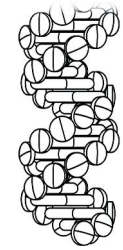


# The analysis of microarray data

*Ramesh Hariharan*

*Strand Genomics Private Limited, and Indian Institute of Science, Bangalore, India, 560080*
*Tel: +91 80 361 1349;*
*Fax: +91 80 361 8996;*
*E-mail: ramesh@strandgenomics.com*

This article describes issues, techniques and algorithms for analyzing data from microarray experiments. Each such experiment generates a large amount of data, only a fraction of which comprises significant differentially expressed genes. The precise identification of these interesting genes is heavily dependent not only on the statistical data analysis techniques used but also on the accuracy of the previous oligonucleotide probe design and image analysis steps as well. Indeed, wrong decisions in these steps can multiply the number of false positives by many-fold, thus necessitating a careful choice of algorithms in all three steps. These steps are described here and placed in the context of commercial and public tools available for the analysis of microarray data.

## Introduction

Gene microarrays constitute a powerful and increasingly popular platform for studying changes in gene expression on a large scale. This platform allows tracking changes in gene expression for the entire transcriptome (several thousands or tens of thousands of genes) simultaneously. A single microarray experiment yields gene expression information not only about individual genes but about joint behavior of collections of genes as well. Unraveling this joint behavior facilitates the study of biological phenomena on a hitherto impossible systemic scale. Indeed, microarrays have become invaluable tools for gene discovery, disease diagnosis, pharmacogenomics and toxicogenomics.

There are of course challenges in using microarrays to study biological systems. The chain of events between biological sample and final outcome is very long and involves several experimental and computational steps:

- sample preparation
- oligonucleotide probe design
- oligonucleotide synthesis
- slide preparation and spotting
- hybridization
- washing
- image analysis
- statistical data analysis

Errors in each step will also affect the accuracy of the final results considerably. Indeed data generated from microarrays usually show a large number of false positives (i.e., genes which are not differentially expressed but appear as being so).

This article surveys computational issues which arise in the three main computational steps in the previously mentioned chain, namely, *oligonucleotide probe design*, *image analysis*, and *statistical data analysis*, and studies their effects on the final outcome (i.e., the list of genes declared to be differentially expressed). Choosing the right algorithms in each of these steps is critical as small changes in algorithm can increase the number of false positives by several-fold. As the title of a popular article on gene expression informatics reads, *it's all in your mine* [1].

The oligonucleotide probe design step requires choosing appropriate probes specific to each gene of interest. This is a fairly complex multiparameter problem. A good probe needs to satisfy several properties, some of which are predicted by modeling physical hybridization phenomena which are only partially understood. Therefore, probe behavior in an experiment is not always as predicted or expected, especially at low expression levels, and this can have a significant impact on the final results. Since a microarray image has several tens or hundreds of thousands of spots, image analysis is necessarily an automated step and not always amenable to manual checking or correction. Differences in segmentation and background correction methods in the image analysis step can affect the final outcomes substantially. Finally, the statistical data analysis step performs a variety of statistical analyses on spot quantitated data to assess whether a gene is truly differentially expressed or not. There are several issues which this step needs to address, for example, normalizing arrays to factor out differences due to non-biological conditions variations,

**Keywords:** gene expression, image analysis, microarrays, oligonucleotide probe design, statistical data analysis

*Ashley Publications Ltd*
*www.ashley-pub.com*

performing the right transformations and statistical tests to identify differentially expressed genes so as to reduce the number of false positives etc. This is a very vibrant area of research, and it is probably fair to say that there have been tremendous advances in the last couple of years contributing to a substantially better understanding of various microarray platforms and consequently far more accurate results.

One of the key problems in assessing and comparing various algorithms for microarray data analysis is that there are few or no benchmarks or data sets available for the various available microarray platforms on which the *true* answers are known. Researchers do perform reverse transcriptase polymerase chain reaction (RT-PCR) studies on chosen individual genes to verify their level of differential expression and some of these studies have been reported in literature; however, since RT-PCR studies are performed only on a subset of interesting genes, it is not always easy to draw large scale statistical conclusions from these. Of great use here are experiment sets where a certain number of genes are spiked-in in known concentrations on a common background. Since only the spiked-in genes are truly differentially expressed, these data sets are invaluable in assessing and quantifying the relative accuracies of several algorithms. Some of the discussion in this article as regards comparative analysis will revolve around the Affymetrix Latin Square Dataset [101]; the other commonly used Latin Square Dataset is available from GeneLogic [102]. Of course, the fact that an algorithm performs well on these data sets does not mean that the algorithm will perform equally well on all data sets. Therefore, the individual steps of microarray data analysis should be performed in a quality-controlled fashion in order to find the method best suited for the data set. Possible quality control procedures include inspection of pre/postnormalization scatter plots and the observation of genes with known expression patterns.

### *Roadmap*

The following two sections describe issues in oligonucleotide probe design and image analysis, respectively, and examine the impact of these steps on the final results. This article then describes issues in statistical data analysis. This description is restricted to *primary* analysis (i.e., analysis aimed at identifying significant genes). There are several important steps beyond this primary analysis, notably identifying co-expressed/co-regulated genes using clustering approaches and mapping genes to pathways, which are not addressed in this article.

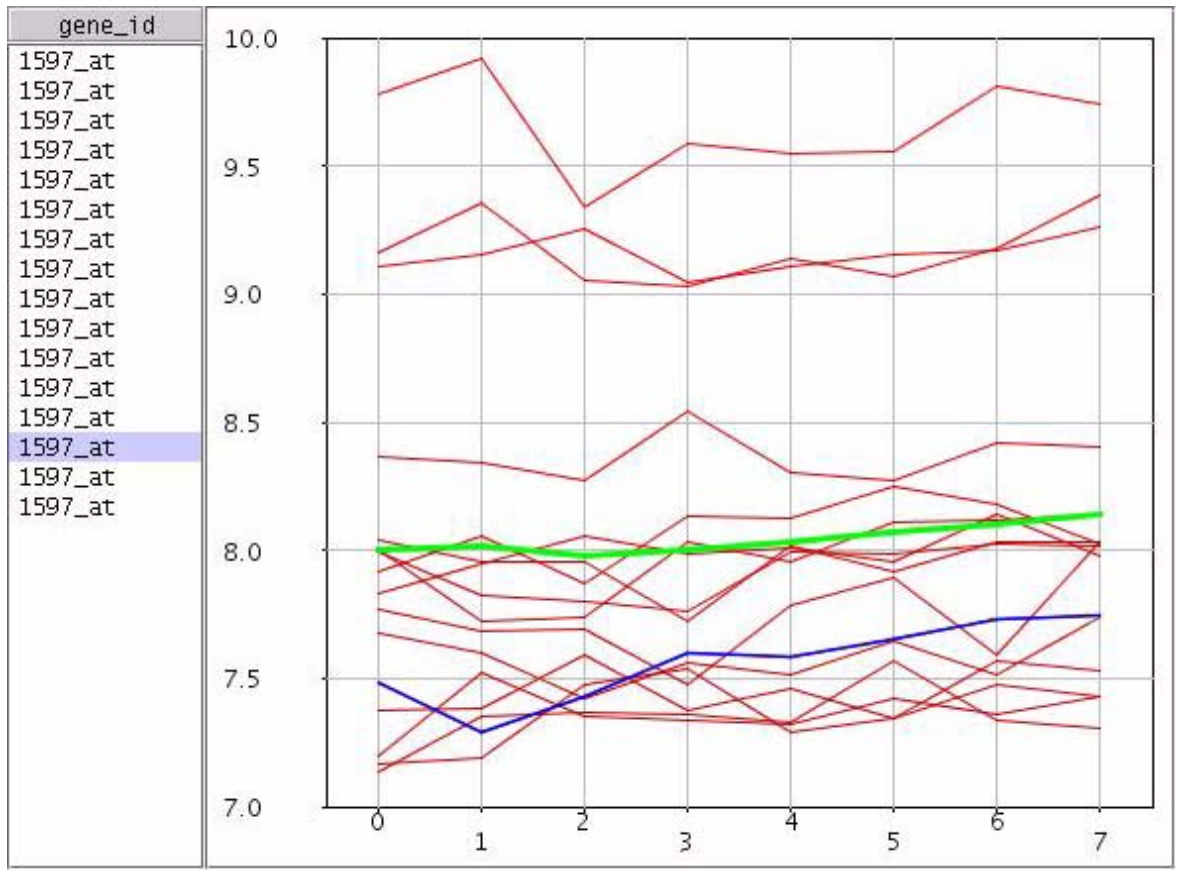
## Oligonucleotide probe design

Oligonucleotide arrays typically have less variation amongst replicates than cDNA arrays and many commercial platforms, for example, Affymetrix, Agilent, Amersham etc., use oligonucleotide arrays. However, the design of oligonucleotides poses some challenging computational problems in ensuring the following properties:

- *Specificity* – a probe should hybridize only to mRNA from the corresponding gene but not to mRNA from other genes.
- *Availability* – a probe should be available to hybridize with mRNA from the corresponding gene. In particular, it should not form secondary structures which prevent this hybridization.
- *Uniformity* – all probes on an array must have somewhat uniform thermodynamic properties so they behave as required under a common experiment condition.

Specificity is usually enforced by ensuring that there is no non-specific match within certain homology limits, for example, either with homology > 75% or having a 15-mer continuous exact match stretch [2,103]. Often homology may not be a sufficient criterion in avoiding cross-hybridization because thermodynamic properties of hybridization depend not only on percentage homology but also on the base composition and, therefore, low homology matches with substantial GC content could still make for a stable binding [104]; it may then be more effective to estimate a *cross-melting temperature* (the melting temperature of the strongest non-specific cross match) and ensure that this temperature is well below the *self-melting temperature* (the melting temperature of the perfect match) [105]. Prediction of melting temperatures is usually performed using nearest neighbor parameters [3-5]. This approach has two problems: first, nearest neighbor parameters are usually obtained from studies in solution which may not be applicable directly to the array surface, and second, these parameters are available only for perfect match and single mismatch duplexes. Furthermore, there seem to be allied parameters which govern the specificity, for example, the number of non-specific matches with a modest predicted melting temperature, the number of low complexity subsequences in the oligo etc.

**Figure 1. Profiles for probeset 1597_at over eight arrays.**




Each red profile is the profile of one of the 16 probes over the 8 arrays. These 8 arrays appear along the x-axis with arrays 0,1,2,3 comprising one group of replicates and arrays 4,5,6,7 comprising the other group. The green line shows the average probeset profile.

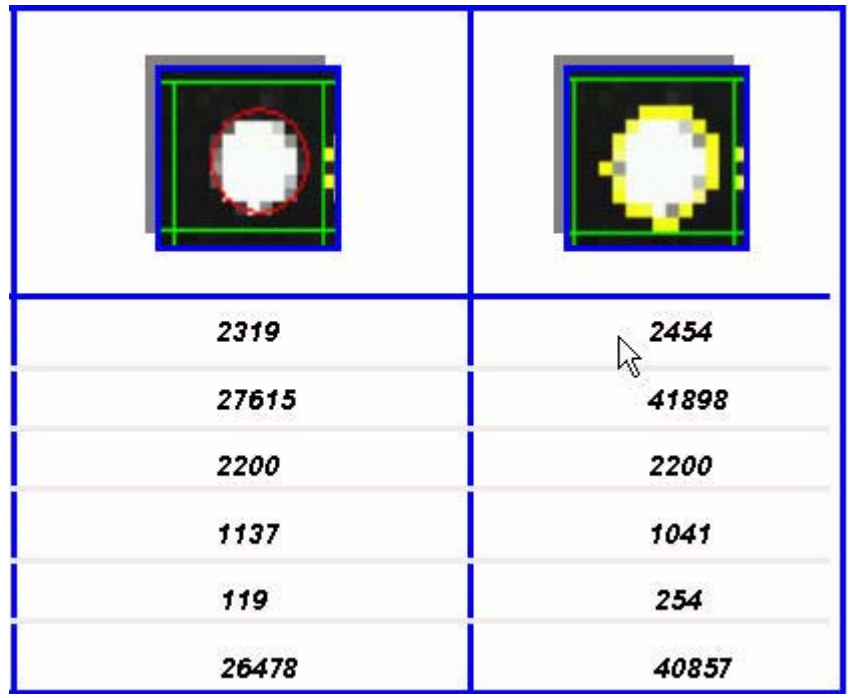
Availability is typically tested using secondary structure prediction for both the oligo and the gene sequence. Uniformity requires a multiparameter optimization scheme to obtain probes which are optimal along several of the above parameters while keeping the parameters uniform. Finally, probe length plays a key role in determining all the above parameters. While Affymetrix uses 25-mers, Bosch *et al.* [106] claim that 70-mers offer greater sensitivity. Agilent arrays use both 25-mers and 60-mers, while Amersham is focussing on 30-mers.

### *Computing with noisy probes*

While predictions of probe behavior made from *in silico* models as described above are faithful at a coarse level and for most probes, there seem to be no conclusive studies published demonstrating accurate specificity predictions for all probes at low expression levels. Indeed, at low expression levels, probe behavior is often very noisy as illustrated by the example in the next paragraph. However, using several noisy probes and averaging over these probes can attenuate the effect of noise here.

As an example consider probeset 1597_at on the Affymetrix HGU95 chip and consider the expression values obtained from the Affymetrix Latin Square Dataset [101]. The spike-in concentration of this gene goes from 0 in the first set of four replicates to 0.25 pm in the second set of four replicates. The profiles of all probes in this probeset over these eight arrays are shown in **Figure 1**. Note that the behavior of most probes is quite choppy; indeed, testing the individual probes for differential expression yields only two probes with a p-value smaller than 0.05. On the other hand, the average profile shown in green in **Figure 1** seems to be smooth and shows a distinctly higher value in the last four replicates as

**Figure 2. Spot segmented by circle fitting on the left and by exact boundary tracing on the right.**




This figure shows a spot segmented by circle fitting on the left and the same spot segmented by exact boundary tracing on the right. The associated spot statistics are displayed below the spots: the figures, in order from top to bottom, are Channel 1 and 2 intensities, Channel 1 and 2 backgrounds, and then Channel 1 and 2 background corrected values. The circle fitting approach on the left yields a background corrected value which is much smaller than the one obtained by accurate tracing on the right.

compared to the first four replicates; testing this average profile indicated differential expression with a much smaller p-value of 0.014. Thus, averaging over several probes leads to more accurate results, primarily because of cancellation of random noise on addition. Indeed, Affymetrix uses 10–20 probes per probeset to increase accuracy. However, increasing the number of probes per probeset produces uniformity challenges.

## Image analysis

Image analysis needs slightly different treatment for spotted arrays as opposed to synthetic arrays.

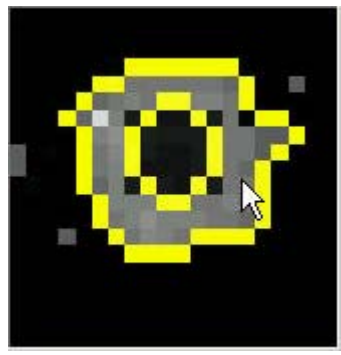
### *Spotted arrays*

There are three steps in image analysis of spotted arrays. The first step is called *addressing* or *gridding* and involves associating spots on the array with the row and column coordinates. The second step is *segmentation* which involves tracing the spot boundary so as to separate the spot foreground from the surrounding region. The final step involves *spot quantification* and *background correction* (i.e., computing foreground and background expression values for each spot and performing background correction to obtain a net expression value for each probe). Various algorithms used in the above steps are surveyed in Yang *et al.* [6].

The gridding problem is usually solved by a semiautomatic process where the user gives some manual tips (usually in the form of clicking at a few key points, e.g., the top left corner of the top left grid) and the machine figures out the rest, allowing the user to perform some final fine-tuning by hand. With the right user interface, the correctness of the gridding process can also be verified very quickly.

As regards segmentation, currently available tools follow one of two approaches. The first approach fits circles onto the spots of either fixed size (e.g., Scanalyse or Quantarray™) or an adaptive size (e.g., GenePix®). The second approach actually traces spot boundaries using

**Figure 3. An accurately traced doughnut spot.**



Only the portion inside the outer boundary and outside the inner boundary is considered for foreground computation.

more sophisticated algorithms (e.g., Spot [107] which uses the seeded region growing algorithm of Adams and Bischof [7], and Chitraka [108] which uses a clustering approach). The advantage of the second approach is that the foreground and background are separated better, leading to better quantitation. The two approaches above can yield substantially different values as indicated by **Figure 2**.

Furthermore, several spots on a spotted array actually have a doughnut like shape, fitting circles on which will attenuate the average foreground values. The accurate tracing methods can easily handle such spots as well, as shown in **Figure 3**.

Background correction can be performed either locally or globally. The local correction approach computes a background estimate from the non-spot regions adjacent to a given spot; typically, this estimate would be a robust average or median of the non-spot regions around a spot and should not be corrupted by brighter pixels belonging to the edges of spots. The global approach computes a background taking the whole array into account; this background value could be computed by taking, for example, the third percentile of all spot foreground values, the motivation being to subtract off background noise due to non-specific hybridization. The former approach is clearly better at handling spatial variations in intensity and the latter could be used in conjunction with the former. Most software packages available implement a local correction approach.

Dudoit *et al.* [8] observe that the choice of background correction method has greater impact on the final log intensity ratios than the choice of segmentation method and that a robust local correction method based on Morphological Opening [9] seems to produce most stable estimates of background.

### *Synthetic arrays*

We discuss only Affymetrix GeneChip® arrays here. Image analysis for these arrays is somewhat simpler as compared to spotted arrays because cells (the analogs of spots on a spotted array) have a fixed rectangular shape and are laid out in a fairly regular grid structure.

Once the corners of the grid have been identified, simple linear interpolation can identify the corners of each cell to within 3 pixels, as stated by Zuzan *et al.* [109]. Zuzan *et al.* [109] propose an algorithm for correcting this up to 3 pixel error. Interestingly, they also show that when the alignment is computed using Affymetrix software, a certain banding pattern occurs when one views an image in which each cell is replaced by a pixel with intensity proportional to the coefficient of variation. They ascribe this banding pattern to faulty alignment and demonstrate that their alignment algorithm corrects this spatial effect.

After alignment, the pixel values in each cell are averaged using a robust measure. Affymetrix software usually reports the 75th percentile of the pixel values within a cell along with the standard deviation. Zuzan *et al.* [109] mention that pixels near the cell borders need to be ignored in computing cell statistics.

### *Background correction*

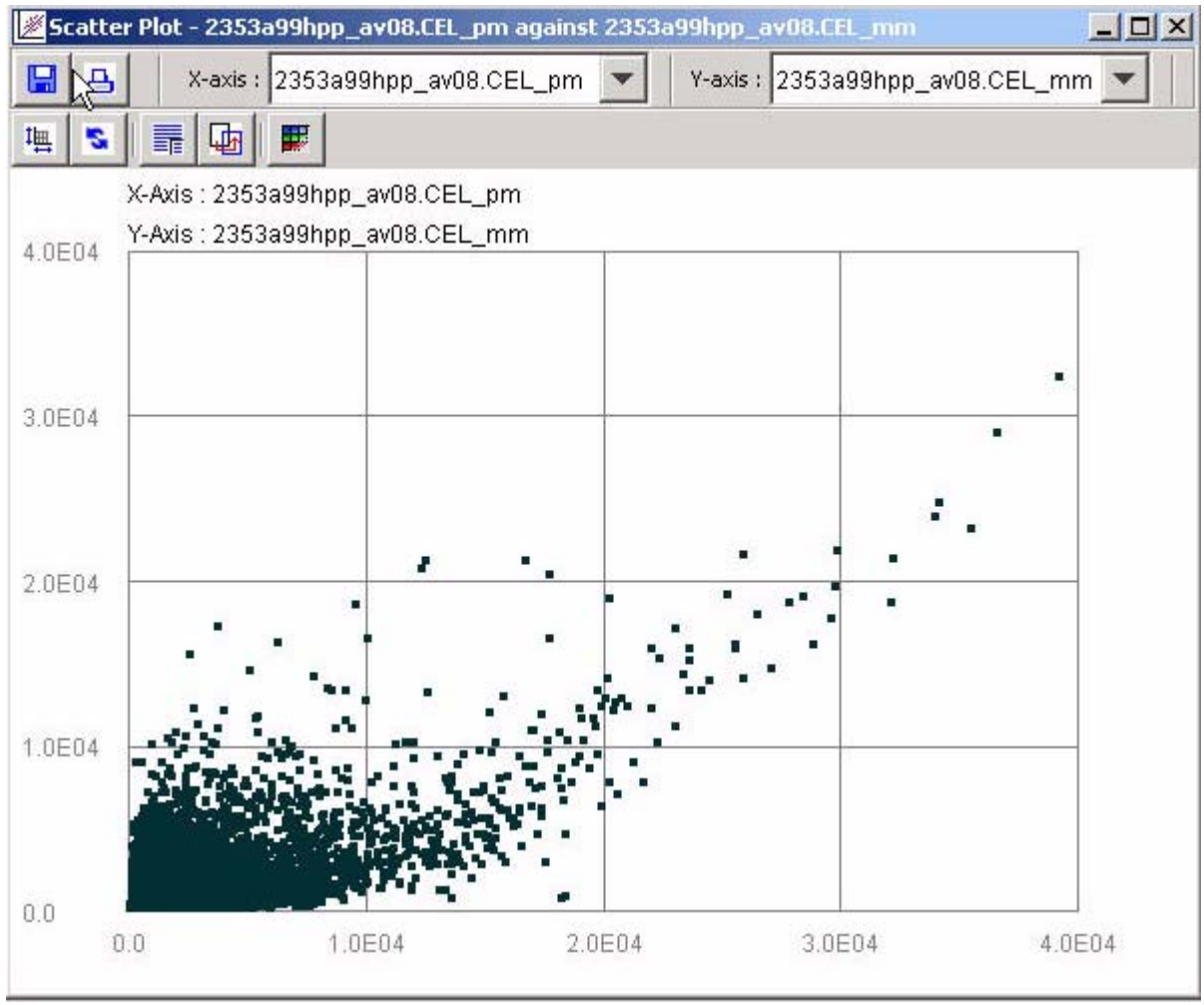
For Affymetrix high density arrays, the image processing task is fundamentally different because probes are synthesized *in situ* and, therefore, the addressing and spot segmentation tasks are much simpler. Since the probes are packed very tightly on the chip, the region surrounding the probes cannot be used for background computation. Therefore, background correction for such high density chips needs to be performed using the probe values themselves as explained later in this article.

## Statistical data analysis

Once the microarray image has been analyzed and quantitated, the next set of steps involves analyzing this data using statistical algorithms. The article reviews relevant issues which arise in each of the following steps.

The first of these steps is concerned with performing the right transformations to the raw

**Figure 4. The MM (Y-axis) values for about 12,000 genes from an array in the Affymetrix Latin Square Dataset plotted against their corresponding PM (X-axis) values.**




A large fraction of the MM values are much bigger than their corresponding PM values. In addition, for a fraction of genes, MM values seem to increase as PM increases.
MM: Mismatch value; PM: Perfect match value.

expression levels measured. There are two issues herein. The first question is whether data obtained from image analysis needs to be analyzed on the linear scale or converted to a different scale for analysis, for example, the logarithmic scale. The second issue is specific to the Affymetrix platform. Probes on an Affymetrix array are paired in PM,MM pairs, where PM refers to the perfect match oligo and MM to a oligo which has one mismatched base relative to the perfect match oligo. In this case, the image analysis phase outputs two values, a PM (or perfect match value) and an MM (or mismatch value). The following question now arises: what is the right way to combine PM and MM to get the expression value for a particular probe?

Once the right transformation is determined, the second step is that of *normalizing* the data to remove effects of non-biological variation. This issue arises naturally in experiments involving multiple arrays where observed expression levels across arrays include both biological variation and other sources of uninteresting variation arising from the production and processing of arrays, varying dye efficiencies in multi-dye arrays (i.e., for the same expression levels, different dyes may report different intensity values), and variations of ambient light intensities

between regions of an array and across arrays. See Hartemink *et al.* [10] for a more detailed description of such sources of variation.

The third step arises from the fact that several microarray platforms use multiple probes for each gene of interest, for example, the Affymetrix HGU95 chips had 16–20 probes per gene and the HGU133 chips have 10 probes per gene. As mentioned earlier, this seems to be a good way to fight noise. But it leads to another computational task: expression values for these various probes need to be aggregated into a gene expression value.

The final step in the statistical analysis phase is *statistical hypothesis testing* which would determine the statistical confidence with which each gene can be declared as *differentially expressed*. Algorithms for hypothesis testing and as well as for determining the trade-off between number of replicates and the number of false positives are relevant here.

### Data transformations

#### PM,MM and background correction for Affymetrix arrays

Several tools which analyze microarray data (e.g., Affymetrix MAS5.0 [110], Affymetrix MAS 4.0, and one of the main DChip options [111]) are based on the premise that MM measures noise due to non-specific hybridization while PM measures the actual signal, and therefore, PM-MM is a measure of the signal corrected for noise.

However, as observed by Irizarry *et al.* [11] and as illustrated in **Figure 4**, MM seems to pick up a fair amount of signal in addition to the noise, and thus PM-MM could result in an attenuated signal. In addition, for several probes, the MM values seem to be much larger than the corresponding PM values leading to the problem of negative values (though the Affymetrix MAS5.0 algorithms avoids negative values by damping the MM values whenever they are too large). For these reasons, PM-MM may not be a good measure of expression.

Indeed as the results in the following table show, PM-MM based methods (e.g., MAS5.0, the DChip Li-Wong PM-MM option) yield relatively poor results on the Affymetrix Latin Square Dataset when compared to pure PM based methods. This is indicated by the p-value ranks of the 14 actually differentially expressed genes, which ideally should get ranks 1 to 14, and may be a little but not much higher, if cross-hybridization/probe design errors etc. are taken into account, in **Tables 1 & 2**. Since the various methods in this table use different normalization methods, it is not obvious that the difference in performance is due solely to the use of PM or PM-MM. This is indeed verified by holding all parameters other than PM/PM-MM constant and running a comparative analysis. Comparing ALG1 versus ALG3 or comparing DChip PM-MM versus DChip PM confirms this hypothesis (though there are a couple of aberrant values in the DChip PM option). Thus, while there seems to be some information in the MM values, it is not clear how this information can be gainfully used to remove noise.

#### Affymetrix background correction

If MM is not used to remove noise, one needs to find alternative ways of removing noise in PM values which arises due to non-specific binding. This becomes particularly important at low values of expression where noise terms can drown out the real signal, for example, if the actual signal in two distinct arrays is $x$ and $2x$, then differential expression as measured by difference on the log scale would be:

$$\log(2x)-\log(x) = 1$$

while an additive noise level of 100 in both arrays would make this difference:

$$\log(100+2x)-\log(100+x)$$

which is close to 0 for small values of $x$. Note that for Affymetrix arrays, probes are so densely packed that regions adjacent to probes cannot be used to calculate backgrounds as for cDNA arrays.

Irizarry *et al.* [11] suggest one such background correction method which is based on the distribution of MM values amongst probes on an Affymetrix array. The key observation is that the smoothed histogram of the log(MM) values as displayed in **Figure 5** exhibits a sharp normal-like distribution to the left of the mode (i.e., the peak value) but not on the right. This observation suggests that the MM values are a mixture of non-specific binding and background noise on the one hand and specific binding on the other hand. In particular, the distribution on the right of the mode is possibly influenced by the MM values picking up some amount of signal and if this did not happen, the guess would be that the curve in **Figure 5** would look symmetric about the peak value (which would then be the mean) and, therefore, the average non-specific hybridization would

**Table 1. Ranks of the 14 spike-in genes in lot 1532 (arrays MNOPQRST) of the Affymetrix Latin Square Dataset where one gene goes down from 1024 pm to 0 pm, another from 0 to 0.25 pm, and all others double in concentration.**

| RMA | ALG1 | ALG2 | DChipPM | MAS5 | DChip | ALG3 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 2 | 1 | 2 | 3 |
| 3 | 2 | 5 | 3 | 2 | 11 | 5 |
| 4 | 5 | 6 | 8 | 5 | 18 | 7 |
| 6 | 7 | 7 | 11 | 9 | 32 | 9 |
| 8 | 10 | 11 | 12 | 24 | 36 | 10 |
| 9 | 18 | 14 | 13 | 43 | 43 | 13 |
| 10 | 22 | 20 | 19 | 54 | 58 | 27 |
| 16 | 24 | 25 | 36 | 69 | 122 | 82 |
| 19 | 28 | 26 | 45 | 75 | 246 | 111 |
| 21 | 29 | 27 | 92 | 159 | 267 | 119 |
| 47 | 30 | 45 | 97 | 271 | 338 | 504 |
| 99 | 46 | 47 | 442 | 272 | 1052 | 998 |
| 162 | 52 | 64 | 9960 | 2352 | 3571 | 9183 |

These ranks are obtained by performing t-tests using different probe aggregation and normalization methods. Ideally, the spike-ins should get ranks 0–13 in this process, however all methods give a fair number of false positives. The first four columns are algorithms based on PM alone and perform better on the whole than the last three methods which use the PM-MM measure.
MM: Mismatch value; PM: Perfect match value; RMA: Robust multi-array analysis..

**Table 2. ALG1 has the least maximum rank, namely 52, while RMA does best in the upper reaches.**

| Algorithm | Properties |
|---|---|
| RMA | [102] |
| ALG1 | Robust log(PM-bg), Quantile Normalization, bg = mode(MM) |
| ALG2 | Robust log(PM-bg), Lowess Normalization, bg = mode(MM) |
| DChipPM | [105] PM option |
| MAS5.0 | [114] |
| DChip | [111] PM-MM option |
| ALG3 | Robust log(PM-MM), Quantile Normalization |

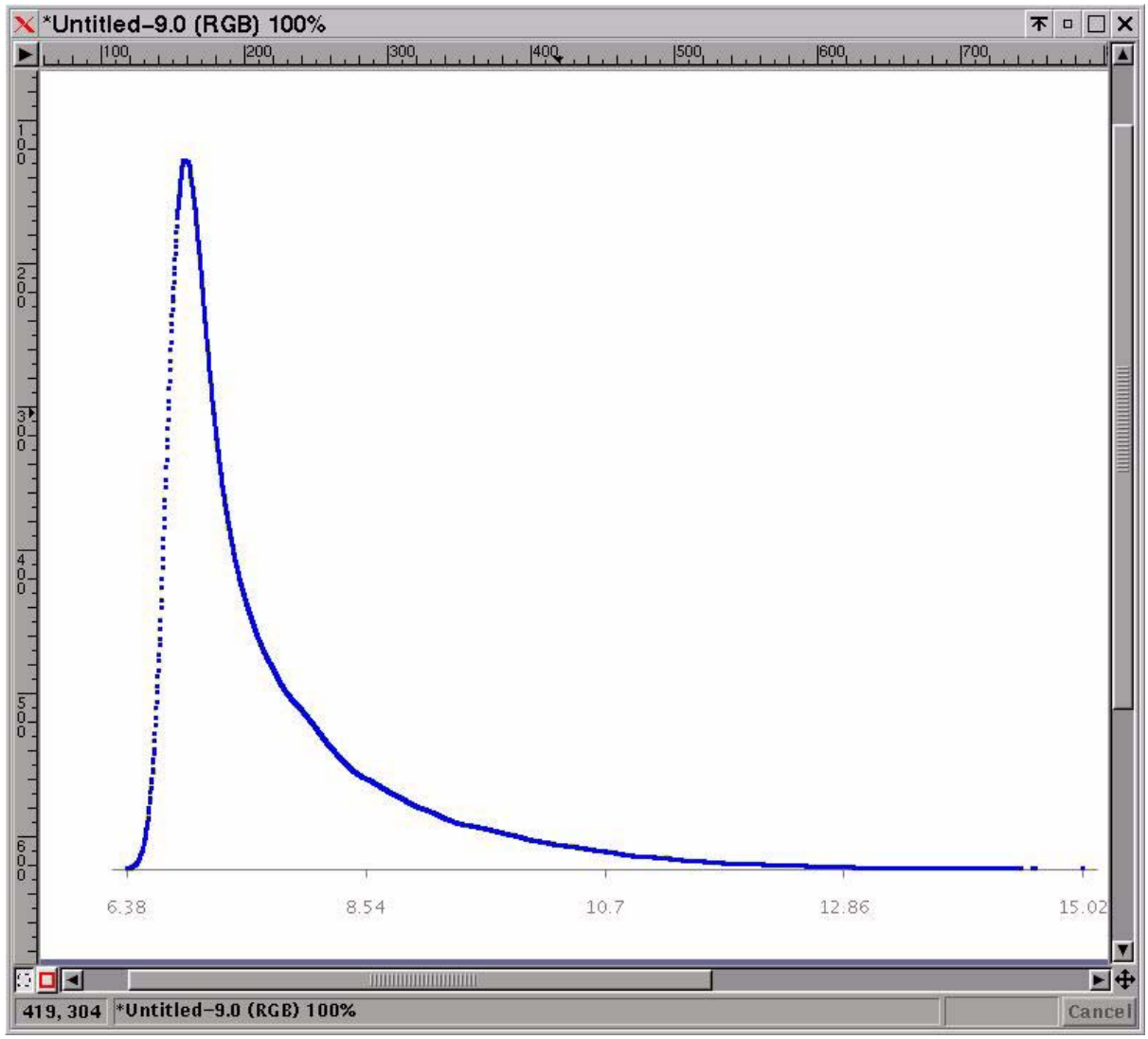
RMA [112] and MAS5 [110] results have been obtained using implementations in Soochika [116] and could therefore differ slightly from their native implementations. ALG1, ALG2 and ALG3 are native to Soochika. Note that the gene order provided by different methods is not the same.

be normally distributed around this mean value. Thus, the mode of the log(MM) distribution is a natural estimate of the average background noise, and this can be subtracted from all PM values to get background corrected PM values. However, the problem of negative values remains.

Irizarry *et al.* [11] solve the problem of negative values by suggesting a further extension of imposing a positive distribution on the background corrected values. They assume that each observed PM value $O$ is a sum of two components: a signal $S$ which is assumed to be exponentially distributed (and is therefore always positive) and a noise component $N$ which is normally distributed. The background corrected value is obtained by determining the expectation of $S$ conditioned on $O$ which can be computed using a closed form formula. However, this requires estimating the decay parameter of the exponential distribution and the mean and variance of the normal distribution from the data at hand. This method is used as part of the robust multi-array analysis (RMA) package in the Bioconductor suite [112].

The MAS5.0 algorithm (as reported in [110]) from Affymetrix uses a completely different method of background correction. The entire array is divided into 16 rectangular zones and the second percentile of the probe values in each zone (both PMs and MMs combined) is chosen as the background value for that region. For each probe, the intention now is to reduce the expression level measured for this probe by an amount equal to the background level computed for the zone containing this probe. However, this could result in discontinuities at zone boundaries. To

**Figure 5. The distribution of log(MM) values on an Affymetrix array.**



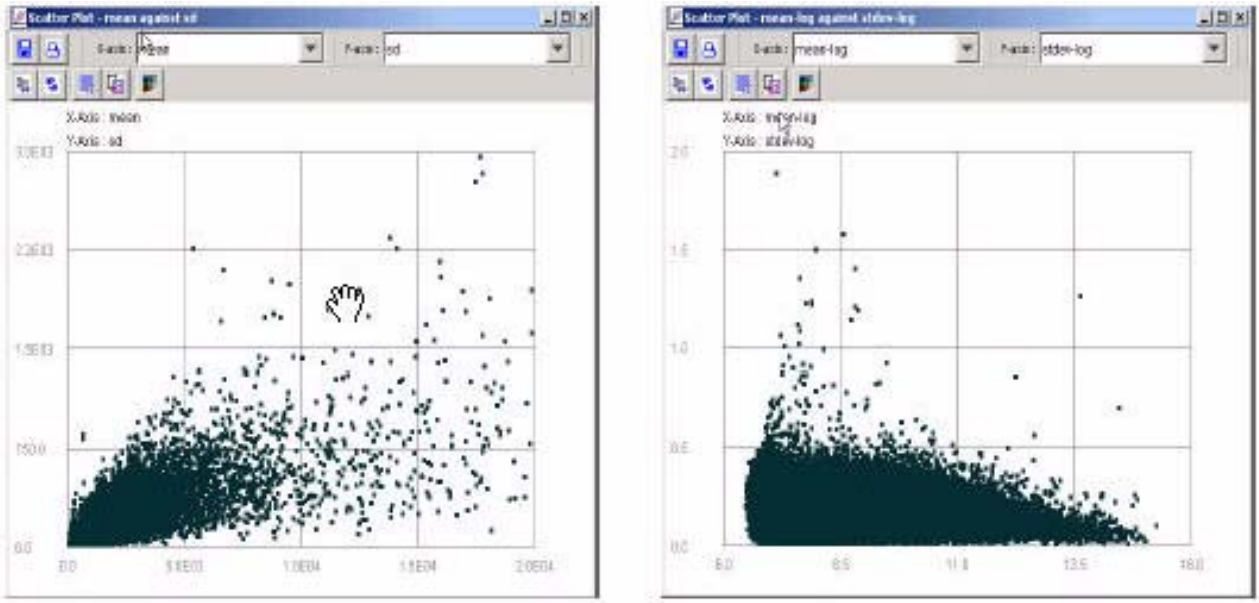

make these transitions smooth, what is actually subtracted from each probe is a weighted combination of the background levels computed above for all the zones. Negative values are avoided by thresholding.

We believe that background correction using the MM values contributes a small but significant improvement to results obtained. For example, running ALG2 from **Table 1** without the background correction increases the maximum rank from 64 to 115.

*The logarithmic scale*

To motivate the problem involved in determining the right transformation to measure gene expression, consider two conditions being studied, each having several replicates. The goal is to determine which genes are differentially expressed between these two conditions. Typically, this analysis proceeds independently for each gene. For a particular gene, typically a t-test (or a one-way analysis of variance, ANOVA, if there were more conditions) is performed. The numerator in this test is calculated from the means of expressions levels within each set of replicates, and the denominator is calculated using the standard deviations of expression values within each set of replicates. A standard assumption in this test is that the expression values in each set of replicates are drawn from a normal distribution; furthermore, the two normal distributions need not have the same mean

**Figure 6. These two figures show the mean (X axis) and the standard deviation (Y axis) of expression values for each of about 12,000 genes over four replicates in the Affymetrix Latin Square experiment.**



Expression values are measured on the linear scale in the figure on the left and on the log scale in the figure on the right. The average standard deviation over all 12,000 genes clearly increases in the figure on the left as the expression value increases. In the figure on the right, the average standard deviation stays somewhat constant or weakly decreases as the mean increases for a good stretch but decreases for very large expression levels.

but must *necessarily have the same variance*. This requires that the variance at different expression levels must be the same (i.e., roughly speaking) genes showing a high expression level should show approximately the same amount of variation across arrays as do genes with low expression levels. This assumption turns out to be not true for microarray data as illustrated in **Figure 6**.

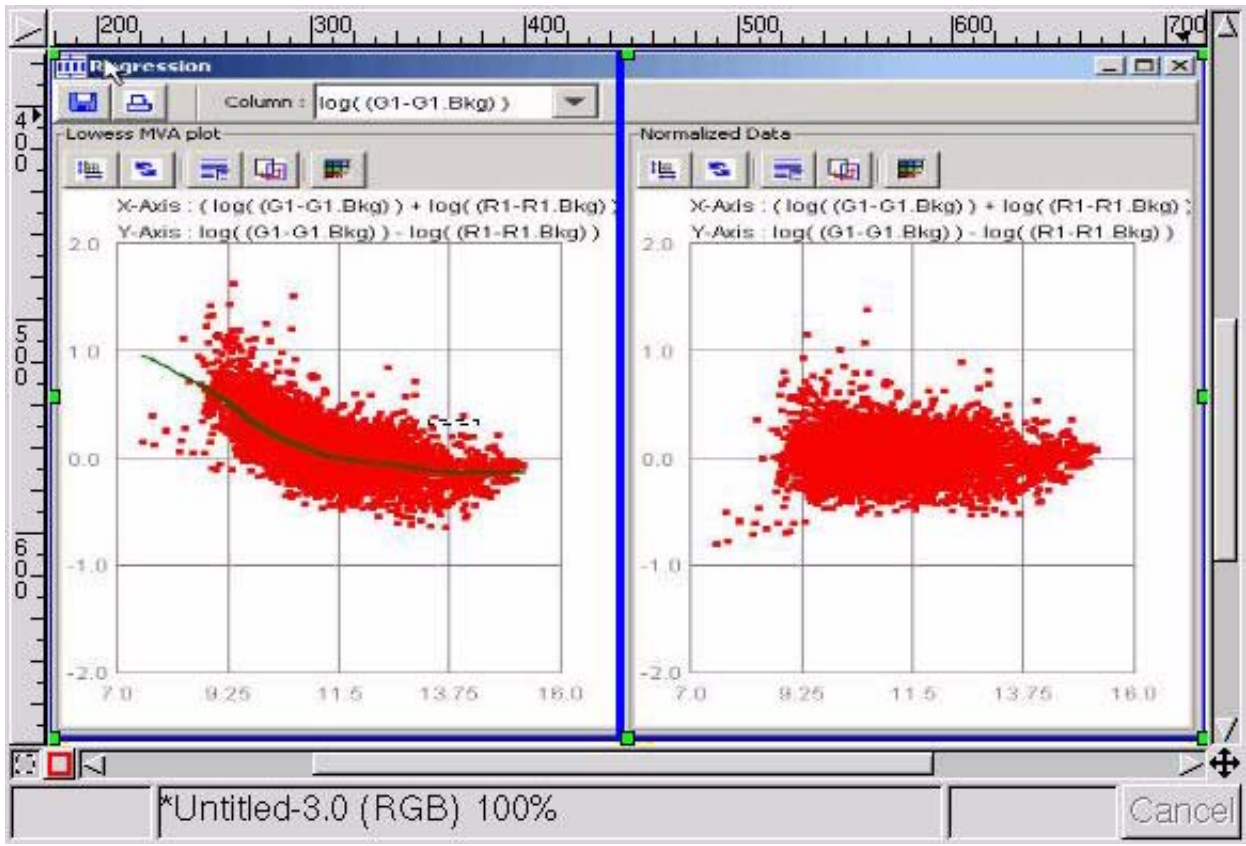
There are two options to work with the above variance behavior. The first requires devising new statistical tests to handle this behavior while the second, and possibly simpler, option is to perform appropriate transformations on the data. Converting the data to a logarithmic scale (using base 2) has various advantages (see the Speed page [113]), one of which is that the effect of expression level on the variance is somewhat mitigated, though not completely. The second advantage is of course for visualization; taking logarithms compresses the scale so that more information is visible within a given area.

The disadvantages of a plain logarithmic transform are listed in Durbin *et al.* [12]. The key operational irritant with the logarithmic transform is that it does not apply to negative values. While it may seem counter intuitive, negative values often do occur in microarray data, largely due to background correction, which often involves subtracting an estimated background value which is larger than the measured expression value itself. For example, the MM value for a probe on an Affymetrix array supposedly detects background noise due to non-specific hybridization and is often larger than its PM counterpart. As argued by Durbin *et al.* [12], a further disadvantage of the logarithmic transform is that the variance for very low expression levels shoots up.

There are several approaches to overcoming the above mentioned disadvantages of the logarithmic transform. Typically, the problem of negative values is avoided by thresholding or by performing background correction as described earlier. To avoid the problem of low expression levels, Durbin *et al.* [12], Huber *et al.* [13], and Munson [14] independently came up with the following generalized logarithmic transform:

$$f(x) = ln\left[\frac{(x + \sqrt{x^2 + c^2})}{2}\right]$$

**Figure 7. MVA plots before and after Lowess normalization. Non-linear curve fitting is essential in this case. The plots show log(Ch1)-log(Ch2) versus $\frac{log(Ch1)+log(Ch2)}{2}$ for a 2-channel array.**




MVA: ?

This transformation behaves linearly near 0 and like the logarithmic function for large expression values, and furthermore, is defined even for negative values. However, the estimation of the parameter *c* requires error modeling of the data, which takes it beyond the realm of simplicity. In practice, working with the plain log transform achieves a good balance between effectiveness and simplicity.

### *Normalization*

Results with and without normalization can be substantially different, for example, in **Tables 1 & 2**, running ALG2 without background correction or normalization yields a massive maximum rank of 9313 instead of 64. Thus, normalization of data is essential before further analysis. A variety of normalization methods have been studied and used by researchers, all of which attempt to remove non-biological variation between arrays or between dyes on the same array based on the premise that most genes are not differentially expressed across arrays. While much of the description below is in terms of normalizing across arrays, the techniques described are applicable to normalizing across multiple dyes on an array as well, as will be described at the end of this section. The need for normalization can be discerned by viewing the so-called MVA plot [8] between expression values on two arrays (the MVA plot shows, for each probe, the difference between the expression levels in the two arrays plotted against the average of these two expression levels). The distribution of points on this plot should be around the horizontal 0 line, assuming that most probes do not show differential expression between two arrays (for example see **Figure 7**). However, in actual practice, the MVA plot may not be centered around this zero line.

**Table 3. Quantile normalization.**

| | array1 | array2 | array3 | array4 |
|---|---|---|---|---|
| ***gene1*** | 10 | 1000 | 10 | 10 |
| ***gene2*** | 10 | 10 | 10 | 10 |
| ***gene3*** | 10 | 10 | 10 | 10 |
| ***gene4*** | 1000 | 1000 | 1000 | 1000 |

**Table 4. Quantile normalization.**

| | array1 | array2 | array3 | array4 |
|---|---|---|---|---|
| ***gene1*** | 10 | 257.5 | 10 | 10 |
| ***gene2*** | 10 | 10 | 10 | 10 |
| ***gene3*** | 257.5 | 10 | 257.5 | 257.5 |
| ***gene4*** | 1000 | 1000 | 1000 | 1000 |

*Mean shifting*

This method removes variations across arrays by equalizing the means of the various arrays. To make this robust (i.e., not sensitive to outliers) typically a trimmed mean is used in which the mean is computed after ignoring a certain fraction of the highest and lowest values. Assuming that expression values are measured on a logarithmic or related scale, mean shifting corresponds to global scaling on the linear scale. This normalization is used as part of the Affymetrix MAS4.0 and MAS5.0 algorithms.

*Interpolation methods*

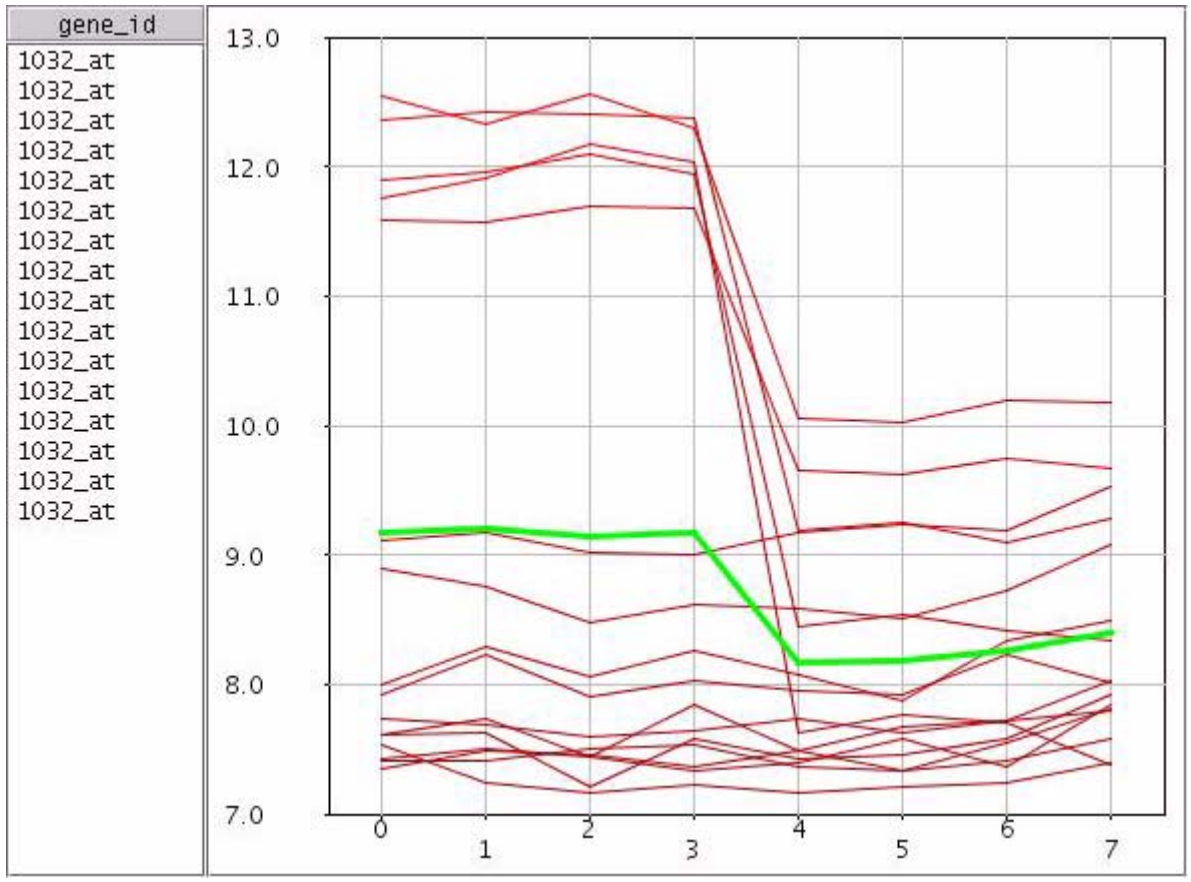
This is a pairwise normalization method in essence and is used to normalize one array against another. The mean shifting algorithm is suitable when the same amount of shift can be applied at all expression levels. However, there are instances where the MVA plot appears as in **Figure 7**. Mean shifting will not normalize the data in this case because the amount of shift required for different expression levels is different. One method which is used in such situations is to fit a straight line or even a piecewise linear curve to the data and shift the expression values on one of the arrays so that this curve moves to the horizontal zero line as in **Figure 7**. The need for a non-linear (or piecewise linear) method arises often in the context of 2-dye arrays because of the dependence of dye efficiency on intensity. In such cases, fitting a non-linear curve and straightening it out to the horizontal zero line as in Schadt *et al.* [15] does the trick. This method is used in the DChip software [111]. One popular method to fit a non-linear curve is the piecewise linear Lowess (or Loess) smoother [114].

As for the case of Mean Shifting, robustness is an issue. One way to make these interpolation methods more robust is to perform the normalization in multiple steps: first fit a curve, remove a certain fraction of points which are furthest away from the curve and which, therefore, are likely to be outliers, and finally fit a curve again on the remaining points. Multiple runs of these can be done as well to further increase robustness.

*Remarks*

Both the above methods are based on the premise that most genes are not differentially expressed between arrays and that the total mRNA content of different samples is the same. As stated in Hoffmann *et al.* [16], this need not be true if cells of different sizes or in different stages of the cell cycle are considered, and even though control of this effect is attempted by loading identical amounts of cRNA onto the arrays, there can still be significant variation in the mean expression level across arrays (see Hill *et al.* [17]). Furthermore, if the experiment has a dramatic effect, for example, causing a diauxic shift, then a good fraction of the genes can actually be differentially expressed. In such situations, it makes sense to perform the above methods (more specifically, either the trimmed mean calculation or curve fitting) on a subset of the probes (called the invariant set) obtained as follows. The probes on each of the arrays are ranked by expression value and probes with similar ranks comprise the invariant set. The trimmed mean or curve fitting is performed on the

**Figure 8. A mixture of two distinct profile types.**




invariant set only and then the actual shift or shifts are performed on all the probes.

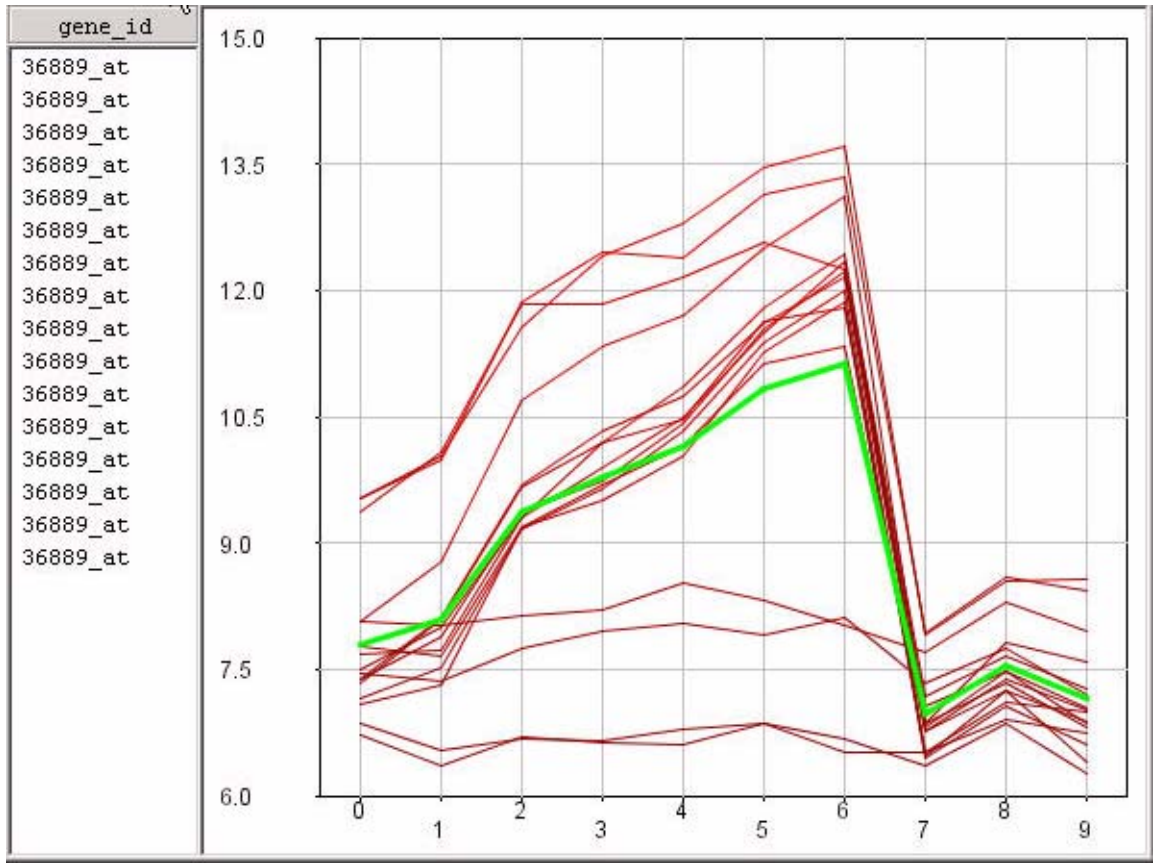
*Quantile normalization*

This normalization method attacks the problem from a different angle. Each array contains a certain distribution of expression values and this method aims at making the distributions across various arrays not just similar, but identical. This is done as follows: imagine that the expression values from the various arrays have been loaded into a spreadsheet with genes along rows and arrays along columns. First, each column is sorted in increasing order. Next, the value in each row is replaced with the average of the values in this row. Finally, the columns are unsorted (i.e., the effect of the sorting step is reversed so items in a column go back to wherever they came from). It is easy to see that the distributions in all arrays become identical in this process. This method is used in the RMA software which is part of the Bioconductor suite [112]. Statistically, this method seems to obtain the sharpest normalizations; however, occasional dangerous side effects could result, for example, consider the following artificial situation where one aberrant value of 1000 for gene 1 can create incorrect values for gene 3 as a result of quantile normalization. While such side effects are very rare, these could go undetected if they do indeed occur. To guard against these, it is best to do quantile normalization before probe aggregation, so that the process of aggregating probes removes any noise created by such side effects.

*Comparative analysis of the above methods*

Note that the interpolation methods are pairwise methods (i.e., each array is normalized against a chosen baseline array) while the mean shift and quantile methods do not require a baseline array. Furthermore, the mean shift, quantile and linear interpolation methods are much faster than the non-linear interpolation method. Comparative analysis of these methods and some further

**Figure 9. Aberrant probes appear flat: Probeset 36889_at from the Latin Square Dataset.**




derivatives of the Lowess method designed to eliminate the need for a baseline array is provided in Bolstad *et al.* [18]. The key conclusion is that the non-linear method and its derivatives which do not require a baseline and the quantile method typically perform better than the plain unnormalized data or normalizing through mean shifting.

### *Normalization on two-dye cDNA arrays*

For two-dye cDNA arrays, there are usually non-linear relationships between red and the green channel intensities and these need to be normalized using a Lowess or related method within each array so that the log $R$-log $G$ = log($R/G$) values are clustered around the horizontal 0 line (in the MVA plot) on each of the arrays (see e.g., Dudoit *et al.* [8]). Further normalization across arrays may not be required. However, as mentioned in Dudoit *et al.* [8], there could be strong spatial or print-tip effects within an array, and thus it may be useful to perform Lowess normalization within each print-tip group. Yang *et al.* tackle these issues in further detail [19].

Finally, it is worth mentioning that normalization is a challenging area and all sources of variation are not yet understood well enough for accurate statistical modeling.

### *Probe aggregation*

As mentioned above, using several probes per gene and averaging over these probe-sets can indeed reduce noise substantially. Thus, once intensity values have been background corrected, transformed to a logarithmic scale, and normalized, the next task is to average individual probe profiles for a gene into a gene profile. This averaging needs to be done robustly to minimize outlier effects as necessitated by the example in **Figure 8**. This figure shows profiles for probeset 1032_at for the eight arrays being considered from the Latin Square Dataset. The first five

**Table 5. The rough maximum rank amongst the 14 spike-in genes computed for the Affymetrix Latin Square Dataset for various numbers of replicates.**

| No. of replicates | Rough maximum rank |
|---|---|
| 1 Replicate | 9000 |
| 2 Replicates | 2000 |
| 3 Replicates | 150 |
| 4 Replicates | 50 |

To catch all 14 spike-in genes, the number of false positives is roughly 9000 for the fold-change method with one replicate, but drops to about 50 for a t-test with four replicates. A big drop happens when going from two to three replicates. However, note that 11–12 of the 14 genes can be caught with far fewer false positives.

probes behave very differently from the remaining probes. In this case, it is not immediately clear that the first five probes can be removed as outliers, as they comprise close to a third of all probes (these are actually heavily overlapping probes, with low complexity repeats, a fact which is not used by any outlier removal algorithms). Nevertheless, the need for outlier removal is indeed clear. There are broadly two categories of probe averaging methods, those which consider one array at a time, and those which consider multiple arrays together.

*Single array methods*

The simplest method which works with a single array at a time takes the mean or median probe value on that array and considers only those probes whose values are within a certain number of standard deviations from this value. ALG1 and ALG2 mentioned in **Tables 1 & 2** use this approach. The MAS5 algorithm uses a related method called one-step Tukey Biweight [110]. This method involves finding the median and weighting the items based on their distance from the median so items further away from the median are downweighted. This could actually be run for multiple steps with the weights computed in each step used to compute the new weighted estimate and then reweighting the items until there is no further change. The Affymetrix MAS5.0 algorithm uses only one step of this procedure.

*Multiple array methods*

The key advantage of working with all arrays together is the following. Occasionally, there are probes on Affymetrix arrays which behave very differently from the remaining probes in their respective probe-sets (see **Figure 9**). These may not always be identifiable when only one array is considered at a time but clearly stand out when all arrays are considered together. Furthermore, applying a robust algorithm on one array at a time could cause the removal of a probe which shows a consistent and expected profile across arrays but exhibits rather high or low expression values. Therefore, using multiple arrays together could lead to greater robustness. The two notable methods which work with all arrays together are due to Irizarry *et al.* [11] and Li and Wong [20,21], respectively.

Irizarry *et al.* [11] model observed probe behavior on the log scale as the sum of a probe specific term, the actual expression value on the log scale, and an independent identically distributed noise term; they then estimate the actual expression value from this model using a robust procedure called *Median Polish*. This is a very elegant method used in the RMA package and deserves a brief description on account of its simplicity. It comprises the following steps. Consider probes along the rows of a spreadsheet and arrays along the columns.

*Median polish steps*

- Compute the median of each row and record the value to the side of the row. Subtract the row median from each point in that particular row.
- Compute the median of the row medians, and record the value as the overall effect. Subtract this overall effect from each of the row medians.
- Take the median of each column and record the value beneath the column. Subtract the column median from each point in that particular column.
- Compute the median of the column medians, and add the value to the current overall effect. Subtract this addition to the overall effect from each of the column medians.
- Repeat steps 1–4 until no changes occur with the row or column medians.

The final vector of column medians serves as the aggregate profile for the gene in question.

Li and Wong [20,21] use a slightly different model; they model observed probe behavior on the linear scale as a *product* of a probe affinity term and an actual expression term along with an additive normally distributed independent error term. The maximum likelihood estimate of the actual expression level is then determined using an estimation procedure which has rules for outlier removal. The outlier removal happens at multiple levels. At the first level, outlier arrays are determined and removed. At the second level, a probe is removed from all the arrays. At the third level, the expression value for a particular probe on a particular array is rejected. These three levels are performed in various iterative cycles until convergence is achieved. This method is incorporated in the DChip package [111].

When comparing probe aggregation between the PM based methods in **Tables 1 & 2**, median polishing of RMA seems to do very well in the upper reaches but the single array methods of ALG1 and ALG2 do much better at lower levels of expression and lower signal to noise ratio.

### *Statistical hypothesis testing*

Once the data at hand have been background corrected, converted to a logarithmic scale, normalized, and probe aggregated, logarithms of the expression levels of each gene (for single dye arrays) or the log-ratios for each gene (for two dye arrays) will be at hand. In what follows, these log values or ratios will be referred to simply as *expression values*. The next step is to perform statistical hypothesis testing on these values to determine which gene(s) show significant differential expression across two or more groups of replicates (for the case of two-dye arrays, the analogous goal is to determine genes which are differentially expressed between the samples in the two channels).

To explain the issues in statistical hypothesis testing, consider the simple case of two groups of experiments, typically a control group and a treatment group, each group having several replicates. Issues in dealing with multiple groups will be touched upon later.

#### *Why is fold-change not a good measure?*

The fold-change measure computes the difference between the group means for each gene (recall that we are working on a logarithmic scale and fold-changes translate to differences on this scale). A cutoff on this quantity is then used to determine genes which are differentially expressed. However, as explained in Tusher *et al.* [22], this gives a very large number of false positives. This stems from the fact that most genes are expressed at low levels where the signal-to-noise ratio is low and, therefore, fold changes occur at random for a large number of genes. Furthermore, at high expression levels, small but consistent changes in expression across arrays are not detected by fold-change. There are better alternatives to a fold-change test as described below.

#### *The t-test*

The standard test that is performed in such situations the so-called t-test, which measures the following t-statistic for each gene $g$ (see [23] for example):

$$t_g = \frac{m_1 - m_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

Here, $m_1$,$m_2$ are the mean expression values for gene $g$ within groups *1* and *2*, respectively, $s_1$,$s_2$ are the corresponding standard deviations, and $n_1$,$n_2$ are the number of arrays in the two groups. Qualitatively, this t-statistic has a high absolute value for a gene if the means within the two sets of replicates are very different and if each set of replicates has small standard deviation.

Thus, the higher the t-statistic is in absolute value, the greater the confidence with which this gene can be declared as being differentially expressed. Note that this is a more sophisticated measure than the commonly used fold-change measure (which would just be $m_1$-$m_2$ on the log-scale) in that it looks for a large fold-change in conjunction with small variances in each group, The power of this statistic in differentiating between true differential expression and differential expression due to random effects increases as the numbers $n_1$ and $n_2$ increase. To identify all differentially expressed genes, one could just sort the genes by their respective t-statistics and then apply a cutoff. However, determining that cutoff value would be easier if the t-statistic could be converted to a more intuitive p-value, which gives the probability that the gene $g$ appears as differentially expressed purely by chance. So a p-value of 0.01 would mean that there is a 1% chance that the gene is not really differentially expressed but random effects have conspired to make it look so. Clearly, the actual p-value for a particular gene will depend on how expression values within each set of replicates are distributed. These distributions may not always be known.

*Obtaining p-values*

Under the assumption that the expression values for a gene within each group are normally distributed and that the variances of the normal distributions associated with the two groups are the same (recall earlier discussion on variance stabilization), the above computed t-statistic for each gene follows a so-called t-distribution, from which p-values can be calculated. However, as stated in Hoffmann *et al.* [16] and Long *et al.* [24], these assumptions may not always be met (even though the t-test is known to be reasonably robust to departures from normality). To get around the problem of normality (i.e., to obtain a p-value in the absence of knowledge of distribution of expression levels within each set of replicates), a *permutation testing* method is sometimes used as described below.

*p-values via permutation tests*

As described in Dudoit *et al.* [8], this method does not assume that the t-statistics computed follows the t-distribution (which it would but only under the assumptions above), rather it attempts to actually estimate this distribution. Implementation-wise, this is a simple method as described below.

Imagine a spreadsheet with genes along the rows and arrays along columns, with the first $n_1$ columns belonging to the first group of replicates and the remaining $n_2$ columns belonging to the second group of replicates. The left to right order of the columns is now shuffled several times. In each trial, the first $n_1$ columns are treated as if they comprise the first group and the remaining $n_2$ columns are treated as if they comprise the second group; the t-statistic is now computed for each gene with this new grouping. This procedure is ideally repeated $\binom{n_1 + n_2}{n_1}$ times, once for each way of grouping the columns into two groups of size $n_1$ and $n_2$, respectively. However, if this is too expensive computationally, a large enough number of random permutations are generated instead. p-values for genes are now computed as follows.

Recall that each gene has an actual t-statistic as computed a little earlier and several permutation t-statistics computed above. For a particular gene, its p-value is the fraction of permutations in which the t-statistic computed is larger in absolute value than the actual t-statistic for that gene.

*Adjusting for multiple comparisons and controlling false discoveries*

Microarrays usually have genes running into several thousands and tens of thousands. This leads to the following problem. Suppose p-values for each gene have been computed as above and all genes with a p-value of < 0.01 are considered. Let $k$ be the number of such genes. Each of these genes has a less than 1 in 100 chance of appearing to be differentially expressed by random chance. However, the chance that *at least* one of these $k$ genes appears differentially expressed by chance is much higher than 1 in 100 (as an analogy, consider fair coin tosses, each toss produces heads with a 1/2 chance but the chance of getting at least one heads in a hundred tosses is much higher). In fact, this probability could be as high as $k * 0.01$ (or in fact $1 - (1 - 0.01)^k$ if the p-values for these genes are assumed to be independently distributed). Thus, a p-value of 0.01 for $k$ genes does not translate to a 99 in 100 chance of all these genes being truly differentially expressed; in fact, assuming so could lead to a large number of false positives. To be able to apply a p-value cutoff of 0.01 and claim that all the genes which pass this cutoff are indeed truly differentially expressed with a 0.99 probability, an adjustment needs to be made to these p-values. However, the actual nature of the adjustment has to take dependencies between the various genes into account for it to be effective.

See Dudoit *et al.* [8] and the book by Glantz [23] for detailed descriptions of various algorithms for adjusting the p-values. The simplest methods are the Bonferroni method and the Sidak method, which are motivated by the discussion in the previous paragraph. In the former, any dependencies in gene behavior are completely ignored and the p-value of each gene is multiplied by $n$, the total number of genes. In the latter, the p-value of each gene is replaced by $1 - (1 - p)^n$, where $p$ is the original p-value for this gene; this method is applicable only if the p-values of the various genes are completely independent of each other. A slightly more sophisticated method is the Holm step-down method in which genes are sorted in increasing order of p-value and the p-value of the $j$th gene in this order is multiplied by $n-j+1$ to get the new adjusted p-value (so the multiplier for the gene with smallest p-value is $n$ and for the gene with largest p-value is 1); this method too ignores dependencies between genes. In typical use, the above methods of p-value adjustment often turn out to be too conservative (i.e., the

p-values end up too high even for truly differentially expressed genes). Furthermore, methods assuming independence do not apply to situations where gene behavior is highly correlated, as is indeed the case in practice. Dudoit *et al.* [8] recommend the Westfall and Young procedure as a less conservative procedure which handles dependencies between genes.

The Westfall and Young [25] procedure is a permutation procedure in which genes are first sorted by increasing t-statistic obtained on unpermuted data. Then, for each permutation, the t-statistics obtained for the various genes in this permutation are artificially adjusted so that the following property holds: if gene *i* has a higher original t-statistic than gene *j*, then gene *i* has a higher adjusted t-statistic for this permutation than gene *j*. The overall corrected p-value for a gene is now defined as the fraction of permutations in which the adjusted t-statistic for that permutation exceeds the t-statistic computed on the unpermuted data. Finally, an artificial adjustment is performed on the p-values so a gene with a higher unpermuted t-statistic has a lower p-value than a gene with a lower unpermuted t-statistic; this adjustment simply increases the p-value of the latter gene, if necessary, to make it equal to the former. Though not explicitly stated, a similar adjustment is usually performed with all other algorithms described here as well.

All the above procedures aim at bounding the probability that even one of the genes declared as significant is not actually differentially expressed (this is called the *Family-wise Error Rate*). Benjamini and Hochberg [26] argue that requiring control of this error rate may be too conservative and suggest using an alternative measure called the *False Discovery Rate*, which seeks to bound the fraction of genes amongst those declared as significant which are not actually differentially expressed. This method assumes independence of p-values across genes; it orders genes in increasing order of p-value and multiplies the p-value of the *j*th gene in the above order by *n/j*. Finally, if genes above a p-value of *p* are considered significant, then the expected fraction of false discoveries in these genes is *p*.

Dow [27] has studied the effect of various p-value adjustment techniques and though the results are not conclusive, the Holm step-down method applied in reverse order (which is called *Reverse Holm* in [27]) is recommended as an appropriate method to balance control of false positives and false negatives.

Tusher *et al.* [22] use a variant of the above described techniques to determine differentially expressed genes and estimate the number of false discoveries (this is part of the significance analysis of microarrays [SAM] package [115]). They perform permutation tests as described above and compute, for each gene, the difference between the actual t-statistic and the average t-statistic over all permutations. Genes beyond a certain threshold of this difference are considered as being differentially expressed. Next, the number of false discoveries is estimated as follows. The smallest t-statistic in absolute value amongst these genes is noted down; call this $\delta$. Then, for each permutation, the number of genes achieving a t-statistic greater than $\delta$ in this permutation is counted. The average of this count over all permutations yields the number of false positives. There are three details which have been suppressed in the above description but which should be noted. First, a slight variant of the t-statistic is used where an additive term is applied in the denominator to dampen variations at low expression levels. Second, the permutations generated need to be *balanced permutations* (i.e., each permutation should mix arrays from the two groups in equal numbers). Finally, the $\delta$ value and the false positive counts are actually computed separately for induced and repressed genes to allow for some asymmetry in these situations.

### *Number of replicates and the number of false positives*

The one factor which is key in the success of all the statistical methods mentioned above is the number of replicates. Dow [27] concludes that at sample sizes lower than 10, the minimum detectable fold-change was higher than the changes induced by the treatment involved in their experiments. Of course, this number will vary depending upon the nature of the experiment. However, increasing the number of replicates can dramatically bring down the number of false positives, as shown by the analysis on the Latin Square Dataset in **Table 5**.

The question of how many replicates are needed has barely been explored in the literature. The answer depends on several parameters, including the type of statistical test performed, the difference in expression levels to be detected, the number of false positives desired, the cutoff probability, and potentially other unknown parameters associated with the test. For example,

**Highlights**

- Careful attention to analysis methods is needed to analyze microarray data. Wrong choices can increase the number of false positives by several-fold.
- While oligonucleotide arrays show less variability than cDNA arrays, oligonucleotide behavior at low expression levels is often choppy, with lot of noise. Using multiple probes (as in the case of Affymetrix arrays) and averaging over these probes does seem to attenuate this noise.
- Variations in the image analysis segmentation and background correction algorithm can often change the background corrected value of a spot substantially; thus, accurate spot segmentation as opposed to fitting spots with circles and robust background correction are important.
- Normalization across arrays and dyes is absolutely vital. Simple scaling and linear approaches are not sufficient and non-linear approaches are needed in several cases.
- On Affymetrix arrays, analysis methods based on the PM intensities seem to perform better than those based on subtracting the mismatch intensity from the perfect match intensity.
- Having sufficient replication in arrays is vital in reducing the number of false positives, though methods for predicting the number of replicates required for various array platforms have barely been studied.
- Fold-change is not a good measure of differential expression. More sophisticated statistical tests can give far more accurate results but these typically require a fair amount of familiarity with statistical analysis.
- Further comparative analysis of various algorithms is hampered by the lack of common benchmarks on all standard microarray platforms; establishing benchmark data and spike-in like data sets for all standard platforms can help in defining and standardizing analysis protocols.

to compute the number of replicates needed to have ten false positives at a p-value cutoff of 0.01 using a regular t-test, even assuming that the genes behave independently, one will need to know the variances in the expression level of each gene over experiments. For specific tests, there has been some research on identifying the unknown parameters associated with the test and then using those to determine the number of replicates needed, for example, see Pan *et al.* [28],

*Other tests and experiment types*

The t-test in its original form is a *parametric* test (i.e., it relies on the normality assumption). The Mann-Whitney rank-sum test [23] is a non-parametric test (i.e., it does not rely on the normality assumption) applicable to the two group situation. However, a larger number of replicates is typically desirable [16]. See Pan [29] for a review and comparative analysis of some more methods, including a regression modeling approach, a mixture modeling approach, and the SAM approach described above.

More complicated experimental designs need more sophisticated tests. For more than two groups, the parametric ANOVA test or the non-parametric Kruskal-Wallis tests are applicable. For the case when the groups involve multiple treatments on the same individuals, the paired t-tests and the Wilcoxon Sign-Rank test are used for two groups and the Repeated Measures test for multiple groups. See [23] for details on these tests. Finally, a special case of only one group of replicates arises for two-dye arrays. This is like a paired t-test where the t-statistic that is computed is simply as below:

$$t_g = \frac{m_1}{\sqrt{\frac{s_1}{n_1}^2}}$$

Here $m_1$ is the mean value within the group and $s_1$ is the corresponding standard deviation. Much of the above discussion on t-tests is applicable here as well, although the use of permutation tests as described above is not obviously applicable.

*Remarks*

Statistical testing for microarrays is a ripe research area with algorithms getting increasingly sophisticated; however, there do not seem to be any unique winners at this point. The absence of common benchmarks and standards makes it harder to compare these algorithms against each other.

## Outlook

The field of microarray analysis has moved beyond the initial simple approaches and is becoming increasingly sophisticated as more powerful algorithms are being used to increase sensitivity and specificity. Unfortunately, several of these algorithms require a fair amount of statistical expertise and are therefore inaccessible to a typical user. This will change with time as these procedures get more accurate and are standardized and incorporated in public and commercial tools, putting more of these techniques in the hands of the person running the experiment. However, there is a pressing need for setting up benchmarks on all standard microarray platforms and keeping these benchmarks current as time passes. Once standardization happens, the emphasis will shift from algorithms for determining differentially expressed genes accurately to algorithms for recreating the biological processes at a systemic level from this information.

## Bibliography


Papers of special note have been highlighted as either of interest (•) or of considerable interest (••) to readers.

1. Bassett DE Jr, Eisen NM, Boguski MS: Gene expression informatics – it's all in your mine. *Nat. Genet.* 21, 51-55 (1999).
2. Kane MD, Jatkoe TA, Stumpf CR, Lu J, Thomas JD, Madore SJ: *Nucleic Acids Res.* 28(22), 4552-4557 (2000).
3. Santalucia J Jr: A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl. Acad. Sci. USA* 95, 1460-1465 (1998).
4. Breslauer KJ, Frank R, Blocker H, Marky LA: Predicting DNA duplex stability from the base sequence. *Proc. Natl. Acad. Sci. USA* 83, 3746-3750 (1986).
5. Owczarzy R, Vallone PM, Gallo FJ, Paner TM, Lane MJ, Benight AS: Predicting sequence-dependent melting stability of short duplex DNA oligomers. *Biopolymers* 44, 217-239 (1997).
6. Yang YH, Buckley MJ, Dudoit S, Speed TP: Comparison of methods for image analysis on cDNA microarray data. *J. Comput. Graphical Stat.* 11, 108-136 (2002).

•• **Comprehensive survey of image analysis issues for spotted arrays.**

7. Adams R, Bischof L: Seeded region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16, 641-647 (1994).
8. Dudoit S, Yang H, Callow MJ, Speed TP: Statistical Methods for identifying genes with differential expression in replicated cDNA experiments. *Stat. Sin.* 12(1), 11-139 (2000).
9. Soille P: Morphological image analysis: principles and applications. Springer (1999).
10. Hartemink A, Gifford D, Jaakkola T, Young R: Maximum likelihood estimation of optimal scaling factors for expression array normalization In: *SPIE BIOS* (2001).
11. Irizarry, RA, Hobbs B, Collin F *et al.*: Exploration, normalization and summaries of high density oligonucleotide array probe level data. *Biostatistics* 4(2), 249-264 (2003).

•• **A good description of methods for analysing Affymetrix data.**

12. Durbin BP, Hardin JS, Hawkins DM: A variance-stabilizing transformation for gene expression microarray data. *Bioinformatics* 18, 105-110 (2002).
13. Huber W, von Heydebreck A, Sültmann H, Poustka A: Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics* 18(Suppl. 1), 96-104 (2002).
14. Munson P: A consistency test for determining the significance of gene expression changes on replicate samples and two convenient variance stabilizing transforms. GeneLogic Workshop on Low Level Analysis of Affymetrix GeneChip Data.
15. Schadt E, Li C, Eliss B, Wong WH: Analysing high-density oligonucleotide gene expression array data. *J. Cell Biochem.* 84(37), 120-125 (2000).
16. Hoffmann R, Seidl T, Dugas M: Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis. *Genome Biol.* 3(7) 0033.1-0033.11 (2002).
17. Hill AA, Brown EL, Whitley MZ *et al.*: Evaluation of normalization procedures for Oligonucleotide array data based on spiked cRNA controls. *Genome Biol.* 2 0055.1-0055.13 (2001).
18. Bolstad BM, Irizarry RA, Astrand M, Speed TP: A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19, 2, 185-193 (2003).

•• **Comparison of various normalization methods.**

19. Yang YH, Dudoit S, Luu P *et al.*: Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.* 30, 4 (2002).
20. Li C, Wong WH: Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc. Natl. Acad. Sci. USA* 98, 31-36 (2000).

•• **Description of the popular Li-Wong method for analysing Affymetrix arrays.**

21. Li C, Wong WH: Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biol.* 2(8) 0032.1-0032.11 (2001).
22. Tusher V, Tibshirani R, Chu G: Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl. Acad. Sci. USA* 98, 5116-5121 (2001).
23. Glantz S: *Primer of Biostatistics (5th edition).* McGraw-Hill (2002).
24. Long AD, Mangalam HJ, Chan BY, Tolleri L, Hatfield GW, Baldi P: Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework. Analysis of global gene expression in *Escherichia coli* K12. *J. Biol. Chem*. 276(23), 19937-19944 (2001).
25. Westfall PH, Young SS: Resampling based multiple testing. John Wiley & Sons, New York (1993).
26. Benjamini B, Hochberg Y: Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B.* 57, 289-300 (1995).
27. Dow GS: Effect of sample size and p-value filtering techniques on the detection of transcriptional changes induced in rat neuroblastoma (NG108) cells by mefloquine. *Malar. J.* 2(1), 4 (2003).
28. Pan W, Lin J, Le CT: How many replicates of arrays are required to detect gene expression changes in microarray experiments? A mixture model approach. *Genome Biol.* 3(5) 0022.1-0022.10 (2002).
29. Pan W: A comparative review of statistical methods for discovering differentially expressed genes in replicated microarray experiments. *Bioinformatics* 12, 546-554 (2002).


## Websites


101. http://www.affymetrix.com/analysis/download_center2.affx
Affymetrix Latin Square Data.
102. http://qolotus02.genelogic.com
Gene Logic Latin Square Data.
103. http://www.mwg-biotech.com/html/d_support/d_faq.shtml
MWG Biotech.
104. http://www.strandgenomics.com/products/sarani/slast.html
Strand Genomics Sarani.
105. http://www.strandgenomics.com/products/sarani/overview.html
Strand Genomics Sarani.
106. http://www.operon.com/arrays/poster.php
Bosch JT, Seidel S, Batra, Lam H, Tuason N, Saljoughi S, Saul R: Validation of sequence-optimized 70 base oligonucleotides for use on DNA microarrays [Poster].
107. http://experimental.act.cmis.csiro.au/Spot/index.php
SPOT.
108. http://www.strandgenomics.com/products/chitraka/overview.html
Strand Genomics Chitraka.
109. http://ftp.isds.duke.edu/WorkingPapers/02-05.html
Zuzan H, Blanchette C, Dressman H *et al.*: Estimation of probe cell locations in high-density synthetic-oligonucleotide DNA microarrays [Working Paper]. Institute of Statistics and Decision Sciences.
110. http://www.affymetrix.com/support/technical/whitepapers/sadd_whitepaper.pdf
Statistical Algorithms Description Document, Affymetrix, Inc.
111. http://www.biostat.harvard.edu/complab/dchip
DChip: The DNA Chip Analyzer.

112. http://www.bioconductor.org
The Bioconductor webpage.
113. http://stat-www.berkeley.edu/users/terry/zarray/html/log.html
Speed T: Always log spot intensities and ratios, Speed Group Microarray Page.
114. http://www.itl.nist.gov/div898/handbook/pmd/section1/pmd144.htm
The Lowess method.
115. http://www-stat.stanford.edu/~tibs/SAM/
Significance Analysis of Microarrays.
116. http://www.strandgenomics.com/products/soochika/overview.html
Strand Genomics Soochika.