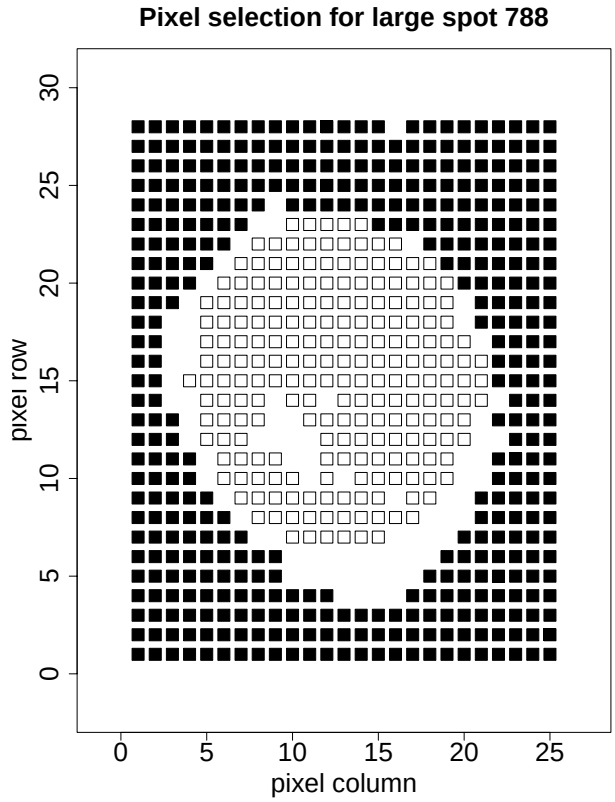

Figure 8
Classification result for spot 788. Selected foreground (white) and background (black) pixels for the relatively large spot 788.

reproducibility for the clustering method, see Figure 11. For intermediate intensities (in the range 8–10), the reproducibility is extremely similar for both algorithms (see Figure 12), and for high intensities, pixel clustering performs better. One has to take into account the smaller number of genes in this range, though.

To summarize, both algorithms create similar reproducibility values, which underlines the competitiveness of the new HYBRID PX_{KMEANS} algorithm. One problem of pixel clustering should be discussed, though. In some cases, the deletion step through mask matching leads to spots with only very few pixels in the foreground area. This is the case for example, if a large artifact in the background contains most of the high-intensity pixels. This also is the reason for the four long lines in the bottom right corner of Figure 10. In all cases, for one of the two replicates the problem mentioned above appears and the spots should be flagged.

Finally, we stress the improvement of the algorithm in comparison with the original simple pixel clustering approach introduced in Bozinov and Rahnenführer [12]. Without the extensions described in the present paper, about 20% of the spots in the low-quality array have to be flagged, because less then 5 pixels constitute for the foreground area. Again, this is caused by small artifacts with a very high intensity. The repetitive clustering eliminates this problem. The mask matching step is a good control, if now correct parts of the spot target area are assigned to foreground and background. For the low-quality array, only four spots are left to be flagged. This demonstrates the usefulness and the suitability of the proposed modifications.

Conclusions

The proposed HYBRID PX_{KMEANS} algorithm represents a true hybrid microarray image analysis solution. Originally being a pure histogram method, the important shape aspect is included through the mask matching step. We

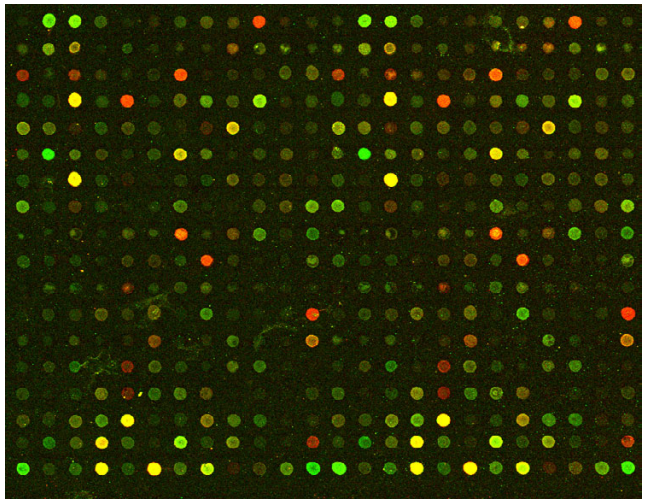

Figure 9
Microarray image of low quality. Microarray image with replicated spots. The whole array consists of 4 blocks, the Figure shows the top left block with 216 replicated spots that are arranged in two blocks next to each other. The expression values on the left side and on the right side should thus coincide.

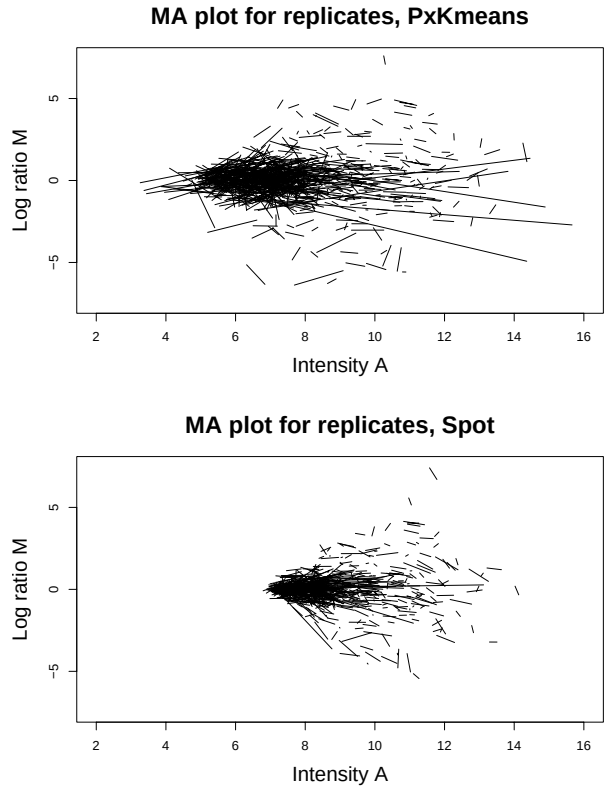

Figure 10
MA plots. MA plots for HYBRID PX_{KMEANS} (top) and Spot (bottom), measurements of two replicates are connected by lines. In both plots, for every spot the mean of the measurements for red and green intensity is plotted against their ratio, on a log2-scale.

use indicator values of the clustering results instead of original intensity values to construct the bivalence mask. This is favorable, since then all microarray spots weigh equally. Single low-quality spots with high intensities cannot impair the results, and many spots with low intensities add valuable information.

Using a high quality array, the 'stability' values provide a good check of the ability of the algorithm to identify spot areas. The median stability of 0.945 for our real data set demonstrates the feasibility of the algorithm. With this knowledge, we actually use the 'stability' as a reliable quality measure for the microarray image itself, both for single spots and for the whole array. Spots with a low 'stability' are flagged. Another criterion to flag spots could be a small absolute foreground area size relative to the bivalence mask.

On a low-quality microarray image with replicates, the new algorithm is compared to the Spot software. The results show the competitiveness of the extended pixel clustering algorithm with established methods. Like other approaches, this one can easily be generalized for different types of arrays, for example with non-competitive hybridizations. The only adjustment then, is to simply apply the cluster algorithm to the one-dimensional values.

The software used for the analyses in this paper is available upon request from the authors. The program was written in Java and thoroughly tested on Windows.

Methods

Clustering microarray images

A fully automated microarray image analysis consists of spot target area determination, partitioning of these areas into foreground and background, and intensity extraction. We first describe the *Gridclus* algorithm that correctly identified grids in a variety of real microarray images. Then, the original pixel clustering algorithm for spot finding and quantification is described, as a basis for its extension, the new hybrid approach.

Grid clustering

Two two-dimensional arrays of intensity values are given, one for the red channel and one for the green channel. For

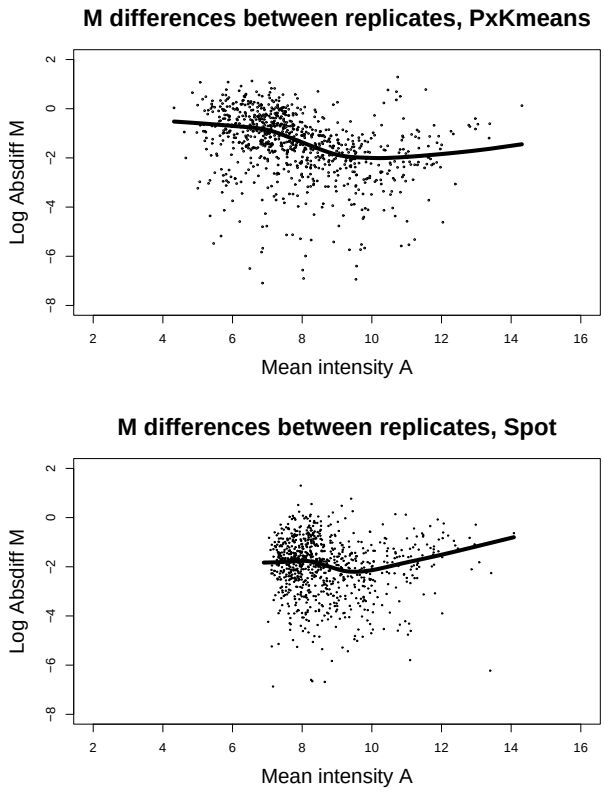

Figure 11
Comparative plot for differences of M values for repeated measurements. Plot of mean intensities against absolute differences of log-ratios between two replicates, on a log2-scale.

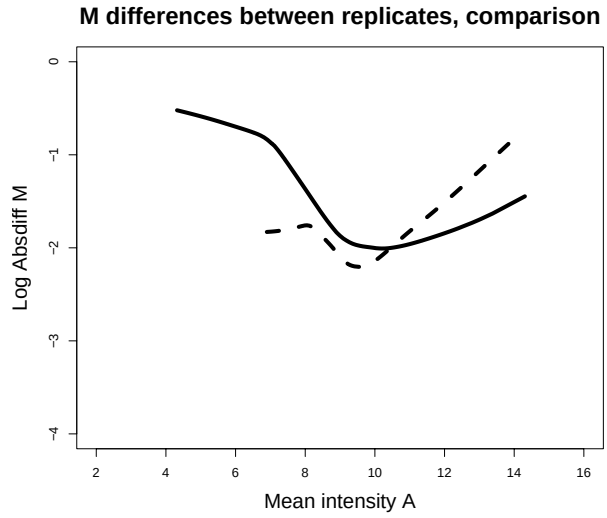

Figure 12
Lowess fits to scatter-plots from Figure 11. A local linear scatter-plot smoother (lowess) is applied to the data from Figure 11.

a pixel in row i_1 and column i_2, let $R(i_1, i_2)$ and $G(i_1, i_2)$ denote the intensity values for the red and the green channel, with $i_1 = 1..N_1$ and $i_2 = 1..N_2$. Here, N_1 and N_2 denote the total number of pixel rows and columns, respectively, on the entire microarray.

Definition: GRIDCLUS

Iterative clustering of all pixels into foreground and background

Define $m_0 := (\min(R), \min(G))$ and $m_1 := (\max(R), \max(G))$. Apply the k-means cluster algorithm with $k = 2$ to the $N_1 \cdot N_2$ two-dimensional values $(R(i_1, i_2), G(i_1, i_2))$. Choose m_0 and m_1 as starting points for k-means. The output is an indicator function $I(i_1, i_2)$ that assigns a cluster membership value to every pixel: 1 (for foreground) to pixels in the cluster with the larger average intensity and 0 (for background) to pixels in the other cluster. Calculate the fraction of pixels f in the foreground by dividing the number of foreground pixels ($I(i_1, i_2) = 1$) by the total number of pixels $N_1 \cdot N_2$. As long as f is smaller than a prefixed value (e.g. 0.2), repeat clustering the group of background pixels from the previous step into foreground and background pixels and reassign the former background pixels, increasing f in every step.

Assignment of pixel rows and columns to foreground or background

For every pixel row or column l, calculate the corresponding fraction of foreground pixels f_l. Apply smoothing to the fractions f_l with a fixed window size (e.g. $w = 7$), i.e. calculate averages of w contiguous fractions. Then, assign a line to foreground, if $f_l > f$, and to background, if $f_l \leq f$. The output are two binary vectors, one for columns and one for rows, that assign all lines to foreground or to background.

Calculation of target area separation lines

Both for rows and columns, determine intervals of contiguous background lines. Calculate means of those intervals as spot target area separation lines and adjust leftmost and rightmost separation line based on median separation line distance.

Determination of fixed size spot target areas

Between each pair of contiguous target area separation lines, determine the maximum of the smoothed fractions f_l. This is the estimated midpoint of a spot row or column, respectively. Choose a fixed size area (e.g. 25 × 25 pixels) around the estimated midpoints as final spot target areas.

In some cases, a further refinement is necessary. Microarray grids are often slightly rotated. Thus, out of a set of potential rotation angles one is chosen that maximizes an optimization criterion for best separation into foreground and background lines. The choice of the smoothing constant $w = 7$ is rather conservative and guaranteed for all tested examples, that contiguous sequences for the foreground and background regions are found. For high-quality arrays, a value of $w = 3$, in some cases even no smoothing can produce the same results. The fixed size area of 25^2 pixels has to be adjusted, if the true average spot size is significantly different. For the clustering algorithms it is most convenient to choose an area that consists of at least twice as many points as belong to the true average spot size.

The feasibility of a basic version of *Gridclus* was demonstrated on one real high-quality microarray image tiff-file (Bozinov et al. [13]). In this old version, there was no iteration in step 1. The iteration is needed only in the presence of few high-intensity outlier pixels to avoid an extreme unbalanced segmentation into foreground and background. Thus the specific value 0.2 for the minimum portion of foreground pixels is not critical and can be used for every array. Furthermore, step 4 was not present in the old version. The centralization is essential for the mask matching step in the hybrid spot clustering algorithm described below. The extensions in *Gridclus* allow the application to microarray images of a wider quality range.

Spot clustering

The basic spot clustering approach uses the k-means algorithm. A more suitable version for typical impediments encountered in microarray spots was developed by Bozinov and Rahnenführer [12] using the algorithm PAM (Partitioning around medoids, introduced by Kaufman and Rousseeuw [14]) with a robust dissimilarity matrix.

Definition: PX_{KMEANS} (Pixel extraction with k-means)

Construction of initial representatives

Define starting midpoints $m_1 = (R_{fg}, G_{fg})$ and $m_2 = (R_{bg}, G_{bg})$, where R_{fg} and G_{fg} are the highest red and green intensity values and R_{bg} and G_{bg} the lowest ones.

Determination of local optimum of cluster problem (k-means)

Repeat alternating the following two steps until convergence:

- Assign each data point to its closest of the two midpoints.

- Calculate two new midpoints as the means of all points assigned to the old midpoints. The outcome are two cluster midpoints m_1 and m_2.

Reduction

Let R_{fg} and G_{fg} be the values of the cluster mean with higher intensity values and R_{bg} and G_{bg} those of the other one. Calculate $(R_{fg} - R_{bg})/(G_{fg} - G_{bg})$ as the final relative abundance estimate.

Several refinements of this algorithm have been introduced in the original paper, especially the use of the median instead of the mean in step 3. This adjustment is not justified in theory. A more theoretically based approach replaces k-means by PAM. Here, steps 2 and 3 are inherently solved in one step. The cluster algorithm minimizes an objective function and returns the corresponding prototypes.

Definition: PX_{PAM} (Pixel extraction with Partitioning around medoids (PAM))

Calculation of dissimilarity matrix of spot pixels

Calculate the Manhattan distances between all pairs of pixels, i.e. the distance between two pixels is defined as the sum of the absolute differences of red and green intensities.

Construction of initial representatives

Build phase of PAM with Manhattan dissimilarity matrix.

Determination of local optimum of cluster problem

Swap phase of PAM.

Reduction

See PX_{KMEANS}, only cluster means are replaced by cluster medoid pixels.

The analysis of real microarray spots showed that both methods yield very similar final results in practice. Whereas the refined PX_{KMEANS} represents a more heuristic solution, the theoretically more meaningful use of PAM is by far more time consuming and sometimes lacks stability. In the following we refer to pixel clustering as implemented in PX_{KMEANS}. For related literature on the 2-center

problem and cluster analysis we refer to Agarwal et al. [15], Gordon [16], Haralick and Shapiro [17] and Jain et al. [18].

Hybrid pixel clustering including the spot shape

Pixel clustering applied to real microarray images on a larger scale disclosed two drawbacks. To address these problems, the pixel clustering algorithm is extended by two steps. We first present the problems and then the improved algorithm.

Very few foreground pixels for low-quality spots

Sometimes spots are comprised of a few outlier pixels with extreme high intensity, due to noise or generally low foreground intensities. Consequently, very few pixels (less then 20 in most cases) are assigned to foreground and the rest to background. Many true foreground pixels are missed, and the spot intensities are overestimated.

Missing incorporation of spot shape

Most spot shapes are recovered well by pixel clustering. In some cases, however, regions within the spot target areas are obviously incorrectly assigned, taking into consideration the known theoretical circular shape of microarray spots. In particular, artifacts in the background are assigned to foreground and inner regions with low intensities are assigned to background.

Definition: HYBRID PX_{KMEANS} (Hybrid pixel extraction with k-means)

Determination of starting values and applying k-means

See PX_{KMEANS}**.**

Potential repeated clustering to increase number of foreground pixels

Choose a minimum number m of pixels for the foreground area, typically at least $m = 50$. As long as this number is not reached, the set of background pixels alone is continuously clustered into two groups and the brighter group is assigned to foreground.

Pixel reduction through mask matching

All target areas have the same size by construction of the grid clustering algorithm. For every pixel with a fixed position in the target area, count the number f, for how many spots of the array the pixel is considered to be in the foreground. Calculate the average $\bar{f}$ of these values over all pixels. Define the 'bivalence mask' as follows. If $f > \bar{f}$, a pixel in the mask is assigned to foreground, otherwise to background. For a single spot, delete all pixels that are assigned to foreground in the spot and to background in the mask and vice versa.

Reduction

See PX_{KMEANS}, only use medians instead of means and omit pixels that are deleted due to mask matching in step 4.

Step 3 is designed to guarantee a minimum size for the foreground area. The idea of step 4 is to overlay all single spot clustering results in order to obtain more reliable information through the aggregated results. The bivalence mask represents a prototype for the average spot shape (for an example see Figure 3). Note that frequencies are not compared to the constant 1/2, but to the average value $\bar{f}$ for all pixels. This can be interpreted as a statistical method, based on the comparison with the null hypothesis of random foreground or background membership. A convenient consequence of the mask matching step is that pixels are divided into three groups instead of two: Foreground, background and deletions. This is desirable, since it allows a separate treatment of artifacts and their elimination from the further analysis.

Quality measure

To demonstrate the practical feasibility of this proceeding, a suitable quality measure is introduced. In a perfect scenario, all spots have the same shape, can be fully identified by the clustering algorithm, and no pixels are deleted through mask matching. The deviation from this ideal case illustrates, how well the algorithm works for a specific array.

Definition: STABILITY

STABILITY

For a fixed spot on the array, the *stability s* is defined as the relative frequency of pixels in the foreground area that are not deleted by mask matching in step 4 of the HYBRID PX_{KMEANS} algorithm.

MEDIAN STABILITY

For a whole microarray, the *median stability* $\bar{s}$ is defined as the median of all single spot stabilities s.

Obviously, it holds $0 \le s \le 1$ and $0 \le \bar{s} \le 1$. Values close to 1 indicate a high quality for the respective spot or array. 'Stability' is well suited as a quality measure, since the original pixel clustering algorithm doesn't take the shape into account at all. Thus the 'stability' represents a good subsequent control for the correct assignment of the pixels. The 'median stability' is a meaningful scalar summary

measure. Especially so-called black holes with higher background than foreground intensities can be easily detected. In such a situation, most pixels are deleted by the mask matching step and the stability yields a number close to 0. In general, low stability values can be caused both by a failure of the algorithm and by poor array quality.

Authors' contributions

JR and DB developed the algorithms. JR carried out testing and fine tuning of the algorithms using the statistical programming language R. DB implemented the methods in a Java software tool. JR drafted the manuscript.

Acknowledgments

This work was supported by the "Deutsche Forschungsgemeinschaft" (JR, RA 870/2-2), by the Nebraska Research Initiative Grant 31-3205-0502 (DB) and through a consultancy (JR) on NIH grant R01HD037804-04 (Claudia Kappen). We thank Andrea Krempler for carefully reading the manuscript. Parts of this work were done at the Department of Statistics, University of California, Berkeley.

References

1. Schena M, Shalon D, Davis RW, Brown PO: **Quantitative monitoring of gene expression patterns with a complementary DNA microarray.** *Science* 1995, **270:**467-470.
2. Shalon D, Smith SJ, Brown PO: **A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization.** *Genome Res* 1996, **6:**639-645.
3. Axon Instruments Inc: **GenePix 4000A User's Guide.** 1997.
4. Biodiscovery Inc: **ImaGene.** 1997 [http://www.biodiscovery.com/imagene.asp].
5. GSI Lumonics: **QuantArray Analysis Software, Operator's Manual.** 1999.
6. Eisen MB: **ScanAlyze.** 1997 [http://rana.Stanford.EDU/software/].
7. Steinfath M, Wruck W, Seidel H, Lehrach H, Radelof U, O'Brien J: **Automated image analysis for array hybridization experiments.** *Bioinformatics* 2001, **17:**634-641.
8. Chen Y, Dougherty ER, Bittner ML: **Ratio-based decisions and the quantitative analysis of cdna microarray images.** *Journal of Biomedical Optics* 1997, **2:**364-374.
9. Buckley MJ: **The Spot user's guide.** *CSIRO Mathematical and Information Sciences* 2002 [http://www.cmis.csiro.au/IAP/Spot/spotmanual.htm].
10. Adams R, Bischof L: **Seeded region growing.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1994, **16:**641-647.
11. Yang YH, Buckley MJ, Dudoit S, Speed TP: **Comparison of methods for image analysis on cDNA microarray data.** *Journal of Computational and Graphical Statistics* 2000, **11:**108-136.
12. Bozinov D, Rahnenführer J: **Unsupervised technique for robust target separation and analysis of microarray spots through adaptive pixel clustering.** *Bioinformatics* 2002, **18(5):**747-756.
13. Bozinov D, Rahnenführer J, Burson CM, Spiegelstein O: **Automated grid alignment for high-throughput Analysis of Microarray Images.** *Proc 2002 Int Conf on Imaging Science, Systems and Technology (CISST'02): Las Vegas* Edited by: Arabnia HR, Mun Y. CSREA Press; 2002:161-167. 24–27 June 2002
14. Kaufman L, Rousseeuw PJ: *Finding groups in data: An introduction to cluster analysis* New York: Wiley; 1990.
15. Agarwal P, Sharir M, Welzl E: **The discrete 2-center problem.** In *Proceedings of the 13th ACM Symposium on Computational Geometry: June 1997; Nice* 1997:147-155.
16. Gordon AD: **A survey of constrained classification.** *Computational Statistics and Data Analysis* 1996, **21:**17-29.
17. Haralick RM, Shapiro LG: **Image segmentation techniques.** *Computer Vision, Graphics and Image Processing* 1985, **29:**100-132.
18. Jain AK, Murty MN, Flynn PJ: **Data Clustering: A Review.** *ACM Computing surveys* 1999, **31(3):**264-323.