

Normalizing Oligonucleotide Arrays*

Magnus Åstrand
Clinical Science
AstraZeneca R & D Mölndal
S-431 83 Mölndal
Sweden

November 15, 2001

1 Introduction

When analysing data from oligonucleotide arrays normalizing is a necessity to allow direct array-to-array comparisons. In this report a method for normalizing oligonucleotide arrays using smooth curves is proposed. It is a method aimed for normalizing the raw feature intensities, i.e. the PM and MM intensities. But the method can just as well be applied to PM-MM or an expression index obtained from the raw feature intensities. When normalizing two arrays, A and B, using a smooth normalizing curve there are two main approaches. Perhaps the most obvious one is to fit a curve with the intensities of array A (or B) on the y-axis and the intensities of array B (or A) on the x-axis. The intensities of array A (or B) are then normalized by simple subtracting the difference between the fitted curve and the line through the origin of coordinates with slope one. Another solution is to fit a smooth curve with the difference A-B on the y-axis and the sum A+B on the x-axis. The differences, A-B, are then normalized by subtracting the fitted curve. This is the basis of the method proposed in this report.

2 In detail

Suppose we have k arrays. Let the raw feature intensities, i.e. all PM and MM intensities, be denoted by the $n \times k$ matrix Y_0 . These intensities are logged and transformed using an orthonormal matrix M yielding

$$Y_1 = [x, y_1, \dots, y_{k-1}] = \log(Y_0) \cdot M^t$$

The first row of M is always the 1-vector times $\sqrt{1/k}$. The other rows are a set of orthonormal contrast. Note that when k equals 2 M is unique, but this is not the case for $k > 2$. When k equals 2, we use $M = M_2$ and when k equals 4 we can use $M = M_4$:

$$M_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \sqrt{\frac{1}{2}} \quad M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \frac{1}{2}$$

This use of an orthonormal matrix is just a change of basis, where the rows of M form the new basis, denoted the alternative basis from now on. When k equals 2 we see that, besides a constant, the alternative basis corresponds to what is called a Bland & Altman plot, i.e. a plot of the difference vs the mean. The change to logarithmic scale before performing the change of basis is used to make the error variances more homogenous.

*In the making

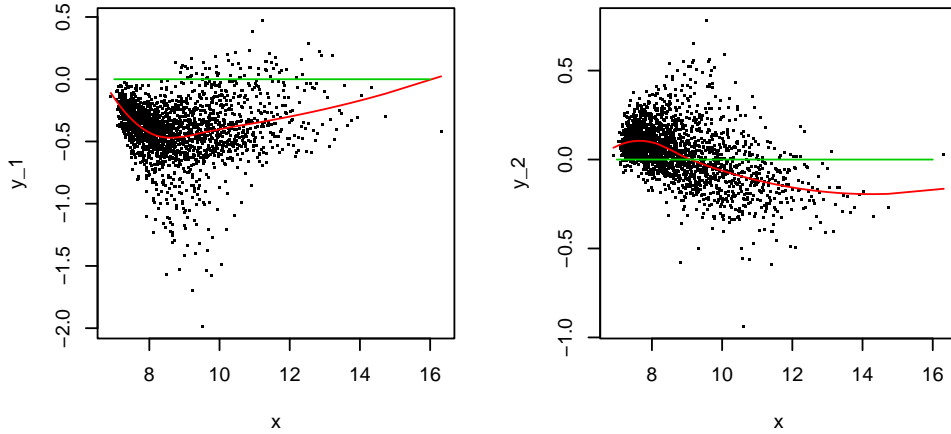


Figure 1: Plots of the 2 contrasts for 3 arrays, A, B and C, prior to normalizing. The red curve is the fitted normalizing curve and the green line is the reference line.

2.1 Fitting the normalizing curve

Using the alternative basis we fit the normalizing curve. Thus, we use the first column of the transformed intensities in Y_1 , i.e. x , as a predictor for column $2, \dots, k$ of Y_1 , i.e. y_1, \dots, y_{k-1} . When doing this it's important to have in mind that the set of orthonormal contrasts is not unique. Thus, the method for fitting the curve should be invariant with respect to choice of contrast. In order to achieve this we fit a smooth curve using a local regression model (loess) to each vector y_i . For the curve to be less sensitive for outliers we use re-descending M estimator with the biweight function as done in the R-function loess (Chambers 1997), but with one important modification. If $\hat{y}_1, \dots, \hat{y}_{k-1}$ are the vectors of the fitted values we take $\hat{\epsilon}$ as the euclidian distance between the rows of $[y_1, \dots, y_{k-1}]$ and $[\hat{y}_1, \dots, \hat{y}_{k-1}]$.

$$\hat{\epsilon} = \sqrt{\sum_{i=1}^{k-1} (\hat{y}_i - y_i)^2}$$

Thus in each iteration the same set of robust weights is used for each of the $k - 1$ contrast vectors, and these weights are invariant to the choice of orthonormal contrasts. Further, since the local regression model is fitted using weighted least squares the fitted curve is invariant to the choice of orthonormal contrasts.

2.2 Normalizing the arrays

Now let $\hat{y}_1, \dots, \hat{y}_{k-1}$ be the fitted values after the last iteration. The normalizing curve can be represented with the matrix $[x, \hat{y}_1, \dots, \hat{y}_{k-1}]$. These points can be viewed either using the alternative basis or the original basis, red curve in figure 1 and left graphs in figure 3 respectively. Hence, we still could choose a baseline array and normalize the others by the fitted normalizing curve, e.g. using the two upper left graphs in figure 3 and normalize B and C to the baseline array A. If doing so, we have used a normalizing curve that is invariant to the choice of baseline array. But the scale that we normalize to still depend of which baseline array we choose.

Another way of normalizing the arrays using the fitted curve is to simply subtract the fitted values using the alternative basis, i.e. figure 1, and then go back to the original basis using the matrix M . In this case the normalized intensities would be

$$\exp\{[x, y_1 - \hat{y}_1, \dots, y_{k-1} - \hat{y}_{k-1}] \cdot M\}$$

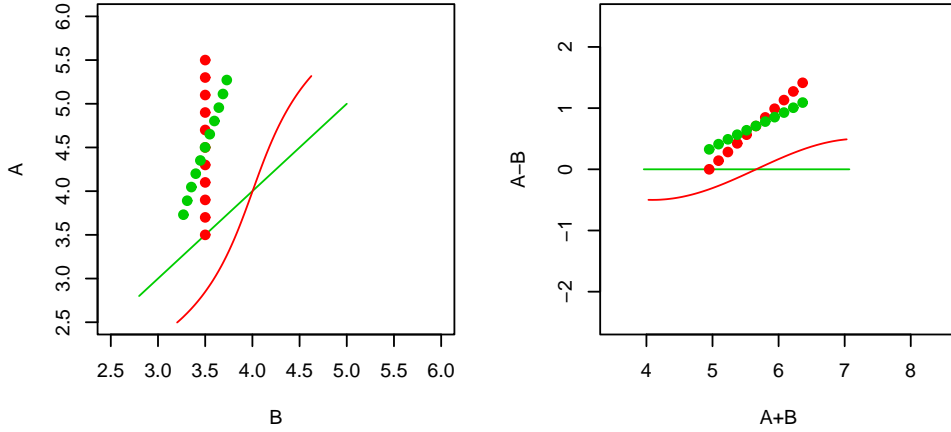


Figure 2: Normalizing two arrays prior change to original basis. Red line is the normalizing curve, green line the reference line. Red and green dots are the intensities before and after normalization respectively.

But this results in a non-smooth normalizing procedure, in the sense that intensities being equal on one array prior normalizing may not be equal after. Figure 2 shows way this is the case. The graphs show a couple of features which all have intensities equal to 3.5 on array B, and intensities ranging from 3.5 to 5.5 on array A prior to normalizing (red dots). The intensities are normalized using the alternative basis in the right graph yielding the normalized intensities shown as green dots. Now the features have intensities ranging from 3.25 to 3.75 on array B.

However, we can still use the normalizing curve with the alternative basis. If $[x, \hat{y}_1, \dots, \hat{y}_{k-1}]$ is a representation of the normalizing curve, $[x, 0, \dots, 0]$ is a representation of what the curve should be after normalization. Hence, the mapping

$$[x, \hat{y}_1, \dots, \hat{y}_{k-1}] \mapsto [x, 0, \dots, 0]$$

defines a transformation that does the job of even out the contrast for the alternative basis. Moreover, the mapping

$$\exp \{ [x, \hat{y}_1, \dots, \hat{y}_{k-1}] \cdot M \} \mapsto \exp \{ [x, 0, \dots, 0] \cdot M \}$$

defines the same transformation but for the original basis and anti-logged scale. This transformation forms a function $F : R^k \mapsto R^k$ that row by row normalizes the matrix of intensities Y_0 . Thus if $F((x_1, \dots, x_k)) = (f_1(x_1), \dots, f_k(x_k))$ f_j is function that normalizes array j . These functions, f_1 , f_2 and f_3 for the set of 3 arrays A, B and C, are shown in the right graphs of figure 3.

Since $x \cdot M = x / \sqrt{k}$ equals $\overline{\log(Y_0)}$, i.e. the mean across the rows of $\log(Y_0)$, this way corresponds to normalizing to a scale determined by $\overline{\log(Y_0)}$, i.e. the geometrical mean of the arrays.

2.3 Adding arrays

Suppose a set of arrays have been normalized and further analyzed, e.g. expression indexes have been computed. Now we have an additional set of arrays that we would like to add to the original ones to use in the same analysis. We would like to do this without affecting the intensities of the original set. This can be done by first normalizing the new set of arrays separately. These arrays are then normalized to a scale determined by their geometrical mean. Thus we have to transform these to the same scale as the original set, i.e. the scale determined by the geometrical mean of the arrays in the original set.

Let Y_{01} and Y_{02} be the normalized intensities of the original and new set of arrays respectively. Also let x_1 and x_2 be the mean across the rows of $\log(Y_{01})$ and $\log(Y_{02})$ respectively. To find a

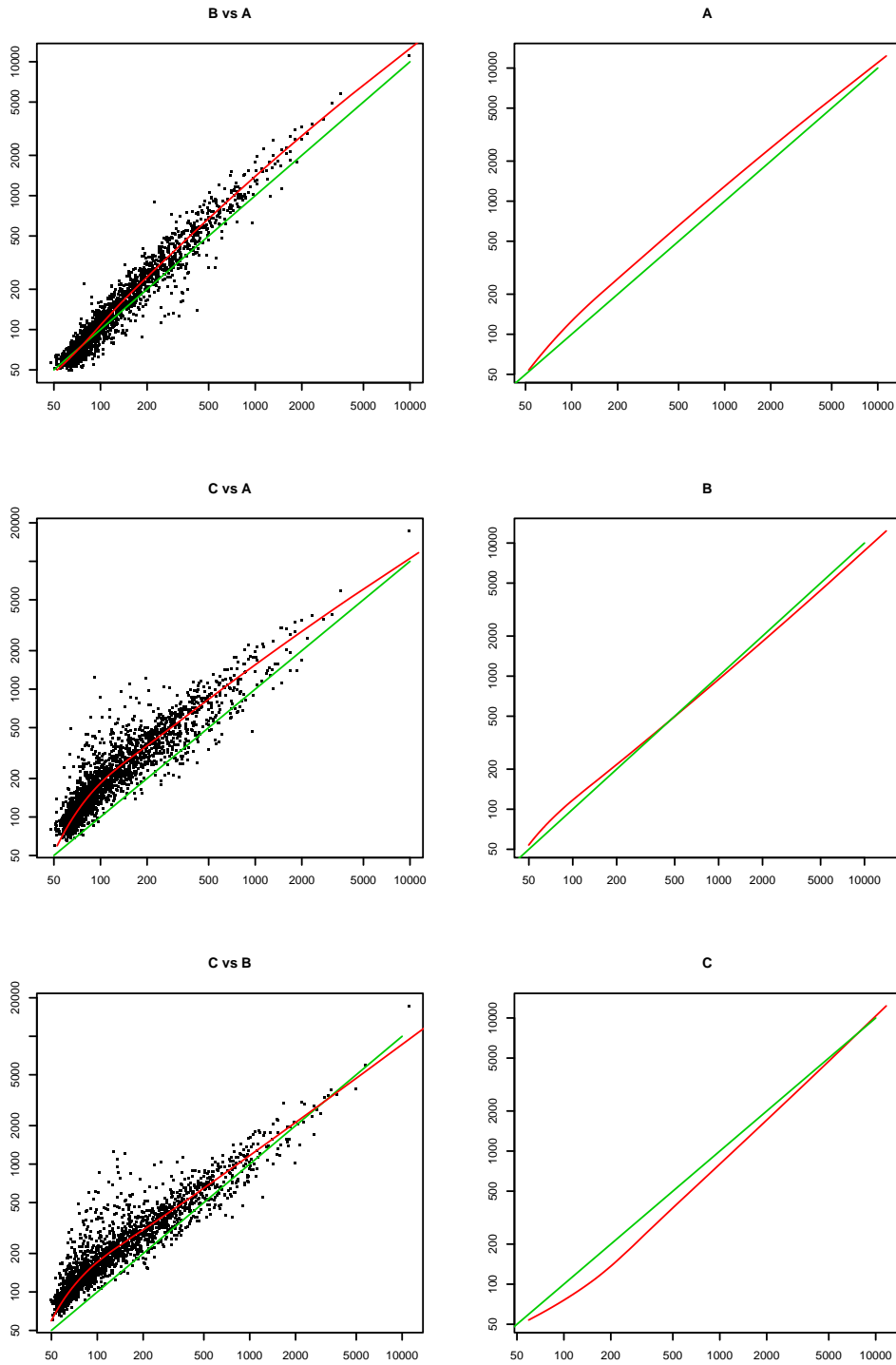


Figure 3: Left graphs: Scatter plots of 3 arrays, A, B and C, prior to normalizing. The red curve is the fitted normalizing curve and the green line is the reference line. Right graphs: Red lines are the normalizing functions for each array defined through the mapping of the fitted normalizing curve on to the reference line in figure 1.

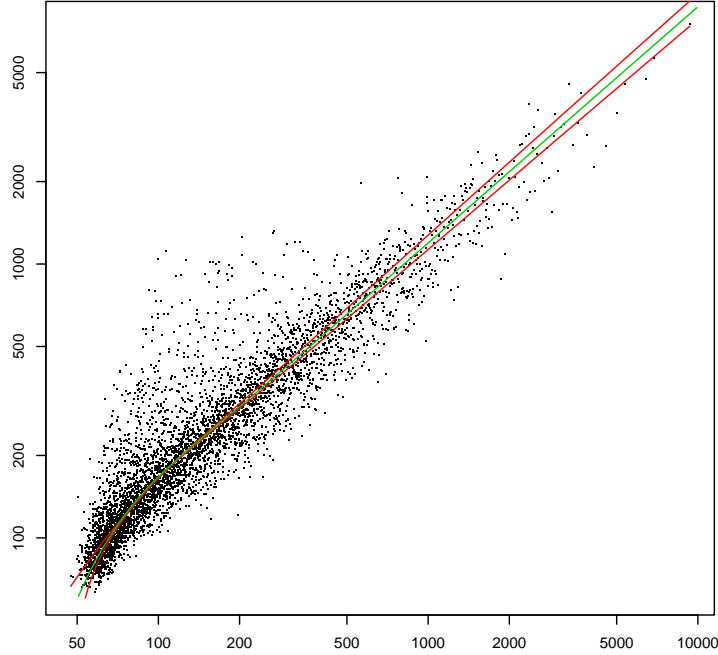


Figure 4: Scatter plot of 2 arrays, A and B. The two red lines are fitted loess curves from using A (and B) as a predictor for B (and A). The green curve is a loess curve fitted using the alternative basis.

transformation that transforms the new arrays the same scale as the original arrays we apply the normalizing method treating x_1 and x_2 as log-intensities of two “arrays”. If \hat{y}_1 is the fitted values for the contrast of these “arrays”, we have the mappings:

$$\begin{aligned} \exp \left\{ \frac{x_1 + x_2}{2} + \frac{\hat{y}_1}{\sqrt{2}} \right\} &\mapsto \exp \left\{ \frac{x_1 + x_2}{2} \right\} \\ \exp \left\{ \frac{x_1 + x_2}{2} - \frac{\hat{y}_1}{\sqrt{2}} \right\} &\mapsto \exp \left\{ \frac{x_1 + x_2}{2} \right\} \end{aligned}$$

that forms the functions f_1 and f_2 that would normalize the two “arrays” to a common scale. However, we only want to change the scale of second one (x_2). To do this we apply $f_1^{-1} \circ f_2$ formed by the mapping

$$\exp \left\{ \frac{x_1 + x_2}{2} - \frac{\hat{y}_1}{\sqrt{2}} \right\} \mapsto \exp \left\{ \frac{x_1 + x_2}{2} + \frac{\hat{y}_1}{\sqrt{2}} \right\}$$

on the intensities of the second “arrays”. Hence the $f_1^{-1} \circ f_2$ is the function that transforms the intensities of the new set of arrays to the scale of the original set.

3 Discussion

As mentioned the most obvious way to normalize two arrays using a smooth curve is probably to fit a curve using the original basis. But it has one obvious drawback; we have to choose the baseline array, i.e. the array to place on the x-axis. How important this drawback is for the result of the down stream analysis of the raw intensities, i.e. computing expression indexes, is hard to tell. A quick test using the software DNA-Chip Analyzer (Li and Wong 2001a) on two arrays, A

and B, implies that it's not negligible. For each choice of baseline array, the ratios of expression indexes, array A to array B, was computed. Of the 8799 probes the ratio differed more than 10% for 1603 probes (18%), when using array A as baseline instead of array B. The difference was most notable among probes with small indexes. But even among the 2500 probes with highest expression indexes, the ratio differed more than 10% for 10% of the probes. The two sets of probes filtered out based on the confidence interval and absolute difference for each choice of baseline array differed. There were 469 and 298 probes in the two sets of which 265 was contained in both sets.

Moreover, figure 4 show a scatter plot of the raw intensities of the same two arrays together with three curves. The two red curves are the loess curves fitted using the original basis with A as the predictor of B and vice verse. The green curve is the loess curve but fitted using the alternative basis. There is a notable difference between the two red curves with the green curve lying in between. Again there is a clear indication that the choice of baseline array does matter.

On the other hand, this approach is simple to apply to a set of k arrays: simply choose a baseline array and normalize the other to that array. Also, if an analysis have been done on a set of arrays, it's easy to add arrays without affecting the analysis of original set. Just use the baseline array of the original arrays to normalize the new arrays.

When using the approach suggested in this report there is no choice of baseline array, instead all arrays are treated uniformly. It's not as straight forward to add extra arrays without affecting the result of the analysis of the original set. But the solution in 2.3 does the job.

4 Software

We have implemented a R-package (Åstrand 2001) containing functions that performs the normalizing method proposed in this report. As mentioned the orthonormal matrix M is not unique. The matrix used in the package is

$$M = \begin{bmatrix} a & a & a & \cdots & a & a \\ a & b & c & \cdots & c & c \\ a & c & b & \cdots & c & c \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a & c & c & \cdots & b & c \\ a & c & c & \cdots & c & b \end{bmatrix} \quad \begin{aligned} a &= \frac{1}{\sqrt{k}} \\ b &= \frac{k-2}{k-1} - \frac{1}{\sqrt{k(k-1)}} \\ c &= -\frac{1}{k-1} - \frac{1}{\sqrt{k(k-1)}} \end{aligned}$$

The evaluation of loess curves for large amount of data takes long time. By choosing a subset of features to be used for fitting the normalizing function the run-time can be reduced considerable. The R-package also contains a function for selecting a subset based on ranks. It works in a similar way as the subset selection described by Li and Wong (2001b). First the intensities are ranked separately for each array. Then the range of the ranks over the arrays is computed for each feature. We then select features with small relative rank-range. This is done iteratively, until a subset of desired size is obtained.

References

- Åstrand, M. (2001). R-package maffy [<http://www.math.chalmers.se/magnusaa/maffy.html>].
- Chambers, John M. Hastie, T. J. (1997). *Statistical models in S*. Chapman & Hall.
- Li, C. and W. H. Wong (2001a). DNA-Chip Analyzer. [<http://www.dchip.org>].
- Li, C. and W. H. Wong (2001b). Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application. *Genome Biology* 2(8), research0032.1-0032.11.