Faster cyclic loess: normalizing DNA arrays via linear models

Karla Ballman
Diane Grill
Ann Oberg
Terry Therneau

Technical Report #68
November 2004

Abstract

Motivation: Our goal was to develop a normalization technique that yields results similar to cyclic loess normalization and with speed comparable to quantile normalization.

Results: Fastlo yields normalized values similar to cyclic loess and quantile normalization and is fast; it is at least an order of magnitude faster than cyclic loess and approaches the speed of quantile normalization. Furthermore, fastlo is more versatile than both cyclic loess and quantile normalization because it is model-based.

Availability: The Splus function for fastlo normalization is available from the authors.

1 Introduction

High-density gene expression array technology allows investigators to obtain accurate quantitative measurement of the expression levels for tens of thousands of genes in a biological specimen. There are two major types of microarray technology. The spotted cDNA microarray technology uses a base sequence matching all or part of a gene attached to a (usually glass) slide. Two different biological samples, one labeled with red dye and the other labeled with green dye, are hybridized on the slide. Using a fluorescence microscope, the intensities reflecting the amount of RNA hybridized to each site (spot) are measured. Investigators often use the log(red/green) ratio intensities of RNA.

The other major type of microarray technology is oligonucleotide arrays. One of the most commonly used platforms is the Affymetrix GeneChip. This technology uses 11-20 probe pairs, called a probe set, to represent a gene. Each probe pair consists of a perfect match (PM) and mismatch (MM) probe. The PM probe is designed to match a subsequence of 25 bases of the gene. The MM probe is identical to the PM probe except the middle (13th) base, which has been changed so it does not match. A biological sample is hybridized on the microarray and the RNA expression for the gene is determined by some function of the PM and MM probes. Examples of methods for obtaining the RNA expression level from the probe set are a robust average difference [1], the Robust Multi-chip Average [2], the Model Based Expression Index [3], and a linear mixed models approach [4].

Expression data obtained from any type of microarray technology has measurement error or variation. One type of variability is due to biological differences between specimen samples. This is what is of interest to the investigator; the investigator would like to know which genes are differentially expressed among different biological samples (e.g. between cancerous and normal kidney tissues, among B-cells challenged with different agents in culture, etc.). Systematic variation also affects the measured gene expression level. Many sources contribute to this type of variation and are found in every microarray experiment. Sources include, but are not limited to, the array manufacturing process, the preparation of the biological sample, the hybridization of the sample to the array, and the quantification of the spot intensities. Hartemink et al. [5] provide a more complete discussion of the sources of systematic variation in the microarray experimental process. The purpose of normalization is to minimize the systematic variations in the measured gene expression levels among different array hybridizations to allow for the comparison of expression levels across

arrays and so that biological differences can be more easily identified.

There are numerous methods for normalizing gene expression data. Generally, normalization methods use a scaling function to correct for experimental variation. These functions are applied to the raw intensities of the spots (gene sequence on spotted arrays or oligonucleotide probes on the GeneChip array) on the microarray to produce normalized or scaled intensities. Types of normalization techniques include mean correction [6], non-linear models [7], linear combination of factors [8], and Bayesian methods [9]. There is compelling evidence that non-linear normalization methods, which are not dependent upon the choice of a baseline array, perform the best [10]. We tend to favor two commonly used non-linear methods: cyclic loess normalization and quantile normalization. Both techniques are nonlinear and perform normalization on the set of arrays as a whole without specifying a reference array. Overall, we prefer cyclic loess because it is not as aggressive in its normalization as is quantile normalization; however, cyclic loess is relatively slow for even a moderate sized set of arrays. Quantile normalization, on the other hand, is very fast for even large sets of arrays. The goal of our investigation was to develop a method that produced normalized values similar to that of cyclic loess but would be considerably faster—on the order of the speed of quantile normalization. The result is a new normalization method called fastlo. Since normalization is performed on the raw intensity values of the spots on the arrays, the methods discussed here are applicable to both major types of array technology: spotted cDNA and oligonucleotide. However, our focus is on data arising from GeneChip arrays and it should be noted that there are additional considerations when normalizing two-color spotted cDNA arrays [7].

In the next section we introduce cyclic loess normalization and useful insights that can be gained from viewing it as a parallel algorithm. Section 3 describes a new type of normalization, fast linear loess (fastlo), arising from the observation that cyclic loess is essentially a smoothing function coupled with a very simple linear model. In Section 4, we introduce quantile normalization. The performances of the three different normalization techniques are compared in Section 5. Comparisons are made using two simulated data sets and a real dataset, one of the benchmark sets for Affymetrix GeneChip expression measures [11]. The simple linear model underlying the fastlo method is extended in Section 6. We close with a discussion of our findings in Section 7.

2 Cyclic normalization

Most methods of normalization, including cyclic loess and the methods introduced later, assume that the vast majority of the genes do not change expression levels under the conditions being studied. In other words, they assume that the average (geometric mean) ratio of expression values between two conditions is 1 or equivalently, the average (arithmetic mean) log ratio of expression is zero for a typical gene. This is biologically plausible for many studies. However, if there is good reason to believe this assumption is not true for a particular study, then the normalization methods described here are not appropriate.

2.1 Cyclic loess

A fundamental graphical tool for the analysis of gene expression array data is the M versus A plot (MA plot); here M is the difference in log expression values and A is the average of log expression values [12]. Figure 1 contains an MA plot for a random sample of 10,000 probes from two GeneChip arrays with a loess smoother superimposed. These arrays have not been normalized. The MA plot for ideally normalized data would show a point cloud scattered about the $M = 0$ axis. In other words, the loess smoother would be a horizontal line at 0 for ideally normalized data.

Cyclic loess normalizes two arrays at a time by applying a correction factor obtained from a loess curve fit through the MA plot of the arrays, call the curve $f(x)$. For example, consider the circled point in Figure 1. This point corresponds to a spot i on each array. On one array, the observed expression level at this spot would be reduced by $1/2$ the distance of $f(x)$ from the $y = 0$ line; in other words, $f(x)/2$ would be subtracted from the expression level of this spot on one of the arrays. The expression value for this spot on the other array would be increased by $f(x)/2$. After correction, the MA plot for this particular pair would be horizontal. One pass of the cyclic loess algorithm consists of performing this pairwise normalization on all distinct pairs of arrays. Passes of the algorithm continue until the computed corrections of a completed pass are essentially zero.

In summary, to perform the cyclic loess algorithm begin with the $\log_2$ of the spot expression intensities arranged as a matrix with one column per array and one row per array spot and proceed through the steps below.

1. Choose two arrays and generate an MA plot of the data. The x-axis is the mean probe expression value of the two arrays and the y-axis is the difference (one point for each spot).

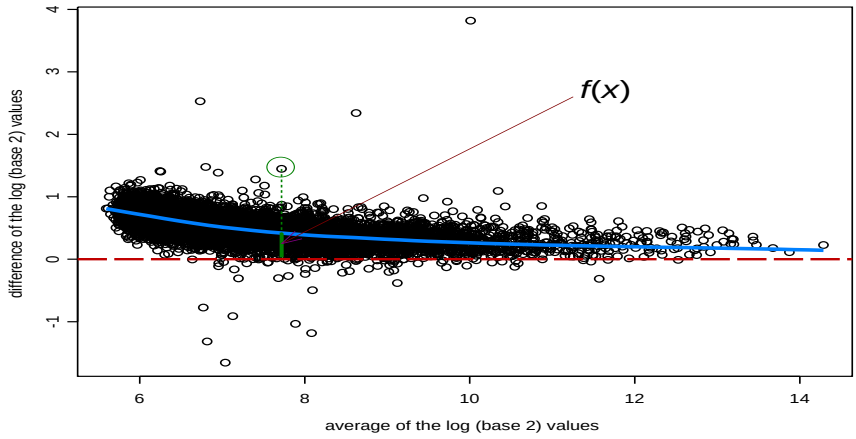

Figure 1: MA plot for 10,000 randomly selected probes.

2. Fit a smooth loess curve $f(x)$ through the data.
3. Subtract $f(x)/2$ from the first array and add $f(x)/2$ to the second.
4. Repeat until all distinct pairs have been compared.
5. Repeat until the algorithm converges.

In practice, the pairs of arrays are chosen by a method that systematically cycles through all pairs. For example, say there were five arrays to be normalized as a group: A, B, C, D, and E. Normally the method for cycling is the obvious: AB, AC, AD, AE, BC, BD, ..., DE. One drawback with cyclic loess is the amount of time required to normalize a set of data; the time grows in an exponential manner as the number of arrays increases. Typically, two or three passes through the complete cycle are required for convergence. Likely, cyclic loess would converge faster if pairings went in a more balanced order. However, the time savings would not be considerable because a loess smooth would still be required for a relatively large number of array pairs.

Further examination of the cyclic loess algorithm reveals some interesting facts. First, the algorithm preserves the row means of the data matrix, Y. At any given step, one number in a row is increased by $f(x)/2$ and another is decreased by the same amount. Second, if all the values in one of the columns is increased or decreased by a constant, the final results of the algorithm (the scaled intensities on each array) are changed only by the addition of a constant.

This is because any one of the plots on which the smooths are based is identical but for the labeling of its axes, and thus any given smooth is changed only by a constant. Finally one pass through the algorithm requires C_2^n loess smooths on all the spots on the array.

2.2 Parallel loess

Cyclic loess is inherently parallel in nature. Viewing it in this manner may provide insights that allow computational time savings. Imagine that we had a parallel machine, so that all the pairwise normalizations could be done simultaneously. Once each of the pairwise corrections is obtained, then the correction for spot i on array j would be the 'average' correction for this spot across all array pairs containing array j. Essentially, the correction to the spots on array j would be the average of all computed corrections of the spots for pairs containing array j. To simplify the logic, we consider the pairing of each array with itself, as well as both orderings of the pairs. This means that for n arrays, there are n^2 pairings. Here, and throughout the remaining text, Y is used to denote the matrix of the $\log_2$ intensity values where the column j corresponds to an array and row i corresponds to a spot.

As the simplest example, consider the noiseless case where each column of Y, corresponding to an array, differs from any other by a constant and array '0' is an imaginary reference representing the true expression level. In other words, the intensity of the spot i, j, y_{ij}, can be expressed as $y_{ij} = y_{i0} + c_j$. The ideal correction of all spots for a given chip is $c_j - \bar{c}$ (a horizontal loess curve) and the average correction for array 1 from the parallel algorithm is

$$(1/n)\sum_{j=1}^{n}(c_1 - c_j)/2 = (c_1 - \bar{c})/2$$

That is, the average correction is 1/2 of what it should be. As a result, we define the correction step for the ith chip in parallel loess to be $(2/n)\sum_j f_{ij}$, where f_{ij} is the smooth for the plots of chips i and j.

Now consider a simple simulation where each column of the data matrix (each array) differs from any other by a constant plus a symmetrically distributed noise term. Array '0' is again the set of 5000 true intensities, which are randomly selected from a uniform distribution with range from 0 to 10. The $\log_2$ intensity levels for the four arrays in the experiment, y_{ij}, are derived from array '0' as follows: $y_{ij} = y_{i0} + j + e_{ij}$, where $e_{ij} \sim t_8$ ($i = 1, 2, \ldots, 5000$ and $j = 1, 2, 3, 4$). A t-distribution was selected for the error term (noise distribution) because it

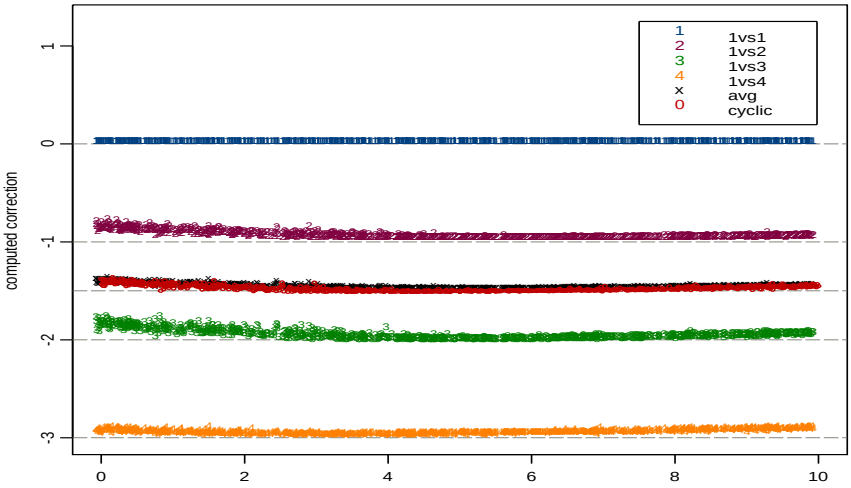

Figure 2: The computed corrections for array 1 from the 4 pairwise smooths for a parallel loess (1v1, 1v2, 1v3, 1v4), for the average of the pairwise corrections (avg), and the for cyclic loess (cyclic).

is symmetric and produces more outliers than a normal distribution. In this case, the true correction for array j is $j - \bar{j}$ or $j - 2.5$ ($j = 1, 2, 3, 4$). The four pairwise corrections for a particular spot involving array 1 have expectations of $0, -1.0, -2.0$, and -3.0. The average of the four corrections for a particular spot is -1.5. Figure 2 shows the four computed corrections for each spot from parallel loess involving array 1 as well as the average of the corrections across arrays. This suggests averaging the corrections to get an overall update, rather than applying each of the separate corrections to the data in turn. Figure 2 also includes the corrections for array 1 after applying cyclic loess. For this case performing the smooths cyclically or in parallel produces equivalent results. For this figure, and all remaining figures, a subset of randomly selected points are plotted to thin the plot and better display the relevant structure.

Using the parallel version would probably be faster than cyclic loess. However, we did not intend to obtain computational speed savings by developing a parallel version of cyclic loess. Rather, we used this construct to gain insight into how cyclic loess works and how it might be made faster for non-parallel machines.

Parallel loess can be shown to be unbiased for this simple case. Let $y_{ij} = \alpha_i + \beta_j + \epsilon_{ij}$; α_i are the true probe values (y_{i0} in the simulation), β_i the simple

array effects, and ϵ_{ij} the error. Assume that a, b are two vectors of constants, and a smooth of Yb on Ya is to be used to estimate the correction. Since the correction should be constant, we want Ya and Yb to be uncorrelated, i.e., that there be no linear bias in the smooth. The covariance matrix of Y has elements of $\sigma_\alpha^2 + \sigma_\epsilon^2$ on the diagonal, and σ_α^2 off the diagonal, where these are the variances of α and ϵ above. Simple algebra shows the covariance of Ya and Yb to be $\sigma_\epsilon^2 \sum a_j b_j + \sigma_\alpha^2 (\sum a_j)(\sum b_j)$. Since the two variances are unknown for a given problem, linear bias in the plot is avoided choosing a and b so that both terms are zero. For parallel loess, the plot for arrays 1 and 2 has $a = (1/2, 1/2, 0, \ldots, 0)$ and $b = (1, -1, 0, \ldots, 0)$, clearly satisfying the criteria, and likewise for any other pair of arrays i, j. Other choices of a and b will be explored below.

2.2.1 Parallel loess variant one

An obvious variant of the parallel cyclic loess algorithm is to replace the n loess smooths, each on p points, with a single loess smooth on np points. In other words, the corrections to be applied to spots on array j would be obtained by placing all the points in the MA plots containing array j (n of them) into a single plot and performing a single loess smooth on this plot. Recall, the corrections for array j are obtained in parallel loess by averaging the n pairwise corrections, obtained from the n MA plots. Also, recall that the corrections are just twice the smooth value. So if the smoother is a linear operator, which loess is other than outlier rejection passes, then the average of the smooths will be equal to a single smooth applied to a plot with all the derived data points, i.e. the points from the n MA plots involving array j all placed on a single plot. Because of the very large amount of data on microarray gene expression arrays, outlier rejection is not normally an important issue. Specifically, when there is a large quantity of data over a given range, no point, even if extreme, can exert much influence.

Figure 3 compares the computed corrections for the array spots on two arrays from our four array example introduced above: true expression values for the array shifted by a constant plus symmetrically distributed error. The computed corrections displayed are those obtained from cyclic loess, from the pure parallel version of cyclic loess, and from a single loess smooth on the plot containing all the points from the four MA plots involving the array to be normalized (parallel variant 1). So that this variant of parallel loess performs similarly to the pure parallel version, the points from $n-1$ of the n MA plots must be shifted horizontally as a group. This is necessary to approximately align the points that

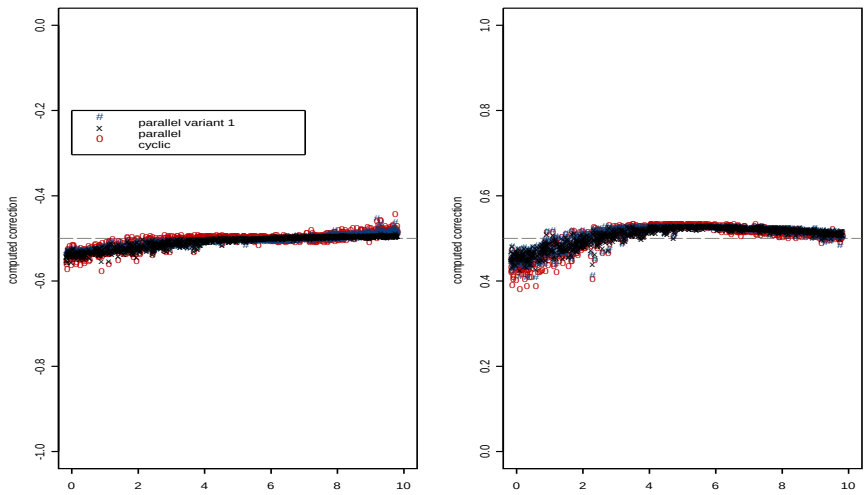

Figure 3: The computed corrections for arrays 2 and 3 from the example of 4 arrays. The corrections are from a single smooth on all the derived 20,000 data points (parallel variant 1), from parallel loess (parallel),and from cyclic loess (cyclic).

correspond to a particular spot on the array vertically on the plot. Recall that for the pure parallel version of cyclic loess, the correction for a particular point is found by averaging the corrections obtained from the n MA plots. As expected, all three methods produced equivalent results.

This parallel variant 1 reduces the number of loess smooths that are computed. One pass through the data with parallel variant 1 requires only n loess smooths compared to C_2^n loess smooths required by cyclic loess. However, each smooth for this parallel variant 1 is performed on np versus p points. Whether the implementation of this version of parallel loess is faster than cyclic loess likely depends on the implementation details of each algorithm.

2.2.2 Parallel loess variant two

The next idea is to reduce the number of points in the plot containing all the points from the MA plots of parallel loess involving array j from np to p. Since there are p spots on an array, the intent is to replace each collection of n points per spot i (one per each pairwise MA plot—array j with array 1, array j with array 2, ..., array j with array n) with a single point, which summarizes the collection of n points for spot i. The new plot would contain p 'summary' points

and we would perform a smooth on this plot to obtain the spot corrections for array j.

A way to produce a point for spot i that summarizes the n points for spot i that involve array j is to set the x-coordinate equal to the average of the n x-coordinates:

$$\frac{1}{n}\left(\frac{(y_{i1}+y_{i1})}{2}+\frac{(y_{i1}+y_{i2})}{2}+\cdots+\frac{(y_{i1}+y_{in})}{2}\right)=\frac{y_{i1}+\bar{y}_{i\cdot}}{2}$$

Here $\bar{y}_{i\cdot}$ is the row mean for the ith row of the data, which is equal to the mean expression value of spot i across the arrays. The vertical position for spot i on array j is the average of the y-coordinates:

$$\frac{1}{n}\left[(y_{i1}-y_{i1})+(y_{i1}-y_{i2})+\cdots+(y_{i1}-y_{in})\right]=y_{i1}-\bar{y}_{i\cdot}\,.$$

The next step is to determine corrections for the spots on chip j by fitting the loess smooth on this set of p points. This would be repeated for each of the $n-1$ other chips; a total of n smooths need to be computed for one pass through all the data, similar to the parallel loess variant 1 above. However, the number of points for each loess smooth is p for this variant compared to np for variant 1.

Repeating the bias computation found in Section 2.2, the plot for array 1 versus array 2 has vertical and horizontal axes aY and bY, respectively, with $a=(1-1/n,-1/n,\ldots,-1/n)$ and $b=(1+1/n,1/n,\ldots,1/n)/2$. This yields $\sum a_j=0$, $\sum b_j=1$ and $\sum a_jb_j=(n-1)/n$ leading to positive correlation between aY and bY and a potentially biased correction. Figure 4 compares the computed corrections produced by this variant of parallel loess (parallel variant two), parallel loess, and cyclic loess for two of the four arrays in our simple data example. Clearly, the bias is severe. In the next section we use a linear models argument to motivate a similar plot, but with x-coordinate of $\bar{y_{i\cdot}}$, leading to $b=(\frac{1}{n},\ldots,\frac{1}{n})$ and $\sum a_jb_j=0$ and a unbiased estimate.

3 Fast linear loess

Cyclic loess can be conceptualized as a smooth (loess), coupled with a very simple linear model. Consider the case of two arrays, 1 and 2, with gene expression levels represented by y_{i1} and y_{i2} for $i=1,\ldots,p$, and the simplest possible linear model for the data $y_{ij}=\alpha_i+\epsilon_{ij}$, an intercept for each spot. The solution is of course $\hat{\alpha}_i=\bar{y}_{i\cdot}$. In the MA plot for the two arrays, the x-coordinate is $(y_{i1}+y_{i2})/2=\bar{y}_{i\cdot}=\hat{y}_{i\cdot}$, the y-coordinate is $y_{i1}-y_{i2}=2(y_{i1}-\hat{y}_{i\cdot})$, and the

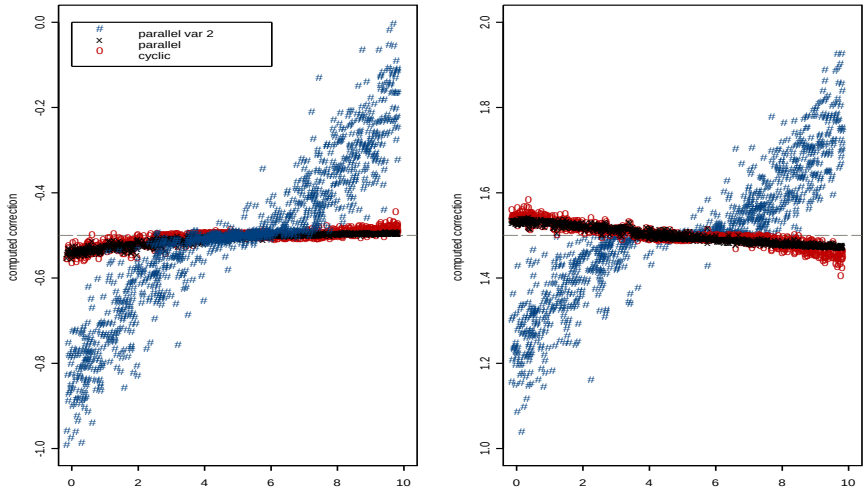

Figure 4: The computed corrections for arrays 2 and 4 of the simulated example of 4 arrays. The corrections are from the smooth of the 5000 derived points (parallel var 2), from parallel loess (parallel), and from cyclic loess (cyclic).

adjustment to spot i on the the first chip is 1/2 the height of the loess smooth on this plot at the x-coordinate corresponding to spot i.

Extending this idea to the n array case suggests creating a modified MA plot for each array j. The modified MA plot consists of array j and a constructed 'average' array; the intensity level of spot i on the 'average' array is equal to the average intensity of spot i across all arrays for $i = 1, \ldots, p$. The computed corrections for array j would be obtained from the loess smooth on the points of this plot. This would be done for each of the $j = 1, \ldots, n$ arrays. This variant of cyclic loess, called *fastlo*, requires n loess smooths each on p points for one pass through the data. Obviously, this is considerably faster than cyclic loess, which requires C_2^n loess smooths. Note that this is the same algorithm as parallel variant 2 with the modification to remove the bias; in other words, it is the same as the parallel variant 2 algorithm that uses $\bar{y}_{i\cdot}$ as a summary x-coordinate.

The steps of fastlo given below are done on the $\log_2$ of the spot intensities:

1. Create the vector $\hat{y}_{i\cdot}$ = the row mean of Y. Note that this is the same as creating an 'average' array.

2. Plot $\hat{y}$ versus $(y_i - \hat{y})$ for each array j. This plot has one point for each spot (modified MA plot).

3. Fit a loess curve $f(x)$ through the data.

4. Subtract $f(x)$ from array j.

5. Repeat for all remaining arrays.

6. Repeat until the algorithm converges.

A further interesting aspect of fastlo is that it requires only 1 or at most 2 iterations to converge. If there is no outlier down weighting, then the loess smoother will be a linear operator and the average of the smooths is the smooth of all the points:

$$\frac{1}{n}\sum Sm(y_n - \bar{y}) \approx Sm(\sum_n y - \bar{y}) = Sm(0) = 0$$

where Sm is the smoother. If this holds, then for any given row of Y, some elements increase and some decrease, but the mean stays the same. If the row means do not change, the algorithm has converged.

4 Quantile normalization

Quantile normalization makes the overall distribution of values for each array identical, while preserving the overall distribution of the values. It consists of 2 steps.

1. Create a mapping between ranks and values. For rank 1 find the n values, one per array, that are the smallest value on the array, and save their average. Similarly for rank 2 and the second smallest values, and on up to the n largest values, one per array.

2. For each array, replace the actual values with these averages.

A simple example is given below, for an experiment with only 5 spots and 3 arrays. First find the smallest values:

	Array 1	Array 2	Array 3
Gene 1	10	29	7
Gene 2	30	37	27
Gene 3	21	24	25
Gene 4	35	36	26
Gene 5	8	15	1

They are 8, 15, and 1, all for gene 5. Now replace them with their average of 8, and find the next set to replace:

	Array 1	Array 2	Array 3
Gene 1	10	29	7
Gene 2	30	37	27
Gene 3	21	24	25
Gene 4	35	36	26
Gene 5	*8*	*8*	*8*
	20.8	28.2	16.6

These three values, not all for the same gene, will be replaced with (10+24+7)/3 = 13.67. The process continues until the three largest values of 35, 29 and 27 have been replaced with their average. As mentioned, this produces identical distributions of values on each array; quite an aggressive normalization process. On the other hand, quantile normalization is extremely fast—it only requires a sort of the arrays and a computation of means across sorted rows. It completes in a single pass through the data.

5 Results

As a final step, we compare the performance of the three normalization algorithms: cyclic, quantile, and fastlo. Results of applying these algorithms to three datasets are presented. Two of the datasets are simulated data and the third is a subset from the GeneLogic dilution experiment. Descriptions of the datasets and a comparison of the results follow below.

5.1 Simulated dataset one

The first simulated dataset was introduced in Section 2.2. Recall that there are four arrays and each array has 5000 spots and that the true correction to be applied to all spots on array j should be a constant, $j - 5/2$, for $j = 1, 2, 3, 4$.

Figure 5 contains a comparison of the computed corrections for each of the three methods for one of the four arrays, the three arrays not shown behave similarly. All three methods appear to be performing similarly with their computed corrections close to what the true correction should be, depicted by the dashed line. The normalized values using each of the three methods on the set of four arrays is shown in Figure 6. The distribution of the values from the quantile normalization is identical for all four arrays, as it should be. Cyclic loess and

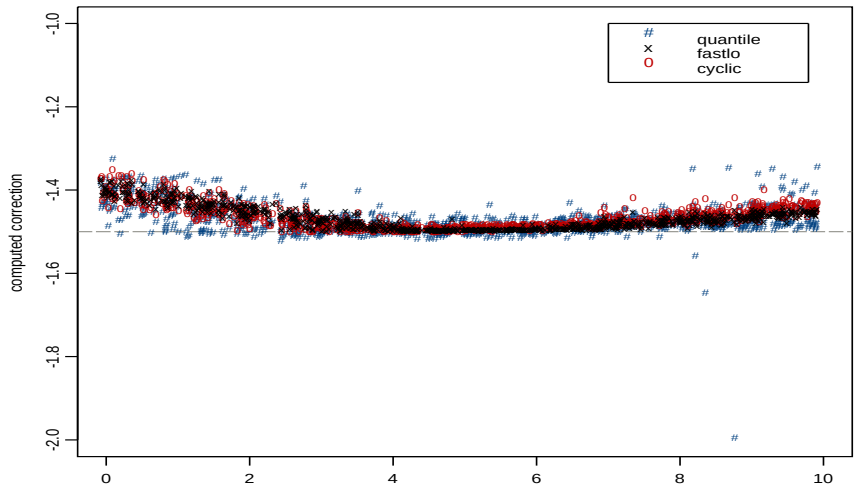

Figure 5: The computed corrections for array 1 of simulated dataset 1. The corrections are from cyclic loess (cyclic), quantile (quantile), and fastlo (fastlo). The dashed line represents the true correction value for each spot.

fastlo appear to produce equivalent normalizations and these are similar to what is produced by quantile normalization.

In Figure 5, quantile normalization appears to have the most variability among its computed corrected values. The standard deviations of the computed corrections around the true value (the horizontal dashed line) are 0.016, 0.017, and 0.031 for fastlo, cyclic loess, and quantile normalization, respectively. However,looking at the corrections alone is not sufficient, the variation in the corrections needs to be compared to the underlying offsets themselves and to the standard deviation due to biologic variability, both of which are of order one for this data. This is seen in figure 6, which displays the distributions of the normalized values. The MSEs across the set of arrays are 1.304, 1.305, and 1.304 for fastlo, cyclic, and quantile normalization; all three methods have essentially the same performance on average. In fact, all three methods are relatively perfect in that the variation that remains is that of the variance of a t-distribution with 8 df. This is also reflected in Table 1, which displays summaries of the standard deviations of the final spot intensities about their means.

In terms of computation time, quantile normalization is the fastest. It requires four sorts, each on 5000 values, followed by the computation of 5000 means of four values. These operations can be done very quickly. Fastlo is the

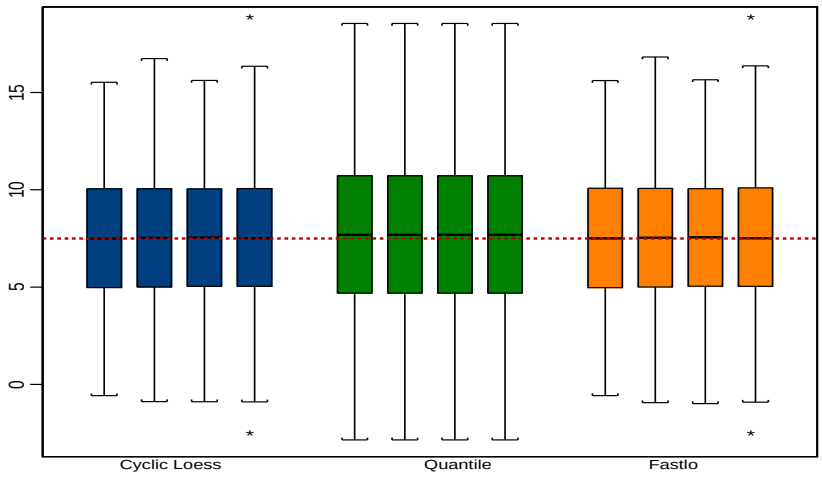

Figure 6: Boxplots of the normalized values for the four arrays of the toy example. The normalizations are from cyclic loess, quantile, and fastlo. The dashed line is the average of the ideal normalized values for the spots.

	min	Q1	median	Q3	max	mean
cyclic	0.0467	0.6734	0.9567	1.3097	3.8738	1.0289
quantile	0.0251	0.6750	0.9562	1.3125	3.8793	1.0289
fastlo	0.0578	0.6733	0.9570	1.3115	3.8509	1.0289

Table 1: Five number summary plus the mean for the distribution of standard deviations of the 5000 spot intensity values across the 4 arrays.

next fastest—it requires the computation of 5000 means of four values and then 4 loess smooths, each on 5000 points, for one pass through the data. Finally, cyclic loess normalization requires the most computations. For one pass through the data, it requires 6 loess smooths, each on 5000 points. For this problem, both fastlo and cyclic loess required two passes through the data before the normalized values sufficiently converged.

5.2 Simulated dataset two

The spot intensity values used in the simulated data above are not reflective of the distribution of the spot intensity values seen in real data. Likewise, the assumption of a symmetric error structure is not realistic. Simulated data set 2 was generated from a set of 5 arrays from the GeneLogic dilution exper-

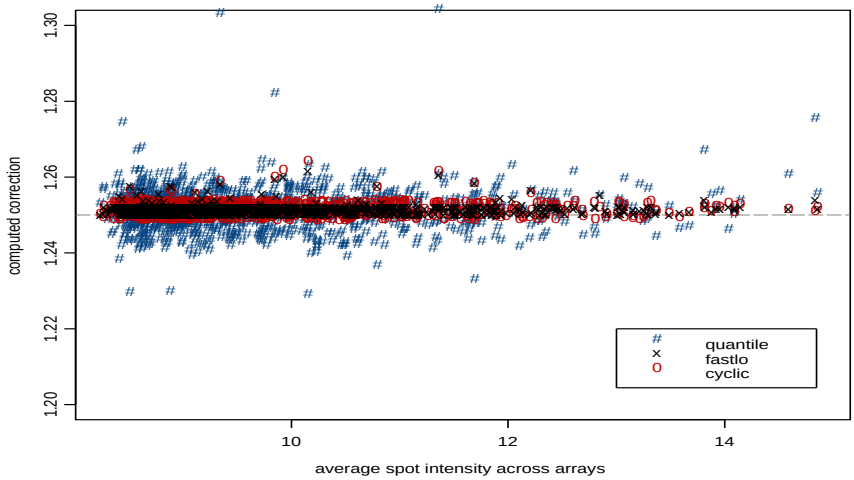

Figure 7: The computed corrections for array 8 of simulated dataset 2. The corrections are from cyclic loess (cyclic), quantile (quantile), and fastlo (fastlo). The dashed line represents the true correction value for each spot.

iment (http://www.genelogic.com/media/studies/dilution.cfm). The 5 arrays used were hybridized to human liver tissue with a total cRNA concentration of 5 μg; this set will be denoted as *liver5*. The simulated dataset has 10 arrays and each array has 20,000 spots. The 20,000 spots were randomly selected from among the PM spots on the Affymetrix U95Av2 GeneChip. The spot i intensity on array j was generated as $y_{ij} = y_{i,rand(liver5)_j} + j/2$. The value $y_{i,rand(liver5)_j}$ is randomly selected from among the group of five spot i values from the arrays in the *liver5* set. A constant, $j/2$, is added to spot i on array j for all $i = 1, 2, \ldots, 20000$ spots and all $j = 1, 2, \ldots, 10$ arrays.

As for the simulated data set in the previous section, the true correction of the systematic bias is a constant for array j, $j = 1, 2, \ldots, 10$. In this case, the constant for array j is equal to $j/2 - 2.75$. Again, all three algorithms yield similar results in terms of correcting for the known constant offset (Figure 7, arrays not shown yield similar results). The distributions of normalized values of the 10 arrays yielded by each of the three methods were also essentially equivalent (results not shown). As before, quantile normalization was the fastest followed by fastlo and the slowest was cyclic normalization; the ratios of CPU time required were $1.0 : 1.1 : 16.1$ for quantile:fastlo:cyclic loess.

For this simulated setting, we know the true amount of systematic error

	$\bar{x}$			s		
	quantile	cyclic	fastlo	quantile	cyclic	fastlo
array 1	−2.25	−2.25	−2.25	0.007	0.002	0.002
array 2	−1.75	−1.75	−1.75	0.008	0.003	0.003
array 3	−1.25	−1.25	−1.25	0.008	0.003	0.004
array 4	−0.75	−0.75	−0.75	0.006	0.002	0.002
array 5	−0.25	−0.25	−0.25	0.009	0.005	0.004
array 6	0.25	0.25	0.25	0.008	0.003	0.002
array 7	0.75	0.75	0.75	0.009	0.005	0.004
array 8	1.25	1.25	1.25	0.007	0.002	0.001
array 9	1.75	1.75	1.75	0.007	0.004	0.003
array 10	2.25	2.25	2.25	0.008	0.002	0.003

Table 2: Means ($\bar{x}$) and standard deviations (s) of the 20,000 spot intensity corrections produced by each of the three normalization methods when applied to the 10 arrays of simulated dataset 2.

needed to be removed from each array via normalization. A normalization technique that performed well would yield unbiased corrections and corrections with the smallest standard deviation. Results in Table 2 reveal all three methods to be unbiased. Fastlo appears to yield the smallest standard deviations, in general, followed closely by cyclic loess and quantile normalization having the largest. For practical purposes, the differences are so small as not to matter.

5.3 GeneLogic dilution data

A final comparison of the three methods was done on real data from the GeneLogic dilution experiment. The data consists of six groups of human liver cRNA at concentrations of 1.25, 2.5, 5.0, 7.5, 10.0, and 20.0 μg total cRNA. Five replicate arrays were made for each concentration level. See Irizarray et al. [2] for more details; the data can be obtained through the GeneLogic website (http://www.genelogic.com/media/studies/dilution.cfm). These data were normalized by each of the three normalization methods under investigation: cyclic loess, quantile, and fastlo. Each group of five arrays were normalized together and we only used the spot intensities corresponding to the PM probes. A comparison of the boxplots of the normalized values (not shown) revealed essentially equivalent distributions of the normalized spot intensities generated by the three methods.

As in the simulated data sets, we expect that the true spot intensity levels

should be equal across the replicate arrays for a given dilution concentration. For this situation, a desirable quality of a normalization technique would be one that yields the smallest standard deviation at a spot i across the five arrays, for all i. Table 3 summarizes the distribution of the standard deviations of all the spots (there were 201800 total spots on each array). These summaries are remarkably similar for all three methods. Quantile normalization tended to produce the largest standard deviation on average (either measured by the median value or the mean value) and cyclic loess and fastlo having the smallest in many of the sets. The largest differences were in the maximum of the distribution of the standard deviations across the spots, but there was no identifiable pattern in these differences in terms of one method always producing the largest or smallest maximum value. Overall, the differences in these distributions are probably not of practical significance. Interestingly, the distributions of the standard deviations did not appear to differ in identifiable ways among the different concentration levels; specifically, the standard deviations in spot intensities did not appear to increase as the concentration levels of the total cRNA increased.

6 Simple extension of fastlo

When normalizing arrays that are to be compared, a decision is made as to whether to normalize the entire set or to normalize separately within biologically identifiable subsets. For example, consider a situation where expression levels are to be compared between two distinct groups such as cancerous tissues and normal tissues. In this case, the entire set of arrays could be normalized together or separate normalizations could be done for the set of cancerous tissue arrays and for the set of normal tissue arrays. Determination of the most appropriate normalization approach depends upon which process is responsible for most of the systematic variation in expression levels among the arrays: the biological process of interest or the microarray experimental process. For instance, assume that the global spot average was 12.0 in the arrays from cancer tissues and 11.4 from the normal tissues. Do we ascribe this overall difference to the treatment (e.g. more genes are expressed in the cancer tissues) or to the arrays (variation in experimental conditions that produced the arrays)? If most of the systematic variation is believed to arise from biological mechanisms of interest, then it is reasonable to perform a separate normalization for each group being compared. On the other hand, if it is believed that most of the systematic variability is due to the experimental processes (and so not of interest), then it is reasonable

		min	Q1	median	Q3	max	mean
1.25	quantile	0.0035	0.0797	0.1066	0.1377	4.0684	0.1135
	cyclic	0.0052	0.0788	0.1055	0.1361	4.1000	0.1123
	fastlo	0.0050	0.0794	0.1062	0.1370	4.2428	0.1129
	fastlo2	0.0042	0.0797	0.1067	0.1377	4.3301	0.1135
2.5	quantile	0.0000	0.0802	0.1080	0.1406	2.6932	0.1172
	cyclic	0.0030	0.0795	0.1071	0.1394	2.7350	0.1163
	fastlo	0.0034	0.0797	0.1072	0.1394	2.6653	0.1162
	fastlo2	0.0062	0.0798	0.1075	0.1397	2.7690	0.1166
5.0	quantile	0.0058	0.0774	0.1039	0.1352	3.6135	0.1121
	cyclic	0.0046	0.0773	0.1037	0.1346	3.7939	0.1118
	fastlo	0.0022	0.0774	0.1038	0.1348	3.6643	0.1119
	fastlo2	0.0046	0.0773	0.1038	0.1349	3.6019	0.1120
7.5	quantile	0.0060	0.0845	0.1134	0.1479	3.5267	0.1234
	cyclic	0.0009	0.0827	0.1115	0.1462	3.5527	0.1223
	fastlo	0.0029	0.0835	0.1122	0.1466	3.6194	0.1229
	fastlo2	0.0055	0.0829	0.1118	0.1470	3.7071	0.1233
10.0	quantile	0.0055	0.0744	0.1002	0.1302	2.8603	0.1062
	cyclic	0.0035	0.0743	0.0998	0.1294	3.3200	0.1058
	fastlo	0.0043	0.0743	0.0997	0.1292	3.3048	0.1058
	fastlo2	0.0035	0.0743	0.0998	0.1296	3.3075	0.1061
20.0	quantile	0.0027	0.0822	0.1110	0.1457	3.0393	0.1199
	cyclic	0.0038	0.0819	0.1105	0.1451	3.0374	0.1195
	fastlo	0.0034	0.0820	0.1105	0.1447	3.0400	0.1191
	fastlo2	0.0033	0.0820	0.1106	0.1447	2.9952	0.1191

Table 3: Five number summary plus the mean for the distribution of standard deviations of 201800 normalized PM spot intensity values across the 5 arrays within each concentration level. The four methods compared are quantile, cyclic loess, fastlo, and a second version of fastlo, which normalizes all 30 arrays simultaneously using a group variable (fastlo2).

to perform a single normalization on the entire group. Unfortunately, it is often not clear which normalization approach is the most appropriate.

Both cyclic loess and quantile normalization require the analyst to decide whether the biologic mechanism or microarray experimental process dominates the systematic variation in spot intensity levels among arrays. On the other hand, fastlo can be extended so that the analyst does not have to commit to one or the other of these assumptions. This is a consequence of the model-based nature of fastlo. Recall that the ideal plot for performing normalization for an array would be one with true spot intensity on one axis, say the x-axis, and the observed $-$ true spot intensity on the y-axis. Since we do not know the true spot intensity, we could use a linear model to estimate it. Fastlo uses the simplest linear model, the mean spot intensity value across the arrays. Other linear models could be used instead such as one that also incorporates a grouping variable, which indicates the group to which an array belongs. This allows the data to determine the relative contributions of the biologic mechanisms of interest (represented by a grouping variable) and of the microarray experimental process to the systematic variation of spot intensity levels among the arrays.

Consider the set of GeneLogic liver dilution data (Section 5.3). The grouping variable for this data represents the concentration level of the liver tissue cRNA. It seems reasonable to assume the systematic variability in spot intensity levels among the 30 arrays is dominated by the grouping variable or the concentration level. This suggests the best normalization approach would be to separately normalize the six sets of five arrays separately. This can be done with each of the three normalization techniques: quantile, cyclic loess, and fastlo. Extending fastlo allows us to perform a single normalization for the entire set of 30 arrays. There are three basic models that could be used to accomplish this.

Let k index the treatment group and g the genes; the array index j is nested within k and the spot index i is nested within g (11-20 PM spots per gene for the Affymetrix GeneChip platform). Simple fastlo on all 30 arrays together corresponds to the linear model $y_{ij} = \alpha_i + \epsilon_{ij}$. Conversely, the model $y_{ij} = \alpha_i + \beta_k + (\alpha\beta)_{ik} + \epsilon_{ij}$, containing all treatment and spot interactions, is equivalent to separate fastlo fits to each treatment group. Intermediate models include $y_{ij} = \alpha_i + \beta_k + \epsilon_{ij}$, a simple main effect for treatment or $y_{ij} = \alpha_i + \beta_k + (\alpha\beta)_{gk}$, which has a separate treatment effect per gene.

As proof of principle, we modified the fastlo algorithm to fit this last model and applied it to all 30 arrays in the GeneLogic liver dilution dataset. The modified fastlo algorithm is called fastlo2. Figure 8 compares the normalized $\log_2$ spot intensity values for the set of five arrays with a total cRNA of 10μg pro-

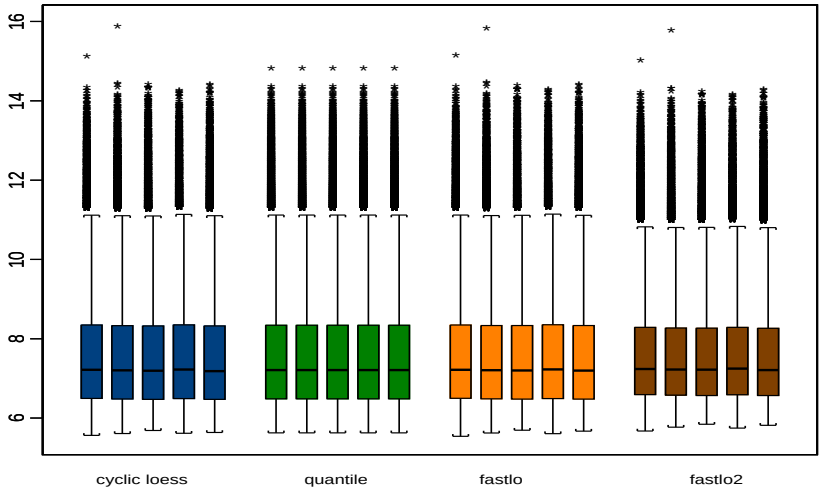

Figure 8: Boxplots of the normalized spot intensity values for the five arrays of the GeneLogic dilution data at total cRNA concentration of 10 μg. The normalizations are from cyclic loess, quantile, fastlo, and fastlo2, which uses a linear model with spot and grouping variables.

duced by four different normalization approaches: cyclic loess, quantile, fastlo, and fastlo2. Cyclic loess, quantile, and fastlo were done separately on each set of five arrays corresponding to the six different concentration levels, and fastlo2 was performed on the entire set of 30 arrays simultaneously. The distribution of the normalized spot intensities produced by the four methods are essentially equivalent. Similar results (not shown) were observed for the five other concentration levels. Table 3 summarizes the distribution of the standard deviations of all the spots for the four methods. The summary values for fastlo2 were remarkably similar to the other three methods. In particular, the results produced by fastlo2 were generally smaller than those of quantile but larger than cyclic. As observed earlier, the differences in these distributions are so small that they are not of practical significance.

7 Discussion

We have accomplished our goal of developing a method, fastlo, that yields normalized values similar to those produced by cyclic loess normalization but is considerably faster. Results from both the simulated data and the real data

indicate that cyclic loess and fastlo produced equivalent normalizations. Fastlo only requires n loess smooths for one pass through a set of n arrays compared to C_2^n loess smooths required by cyclic loess. Also, fastlo requires no more, and often fewer, passes through the data to reach convergence compared to cyclic loess. This results in at least an order of magnitude in computational (and time) savings. In addition, fastlo appears to have the same order of magnitude in computation (and run-time) as quantile normalization; indeed, fastlo was faster than quantile normalization for a relatively small set ($n = 5$ arrays) but quantile normalization becomes faster for the set of ten arrays. The relative speed depends on the implementation details of fastlo and quantile normalization.

All three methods appear to yield similar normalized values. It is worth noticing that the datasets used above were ideal for quantile normalization; the true spot intensities (and so their distributions) were identical across the arrays making this perfect for a method that forces the spot intensity distributions of the arrays to be identical. In situations where spot intensities do differ among tissue initially thought to be similar, e.g. a molecular subtype within brain cancer tissue, quantile normalization may be too aggressive and attenuate differences of interest. These issues are yet to be resolved.

An advantage of fastlo not shared by the other methods is that it is model-based. As demonstrated above, this allows us to normalize an entire set of arrays to be compared as a group—i.e. no need to perform separate normalizations for subgroups. This is useful in situations where it is not clear to what degree the biological variation between the groups and the microarray experimental process contribute to the systematic variation among the arrays. Using a grouping variable as well as spot variables permits the normalization of an entire set of arrays simultaneously allowing the systematic variation to be shared between biologic variation among groups and the microarray experimental process. This model can be easily extended to incorporate other variables that are part of the underlying experimental design such as subject demographic variables, time, dose level, etc. Other estimation options are also open, e.g. shrinkage of the treatment coefficients toward zero in an experiment where it was felt that only a small percentage of genes are differentially expressed.

Finally, a more robust estimator could be used to generate $\hat{y}$. If the number of arrays to be normalized is relatively small, the mean estimate can be significantly influenced by a single outlier. We explored using the median of spot i across all arrays to generate the reference array. The results were not promising; they resulted in a considerably biased estimate. Recall the data to

be normalized is a matrix of arrays as columns and spot intensities as rows. Fastlo iterates between fits to the rows ($\hat{y}$) and loess smooths on the columns of the array. It appears that the same type of estimator must be used for row and column operations in the algorithm. We have not pursued this further.

Overall, fastlo (1) produces normalized results similar to cyclic loess (as well as quantile) normalization, (2) is considerably faster than cyclic loess, and (3) has added versatility above quantile and cyclic loess normalization because it is model-based.

References

[1] Affymetrix. *Microarray Suite User Guide, Version 5.* Affymetrix, Inc., 2001. http://www.affymetrix.com/support/technical/manuals.affx.

[2] R. Irizarry, B. Hobbs, F. Collins, Y. Beazer-Barclay, K. Anntonellis, U. Scherf, and T. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[3] C. Li and W. Wong. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc. Natl Acad. Sci.*, 98:31–36, 2001.

[4] T-M. Chu, B. Weir, and R. Wolfinger. A systematic statistical linear modeling approach to oligonucleotide array experiments. *Mathematical Biosciences*, 176:35–51, 2002.

[5] A. Hartemink, D. Gifford, T. Jaakola, and R. Young. Maximum likelihood estimation of optimal scaling factors for expression array normalization. In *SPIE BIOS 2001*, 2001.

[6] T. Richmond and S. Somerville. Chasing the dream: plant est microarrays. *Curr Opin Plant Biol*, 3(2):108–116, 2000.

[7] Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. Speed. Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, 30(4):e15, 2002.

[8] O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression processing and modeling. *Proceedings of the National Academy of Sciences*, 97(18):10101–10106, 2000.

[9] M. A. Newton, C. M. Kedziorski, C. S. Richmond, F. R. Blattner, and K. W. Tsui. On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8(1):37–52, 2001.

[10] B. Bolstad, R. Irizarray, M. Astrand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*, 19(2):185–193, 2003.

[11] L. M. Cope, R. A. Irizarry, H. Jaffee, Z. Wu, and T. P. Speed. A benchmark for affymetrix genechip expression measures. *Bioinformatics*, 1(1):1–13, 2003.

[12] S. Dudoit, Y. H. Yang, M. J. Callow, and T. P. Speed. Statistical methods for identifying differentially expressed genes in replicated cdna microarray experiments. *Statistica Sinca*, 12:111–139, 2002.