BMC Bioinformatics

Methodology article

Methodological study of affine transformations of gene expression data with proposed robust non-parametric multi-dimensional normalization method

Henrik Bengtsson*[1] and Ola Hössjer[2]

Address: [1]Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Box 118, SE-221 00 Lund, Sweden and [2]Mathematical Statistics at the Mathematics Department, Stockholm University, SE-106 91 Stockholm, Sweden

Email: Henrik Bengtsson* - hb@maths.lth.se; Ola Hössjer - ola@math.su.se

* Corresponding author

Published: 01 March 2006

BMC Bioinformatics 2006, **7**:100 doi:10.1186/1471-2105-7-100

Received: 18 May 2005
Accepted: 01 March 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/100

Abstract

Background: Low-level processing and normalization of microarray data are most important steps in microarray analysis, which have profound impact on downstream analysis. Multiple methods have been suggested to date, but it is not clear which is the best. It is therefore important to further study the different normalization methods in detail and the nature of microarray data in general.

Results: A methodological study of affine models for gene expression data is carried out. Focus is on two-channel comparative studies, but the findings generalize also to single- and multi-channel data. The discussion applies to spotted as well as in-situ synthesized microarray data. Existing normalization methods such as curve-fit ("lowess") normalization, parallel and perpendicular translation normalization, and quantile normalization, but also dye-swap normalization are revisited in the light of the affine model and their strengths and weaknesses are investigated in this context. As a direct result from this study, we propose a robust non-parametric multi-dimensional affine normalization method, which can be applied to any number of microarrays with any number of channels either individually or all at once. A high-quality cDNA microarray data set with spike-in controls is used to demonstrate the power of the affine model and the proposed normalization method.

Conclusion: We find that an affine model can explain non-linear intensity-dependent systematic effects in observed log-ratios. Affine normalization removes such artifacts for non-differentially expressed genes and assures that symmetry between negative and positive log-ratios is obtained, which is fundamental when identifying differentially expressed genes. In addition, affine normalization makes the empirical distributions in different channels more equal, which is the purpose of quantile normalization, and may also explain why dye-swap normalization works or fails. All methods are made available in the aroma package, which is a platform-independent package for R.

Background

The objective of most gene-expression measurements is to assess the *expression levels* of (all or a subset of) genes in one or several cell populations. Typically, mRNA abundances are measured, although techniques for measuring protein-levels also exist. The *microarray technique* [1] provides a way to measure mRNA transcripts for a large number of genes simultaneously, typically in the order of 10^3 – 10^5 or more. Microarrays have well defined immobilized regions, which each consists of clones or synthesized sequences of DNA specific to a unique gene. We refer to these (non-hybridized) regions or spots as *probes* [2]. A cocktail of cDNA created from the RNA extract from the cell population in study is then, for a few hours, *hybridized* to the DNA on the microarray after which excess cDNA is washed off. The result is that each region of the microarray contains a certain amount of hybridized DNA unique to the corresponding gene. By first labeling the cDNA strands in the sample cocktail with a radioactive or a fluorescent probe, the amount of hybridized DNA can be measured utilizing radioactive sensitive film or a color-sensitive scanner, respectively.

By measuring the gene expression for a specific gene, we try to assess how active that gene is (measured on some scale). Because it is hard to identify an *absolute* scale to measure on, often, but also for various other reasons, a reference is used to obtain a *relative* scale. As even genes from the same sample are not directly comparable to each other, each gene gets its own reference, which is typically the same gene from a reference sample. With this approach, we can obtain *gene-expression ratios* for every gene, which for instance can be used to test the hypothesis if a gene (in the test sample) is *differentially expressed* or not (compared to the gene in the reference sample). This is the core idea behind the two-channel microarray technology, in which the test and the reference cDNA cocktails are hybridized simultaneously and in a competitive way to the same array. The same idea has been adopted by single-channel hybridization technologies where the comparison instead is done numerically in the data analysis step. Even if gene-by-gene references are used, the measurements are not perfect and they are likely to contain systematic errors, which possibly vary from measurement to measurement, and the obtained gene-expression ratios may still be biased and not comparable to each other. What we ultimately would like to do is to measure all control and all reference samples under identical conditions. The aforementioned two-color microarray technology tries, in some sense, to do this by measuring the control/reference pairs for each gene in one hybridization (although it is not clear if the gain from co-hybridizing two samples with different labels is larger than hybridizing twice with identical labels and then scanning the samples separately).

In this paper, we present an affine model that explains many of the systematic effects frequently observed when gene-expression levels from two (or more) samples are compared. The main contributors to such systematic effects are offsets in the individual channel signals, which give non-linear systematic effects in ratios. We will not provide an error model, but only a deterministic model. The main reason for this is that an error-free model makes it easier to understand the impact that channel offsets have on the downstream analysis regardless of gene-expression technology used. This is especially of interest as these are often implicitly assumed to be small and of no effect, which we believe is a too strong assumption. The impact of channel offsets is much larger that the noise, which is why we allow us to assume zero noise in the discussion. Although some error models have been suggested for microarray data [3], we believe research beyond this article is required before we can understand and correctly model the various error sources introduced in the microarray process.

The outline of this paper is as follows. In the Model section, a general model that incorporates all steps of any gene-expression technology is given. By dissecting the generic model and focusing more on the microarray technologies, an affine model is introduced. Here is also the widely adopted and accepted log-ratio log-intensity transform under affine transformations formalized. The Results section consists of three main parts. In the first, we show how the affine transform introduces intensity and fold-change dependent biases in the log-ratios. In the second part, we revisit common normalization methods, to which dye-swap and background correction may also be counted, and discuss them using the affine model. In the third and concluding part, we suggest a novel and multipurpose robust normalization method to back-transform data to the linear (proportional) space. We end the paper with a Discussion section where we give similarities to other normalization methods followed by a Conclusions section. Details on calculations and the data set used are given in the Methods appendix.

Results

General model

Consider an experiment with genes $i = 1,..., I$ from RNA extracts $c = 1,..., C$. For example, in oligonucleotide microarrays each slide measures the gene-expression levels of exactly one RNA extract whereas for a two-color microarrays each slide measures two RNA extracts, one in each channel. From now on, we refer to the RNA extracts or replicates of such as *channels*. Let $x_{c,i}$ be the true gene-expression level of gene i in channel c and let $y_{c,\ i}$ be the corresponding observed gene-expression level. The relationship between the observed and the true expression levels can be written as

$$y_{c,i} = f_c(x_{c,i}) + \varepsilon_{c,i} \quad (1)$$

where f_c is a channel specific *measurement function*, which includes all steps in the gene-expression acquisition process. Most generally, we have that $E[\varepsilon_{c,i}] = 0$ and $V[\varepsilon_{c,i}] = \sigma^2_{c,i}$, where the variance can take any form. Importantly, the properties of ε_c are not well understood and depends on platform used, but also which part of the process that is studied. For this reason and because of the many interesting effects that the affine transformation (presented below) generates by itself, we conduct this study under the assumption of noise-free data. Relationship (1) may be specified for subsets of genes or probes, e.g. print tip [4], microtiter plate or clone library [5] groups. Spatial dependencies may also be modeled. However, to simplify the discussion that follows, we avoid such details.

Since inference is ideally based on $x_{c,i}$, the inverse of f_c has to be identified, something that, in theory, is possible if it is strictly increasing. Violation of this constraint has been observed in, for instance, two-color microarray data. This can be due to too high concentrations of fluorophores, which sometimes quenches the signal so much that the signal decreases when the concentration increases [6,7]. Extreme saturation in the scanner, which is commonly observed when the PMT gain is set too high, results in censored signals, which in turn prevents a unique inverse of the measurement function to be found. This paper does not discuss saturation further, because we believe that saturation can and should be avoided.

Dissection of the overall measurement function

Formally, each step in the microarray process can be seen as a function that takes a set of input objects and outputs another set of objects. The sequential nature of the process makes it possible to think of the measurement function f_c as a *composite function* (function of functions); $f_c = f_{c,S} \circ f_{c,S-1} \circ \cup \circ f_{c,1}$, where S is the number of steps in the process. For instance, and of course simplified, it could be that $f_{c,1}$ models the extraction of the RNA from the cell, $f_{c,2}$ models the reverse transcription of RNA into cDNA and so on. Some of these *submeasurement functions* are shared by several channels and others are channel specific or even gene specific. Moreover, there may be joining subfunctions too, e.g. the hybridization of labeled cDNA sequences to the probes on the array. In this paper, measurement functions of different channels are treated independently.

A first-order Taylor series expansion of an arbitrary measurement function $f_c(x_{c,i})$, has the form

$$f_c(x_{c,i}) = a_c + b_c x_{c,i} + R_c(x_{c,i}), \forall c,i. \quad (2)$$

From the above dissection of a measurement functions, it is easy to argue that some of the subfunctions may introduce offset (bias) and that there for this reason ought to be an offset in f_c (we will use the terms bias and offset interchangeably). For instance, the offset terms may be due to non-uniformity of the reverse transcription, the labeling [7] or the hybridization, due to dark noise in the PMT [8] or laser scatter light in the scanner, background noise, non-uniformity of the scanned glass slide [9], or threshold effects etc. In [10] it is shown how various background estimates based on different image analysis methods may introduce bias. Similarly, we have shown that different scanners may introduce bias [11].

The affine measurement function

In order to focus on the effects of a_c and b_c, but also because it results in the simplest parametric measurement function possible, we assume $R_c(x_{c,i})$ in (2) to be small. The *affine measurement function* is

$$f_c(x_{c,i}) = a_c + b_c x_{c,i}, \forall c,i, \quad (3)$$

with unique inverse

$$x_{c,i} = f_c^{-1}(y_{c,i}) = \frac{y_{c,i} - a_c}{b_c}, \forall c,i, \quad (4)$$

where a_c is the overall offset (bias) and $b_c > 0$ is the overall scale factor in channel c. The a_c parameters are commonly positive, but under certain circumstances, for instance, as demonstrated later, when two different measuring techniques are compared, the effective offset may be negative. Modeling microarray data by an affine transform is not novel [3,12-14], but the reasons for it might have been different in those papers.

The log-ratio log-intensity transform

In two-color but also in oligonucleotide microarray experiments, it is convenient to do statistical analysis on the log-ratios and the log-intensities [15] of the gene-expression levels in two channels instead of on the expression levels directly. For gene i we have that

$$M_i = \log_2 \frac{y_{R,i}}{y_{G,i}} = \log_2 \frac{f_R(x_{R,i})}{f_G(x_{G,i})} \quad (5)$$

$$\begin{aligned} A_i &= \frac{1}{2}\log_2(y_{R,i} \cdot y_{G,i}) \\ &= \frac{1}{2}\log_2(f_R(x_{R,i}) \cdot f_G(x_{G,i})). \end{aligned} \quad (6)$$

For simplicity, we denoted channels 1 and 2 by R and G, which are mnemonics for the red and the green dyes commonly used in two-color microarray data. A rationale for

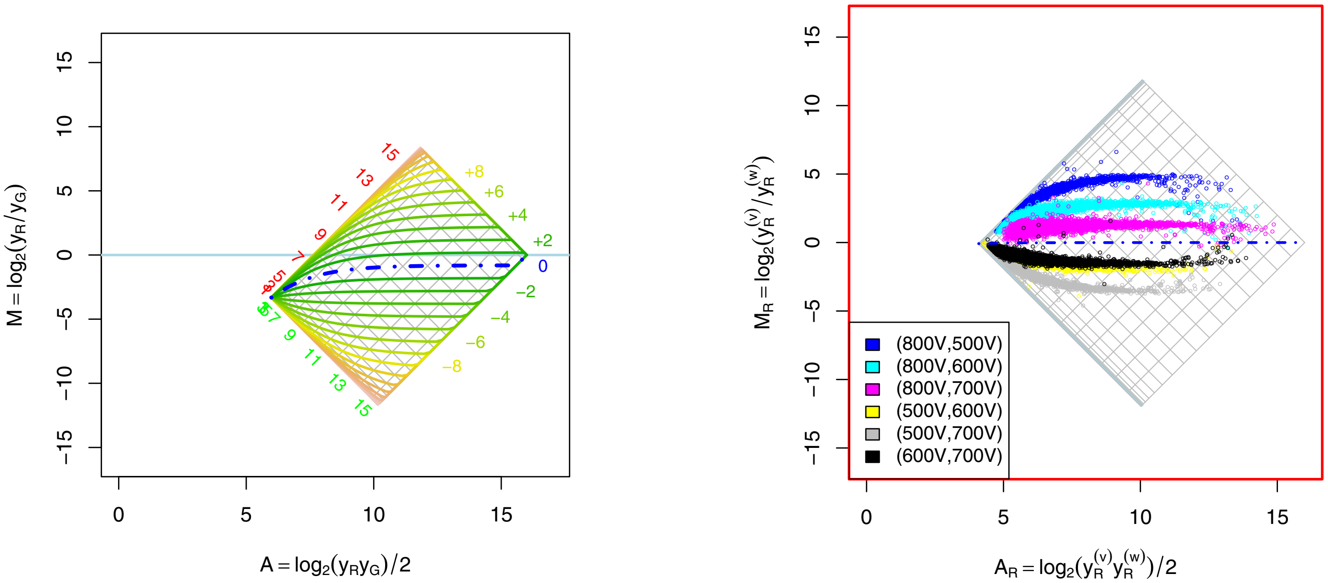

Figure 1

Affine transformation of the red and the green signals. *Left*: Affine transformation of the red and the green signals for $\mathcal{A}_1$ = {(a_G, a_R) = (200, 20), (b_G, b_R) = (1.4, 0.8)}. The observed log-ratios as a function of the observed log-intensities for different fold changes. The blue dot-dash curve corresponds to the non-differentially expressed genes and the thinner curves above and below this curve represent $\log_2 r$ = ± 1, ± 2,... as labeled to the right of the curves. The lines in the gray grid, which is rotated 45 degrees (in (2*A*, *M*)), show the levels where the *true* signals $\log_2 x_R$ and $\log_2 x_G$ are equal to ..., -1, 0, 1,..., 16. These levels have been labeled to the left of the grid. No observations can lie outside this grid. *Right*: Real-world example of an affine transformation. The same slide was scanned four times at four different PMT settings. For each of the six scan pairs, the *within-channel* log-ratio and log-intensities were calculated. Data shown is from the red channel, which was estimated to have an offset of a_R = 20.3 for all scans.

this bijective transform (if the observed signals are positive) is that the main measure of interest, the fold change, is contained in one variable. However, since the transform is based on observed expression levels and not the true ones, *M* alone does indeed not carry all information about the biological fold change. This can be seen if the *true fold change* for an arbitrary gene *i* is considered;

$$r_i = x_{R,i}/x_{G,i} \qquad (7)$$

where $r_i > 0$. Dropping gene index *i* in (5) and (6), *M* and *A* can be written as functions of x_G and *r*, i.e. $M = g_r(x_G)$ and $A = h_r(x_G)$. Thus,

$$M = m_r(A) = g_r(h_r^{-1}(A)), \qquad (8)$$

which shows that *M* is a function of *A* (and *r*). Hence, and discussed thoroughly below, commonly observed intensity-dependent effects in the log-ratios may contain valuable information, and consequently, applying normalization methods without care may result in loss of information and introduced bias.

Log-ratios as a function of log-intensities with affine transformations

Under an affine transformation, the relationship between the observed log-ratios and the observed log-intensities for a fixed fold change *r*, omitting gene index *i*, is

$$M = m_r(A) = \log_2 r + \log_2 \beta + \log_2 \frac{\frac{1}{2}\alpha(r) + \sqrt{\frac{1}{4}[\alpha(r)]^2 + r\beta 2^{2A}}}{-\frac{1}{2}\alpha(r) + \sqrt{\frac{1}{4}[\alpha(r)]^2 + r\beta 2^{2A}}} \qquad (9)$$

where $\alpha(r) = a_R - r\beta a_G$ quantifies how much *M* depends on *A* at the given fold change, and $\beta = b_R/b_G$ is the *relative scale factor* between the two channels compared. See Methods for details. Recall that $\log_2 r$ is the variable of interest. The derivative of *M* with respect to *A* for a fixed fold change *r* is

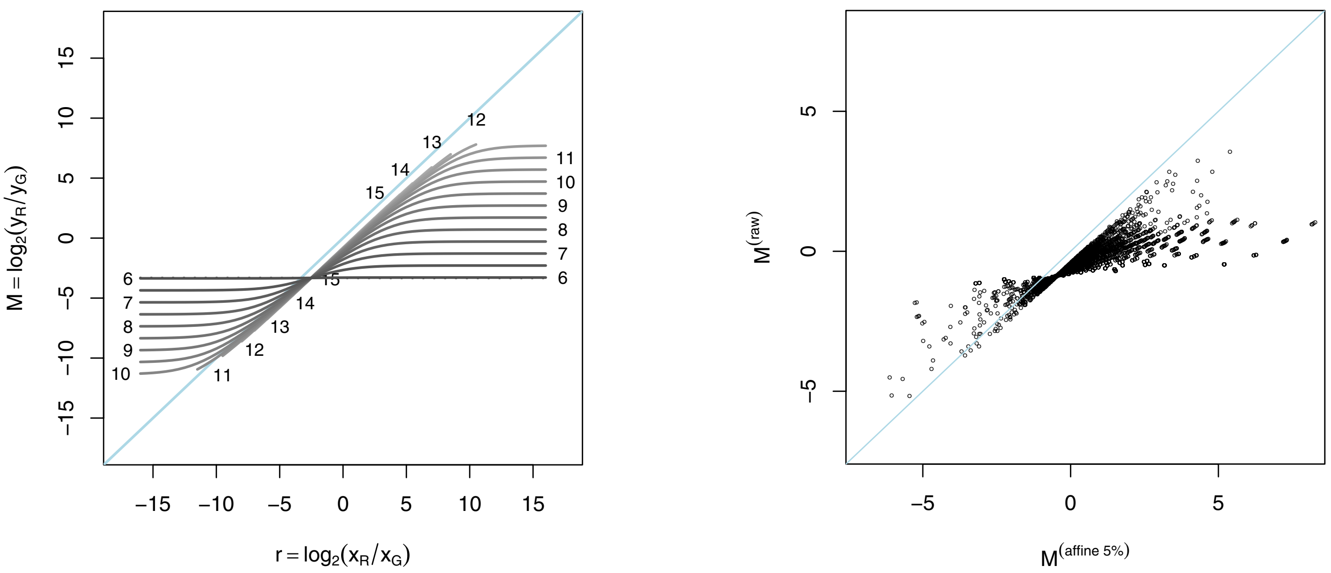

Figure 2

Bias in the log-ratios introduced by the affine transform. *Left*: Bias in the log-ratios introduced by the affine transform $\mathcal{A}_1$. Each line displays the relationship between the observed and the true log-ratios at a certain (observed) log-intensity *A*. Each curve is marked with the value of *A*. We have chosen to truncate the curves when the signals become saturated and the labels for those curves are positioned approximately where they have been truncated. For low intensities there is a great bias (deviance from the diagonal line), especially for large fold changes. At higher intensities the bias is smaller. The curves intersect at the one fold-change level that is independent of the intensity. *Right*: Real-world example of log-ratios for non-normalized versus affine normalized (with 5% negative) signals. The affine parameters are ($\hat{a}_G$, $\hat{a}_R$, $\log_2 \hat{\beta}$) = (45.7, 27.0, -0.418). To clarify the intensity-dependent effect only data points close to *A* = 0.0, 0.5,...,16 are shown.

$$\left.\frac{dM}{dA}\right|_{x_R=rx_G}(A) = -\frac{\alpha(r)}{\sqrt{\frac{1}{4}[\alpha(r)]^2 + r\beta 2^{2A}}}. \qquad (10)$$

Consider a fixed r and define $\alpha = \alpha(r)$. Then there are only two parameters in (9) and (10) that determine the shape of $m_r(A)$, namely α and β. Consequently, when a_R, $a_G \neq 0$, M is independent of A if and only if $\alpha = 0$, that is, when $r = (b_G a_R)/(b_R a_G)$. For this particular value of r, we have that the observed log-ratio is $M = \log_2 (a_R/a_G)$, which is independent of scale factors. Moreover, for log-ratios of non-differential expressions, that is $log_2 r = 0$, to be independent of A, it must be true that $b_G a_R = b_R a_G$ or, equivalently, $b_R/b_G = a_R/a_G$. It is also clear from (10) that the scale parameters cannot introduce any curvature themselves, but only enhance or decrease curvature introduced by the offset. In addition to this, relative scale different from one shifts the log-ratios up or down. Moreover, the size of the effect that the offset terms have on the log-ratios decreases as the intensity increases. At high intensities the only observable effect is that from the relative scale between the two channels. The observed log-ratio for non-differentially expressed genes at high intensity is $M_\infty \approx \log_2\beta$. In the case of a linear transform ($a_R = a_G = 0$), α is (always) zero and M is therefore independent of A for *all* r. The remaining log-ratio bias is $\log_2\beta$. If a_R, $a_G > 0$, the "weakest" observable data point is $(A_0, M_0) = (1/2 \cdot \log_2(a_R a_G), \log_2(a_R/a_G))$, which is independent of both gene expression and scale parameters. All fold-change curves converge to this point. In the left graph of Figure 1 the effect of the affine transform $\mathcal{A}_1 = \{(a_G a_R) = (200,20), (b_G, b_R) = (1\text{-}4, 0.8)\}$ at different fold changes is depicted. The different curves plotted are the functions $M = m_r(A)$ for different fold changes. Note the asymmetry in curvature between up and down regulation. From the above discussion we know that the observed log-ratios are independent of the log-intensities for $log_2 r \approx -2.51$ with value $M_0 \approx -3.32$. The log-ratio for non-differentially expressed genes at high intensities is $M_\infty \approx -0.81$. A real-world example taken from [11], where the same array was scanned four times at various scanner PMT (sensitivity) settings, is shown in the right plot of Figure 1. Observed within-channel log-ratios $M_c = \log_2(y_c^{(v)} / y_c^{(w)})$ are plotted against the within-channel log-intensities $A_c = \log_2(y_c^{(v)}\ y_c^{(w)}) /2$ for the red channel ($c = R$) where $y_c^{(v)}$ and $y_c^{(w)}$ are observations at two different scanner PMT settings. In this case it turned out that all scans share the same offset. For more details, see [11]. For another example, see Figure 9.

Bias in the log-ratios

From (9) we see that the bias in the log-ratios introduced by the affine transform is intensity dependent. This non-linearity can be observed as a propeller shaped graph in Figure 2, where the log-ratios under the affine transform $\mathcal{A}_1$ are plotted against the true log-ratios at different log-intensity levels. If a regression line is fitted between the affine transformed log-ratios and the true log-ratios, the slope will always be *less* than one. Moreover, this is true for all normalization methods that do not overcompensate for channel offsets. This may explain why some studies show that cDNA microarrays tend to compress the absolute log-ratios compared to oligoarrays and QRT-PCR [16-18] including a recent study [19]; the channel offsets in cDNA microarrays are probably larger. When [20] compared cDNA microarray log-ratios to Northern blot log-ratios for their background correction method they found similar behavior, which emphasizes the close relationship between offset and background estimates. We will return to this later. The same patterns is seen in an *M* versus *M* scatter plot for non-normalized versus (affine) normalized data. See right scatter plot in Figure 2. To visualize the intensity dependency of the log-ratios, only data points at certain log-intensity levels are plotted. For details on data, see Methods.

Normalization in general

Depending on the design of the microarray experiment, we expect to observe different types of patterns in data. A typical example is where a subset of the genes studied is expected to be non-differentially expressed in a test sample compared to a reference sample. However, it is common that the patterns of the *observed* expression levels are not in line with the expected patterns of the *true* expression levels. Whenever this happens various strategies can be adopted in order to make the normalized data meet the expectations. Normalization of microarray data is about identifying and removing such artifactual variations that are not due to noise or natural variability. An example is the intensity-dependent log-ratio artifact.

In the following section we will, with the affine model in mind, revisit various more or less well known normalization methods that directly or indirectly remove intensity-dependent artifacts. With the gained knowledge, we then propose a generic and robust multi-dimensional normalization method for affine transformed data.

To be more precise in what follows, we will refer to methods that correct for differences in observed and expected data, that is, conform the signals to a standard or a norm, as *normalization methods*, where normalization has the meaning of conforming to expectations. Sometimes *calibration data*, also known as control data, which contains true relative or absolute expression levels, is available. Such data can be used to correct for discrepancies between observed and true expression levels. We refer to methods that use calibration (read *known*) data points to correct for artifacts as *calibration methods*. To this category we also count methods that are based on models for which we can find the inverse of the measurement function. For precise definitions, see the introduction of [21]. Calibration methods are not discussed further in this paper.

Typically a normalization method is only capable of estimating $\alpha = a_R - \beta a_G$ for $r = 1$ in (9) and not the individual offset terms. This is because the often used *assumption that most genes are non-differentially expressed* (and/or that there is an equal amount of up and down regulated genes) will only help us identify one fold-change curve, namely $\log_2 r = 0$. For a normalization method, like most calibration methods, to be able to estimate both a_R and a_G more constraints are needed and without known data this can only be done based on more assumptions. As more research is needed, we will not elaborate on such additional assumptions in this paper. Thus, the rest of this paper will only discuss normalization methods based on the commonly accepted assumption that it is possible to identify a set of genes that can be used to normalize the non-differentially expressed genes.

Curve-fit normalization revisited

When [4] first observed the intensity-dependent effects on the log-ratios they suggested a curve-fit normalization method that is often referred to as *lo(w)ess normalization*. The simplest version of this assumes that the majority of the genes are non-differentially expressed regardless of expression level and for this reason the log-ratios are expected to be centered around zero for all intensities. Under the above assumption, curves estimated using robust local regression methods such as lowess [22,23] or loess [24], or curves modeled by smoothing splines [25] will be good approximations for the $m_{r=1}(A)$ function, which then can be subtracted from the observed log-ratios

$$M \leftarrow M - m_{r=1}(A) = m_r(A) - m_{r=1}(A). \quad (11)$$

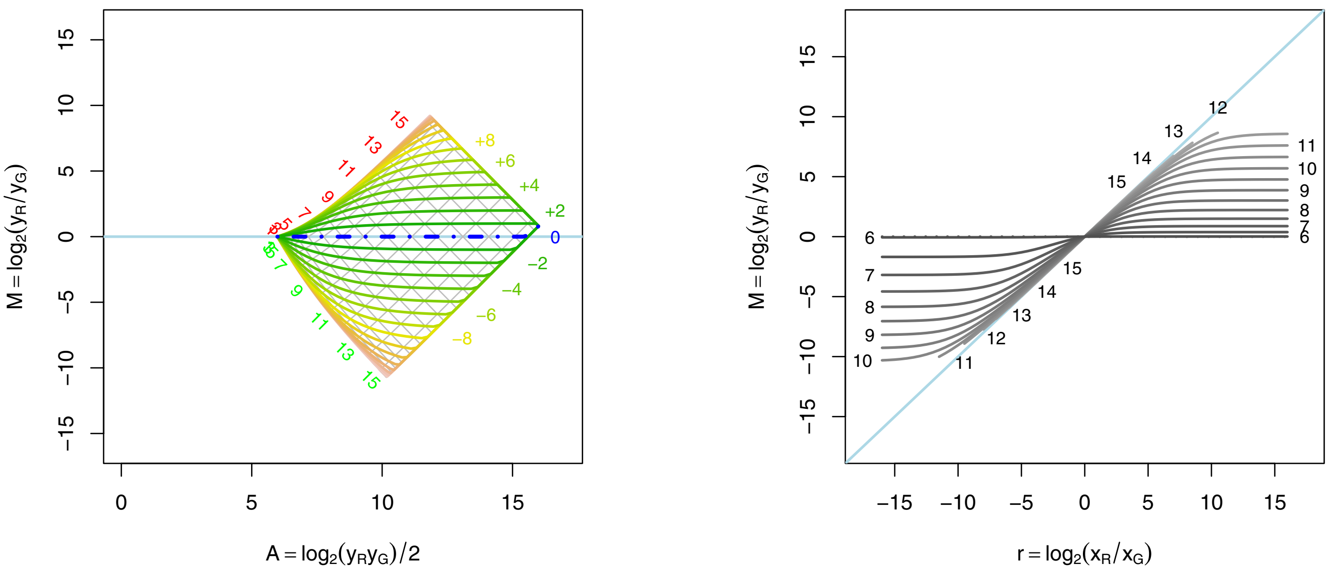

Figure 3

Curve-fit normalization of affine transformed data. Curve-fit normalization of $\mathcal{A}_1$ transformed data. *Left*: Log-ratios as a function of log-intensities for different fold changes. Note that the distance between up- and down-regulated genes at any intensity is the same before and after the normalization. *Right*: Normalized log-ratios versus true log-ratios. We see that intensity-dependent artifacts have been removed for the observed and true log-ratios where all curves intersect (here at (0, 0)).

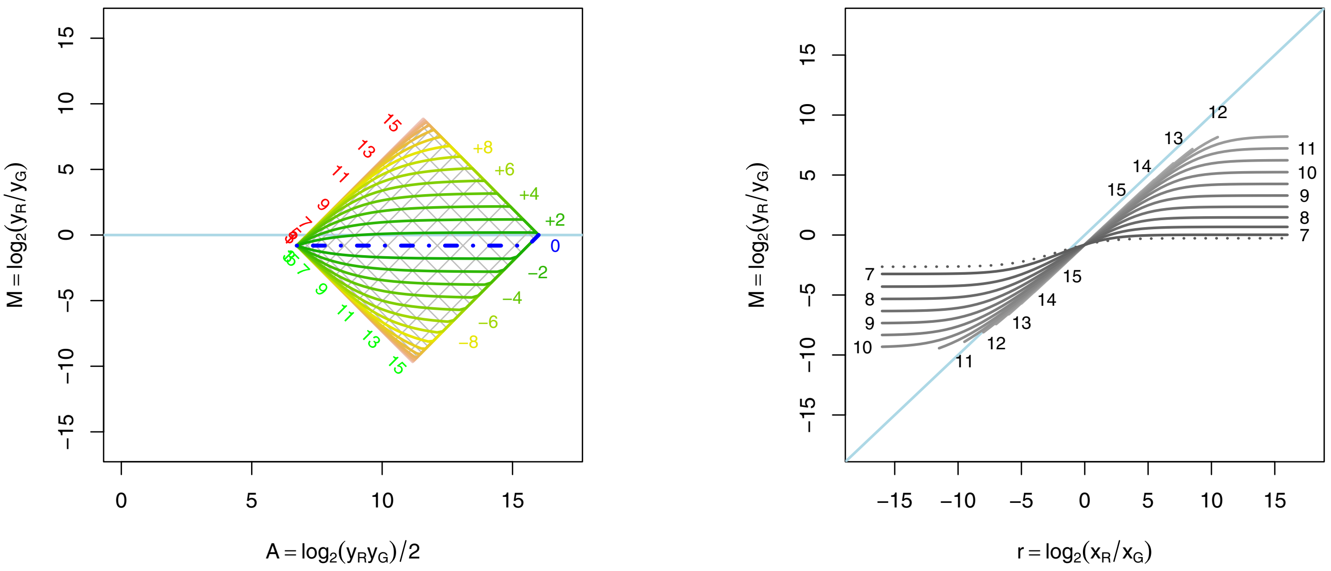

Figure 4

Perpendicular translation normalization of affine transformed data. Perpendicular translation normalization of $\mathcal{A}_1$ transformed data. The optimal amount of normalization shift in the raw data is a = 60, which corresponds to a'_R = 80 and a'_G = 140. *Left*: Log-ratios as a function of log-intensities for certain fold changes. The r = 1 curve (dot-dash blue) is horizontal, that is, for this specific value of r and a the log-ratios are independent of the log-intensities. *Right*: Normalized log-ratios versus true log-ratios. From this graph it is clear that we obtain the minimum error in log-ratios at zero-fold change. The dotted curves correspond to the minimum and maximum log-intensities possible to observe.

Under an affine transform, $m_r(A)$ and $m_{r=1}$ (A) are as in (9), but we do not know of a closed form expression for (11). An example of a curve-fit normalization under the affine transform is depicted in Figure 3. Note that the asymmetry between up- and down-regulated genes is *not* corrected for. Moreover, if we look at the overlaid ($\log_2 x_G$, $\log_2 x_R$) grid in the left graph of Figure 3, we find that the curve-fit normalization warps it and removes the otherwise orthogonal relationship between $\log_2 x_R$ and $\log_2 x_G$ (if the $(2A, M)$ plane is considered).

Perpendicular translation normalization revisited

The perpendicular (shift-log) normalization method proposed by [13] corrects for differences in the channel offsets. It normalizes log-ratios using a translation transform where a constant, $a \in \mathbb{R}$, is added to the signals in one channel and subtracted from the other;

$$\begin{aligned} y_{R,i} &\leftarrow a_R + b_R x_{R,i} + a;\ \forall i \\ y_{G,i} &\leftarrow a_G + b_G x_{G,i} - a;\ \forall i. \end{aligned} \qquad (12)$$

We refer to this translation normalization transform as the *perpendicular translation normalization*, because it moves (x_G, x_R) perpendicular to the $x_R = x_G$ line. From (9), we get that the observed log-ratios $m_r(A)$ can be made independent of the intensities if and only if

$$a = \frac{r b_R a_G - b_G a_R}{b_G + r b_R},\ r > 0. \qquad (13)$$

As this is a function of r, it is only for a single fold change at a time this method can make M independent of A. The most common choice is $r = 1$ for which the optimal perpendicular shift is

$$a = \frac{b_R a_G - b_G a_R}{b_G + b_R}, \qquad (14)$$

which is the weighted difference between a_R and a_G with weights $b_G/(b_G + b_R)$ and $b_R/(b_G + b_R)$, respectively. The distance from the $r = 1$ curve to the $M = 0$ curve for the optimal perpendicular shift is $\log_2 \beta$. In other words, the perpendicular shift normalization will not remove an overall bias in the log-ratios (although it is not hard to estimate β afterward). The optimal shift for $\mathcal{A}_1$ is $a = 60$ with $\log_2\beta = 0.57$. The result of this normalization is depicted in Figure 4. Note that $m_r(A)$ after normalization is constant for $r = 1$.

As suggested by [13], one way to find the optimal shift a is to minimize the curvature by minimizing the variation of the log-ratios after applying the shift a. To do this robustly, the median absolute deviation (MAD) can be used as a measure of variation;

$$\hat{a} = \arg\min_{a} \operatorname*{MAD}_{1 \le i \le I}(M_i(a)). \qquad (15)$$

We have found that the variance of $\hat{a}$ is unnecessarily large.

A problem with the perpendicular translation normalization methods, which is not related to estimator (15), is that the optimal shift can result in non-positive signals making a huge number of expression ratios invalid. The normalization method discussed next does not have this problem, but on the other hand, it will not work or work badly under certain conditions.

Parallel translation normalization revisited

For historical reasons, but also because it contributes to our discussion about background correction, the shift-log method proposed by [26] for stabilizing (read decreasing or shrinking) the variance of the measured log-ratios is of interest. A side effect of this method is that it can correct for intensity-dependent curvature. It is based on a translation transform where the same constant, $a \in \mathbb{R}$, is added to the signals in both channels;

$$\begin{aligned} y_{R,i} &\leftarrow a_R + b_R x_{R,i} + a;\ \forall i \\ y_{G,i} &\leftarrow a_G + b_G x_{G,i} + a;\ \forall i. \end{aligned} \qquad (16)$$

Because (16) moves data (x_G, x_R) parallel to the $x_R = x_G$ line, it is referred to as the *parallel translation normalization*. Again, as this is a function of r, M can only be made independent of A for one unique r at the time, cf. (9). For $r = 1$ the optimal parallel shift is

$$a = \frac{b_R a_G - b_G a_R}{b_G - b_R},\ b_G \neq b_R, \qquad (17)$$

which may be estimated as in (15). For example, for $\mathcal{A}_1$ the optimal parallel shift is $a = 220$ with the $r = 1$ curve 0.57 units below the $M = 0$ line. The result of this normalization is depicted in Figure 5. From the above expression, we also see that an optimal value of a can indeed be negative. For example, if $(a_G, a_R) = (200,140)$ and $(b_G, b_R) = (1\text{-}4, 0.8)$, the optimal parallel shift is $a = -60$, which corresponds to an effective shift of $(a'_G, a'_R) = (140, 80)$. However, it can also result in non-positive signals and therefore undefined log-ratios. For example, with $(a_G, a_R) = (20, 200)$ and $(b_G, b_R) = (1\text{-}4, 0.8)$, the optimal parallel shift is $a = -440$, which corresponds to an effective shift of

(a'_G , a'_R) = (-420, -240). Moreover, from (17) we see that when the scale parameters are equal there is no solution. This is because in such cases data is moved in parallel to the $x_R = x_G$ line making it impossible to get closer. As in the case of the perpendicular shift normalization, the distance between the r = 1 curve and the M = 0 curve is $\log_2 \beta$. Hence, a parallel shift normalization will not remove an overall bias in the log-ratios either and rescaling is necessary.

Single-channel translation normalization

A hybrid of the previous two methods is a normalization method that translates the signals in one of the channels at the time according to

$$\begin{aligned} y_{R,i} &\leftarrow a_R + b_R x_{R,i} + a \cdot \mathbb{I}(\mathrm{a} \geq 0);\ \forall i \\ y_{G,i} &\leftarrow a_G + b_G x_{G,i} - a \cdot \mathbb{I}(\mathrm{a} < 0);\ \forall i, \end{aligned} \qquad (18)$$

where $\mathbb{I}$ is the indicator function and $a \in \mathbb{R}$. This will not generate non-positive signals as only positive translations are applied. Moreover, because only one channel is shifted an optimal shift will always be found.

Rescale normalization

The above translation normalization methods remove curvature for non-differentially by adjusting the offset parameters in $\alpha = a_R - \beta a_G$ keeping the relative scale β fixed. Similarly, if the offset parameters are kept fixed, curvature can be removed by adjusting the relative scale β. In [11] we show that the scanner may introduce scale (PMT) insensitive (read *fixed*) biases to the channels. Thus, by adjusting the PMT settings such that the curvature of the pre-scanned data is as small as possible one minimizes $|\alpha| = |a_R - \beta a_G|$. Indeed, this strategy may in practice be used by many. However, from above we know that this can equally well be done numerically. It is much more important to adjust the PMT (and laser) settings such that the dynamical range of the signals is as large as possible. Furthermore, as scanner settings are often adjusted for each array separately, there will be a discrepancy between arrays, which in any case has to be normalized for.

Dye-swap normalization revisited

Dye-swap normalization, also known as *reverse labeling* and *paired-slides normalization*, is a balanced experimental design for two-color microarrays that can be used whenever two technically replicated hybridizations are available. Consider an experiment with two sets of cell populations, A and B, for which relative gene expressions, $\{r_i\}_i$, are to be investigated. After cDNA is obtained through reverse transcription, the two samples are each split into two identical parts, one which is labeled with a red fluorescent dye and one which is labeled with a green fluorescent dye. The red cDNA cocktail from sample A is mixed with the green ditto from sample B and co-hybridized to the DNA on the first array. After scanning, expression levels $\{(f_{G_1}(x_{B,i}), f_{R_1}(x_{A,i}))\}_i$ are observed. The same is done for the remaining red-green pair for which $\{(f_{G_2}(x_{A,i}), f_{R_2}(x_{B,i}))\}_i$ are observed. Dropping gene index i, the dye-swap normalization suggested by [27] is

$$\begin{aligned} M &= \frac{1}{2}(M_1 + M_2) \\ &= \frac{1}{2}\left(\log_2 \frac{f_{R_1}(x_A)}{f_{G_1}(x_B)} - \log_2 \frac{f_{R_2}(x_B)}{f_{G_2}(x_A)} \right) \\ &= \frac{1}{2}\left(\log_2 \frac{f_{R_1}(x_A)}{f_{R_2}(x_B)} + \log_2 \frac{f_{G_2}(x_A)}{f_{G_1}(x_B)} \right) \\ &= \frac{1}{2}(M'_1 + M'_2) \end{aligned} \qquad (19)$$

and similarly for the log-intensities

$$\begin{aligned} A &= \frac{1}{2}(A_1 + A_2) \\ &= \frac{\log_2(f_{R_1}(x_A) f_{G_1}(x_B)) + \log_2(f_{R_2}(x_B) f_{G_2}(x_A))}{4} \\ &= \frac{\log_2(f_{R_1}(x_A) f_{R_2}(x_B)) + \log_2(f_{G_2}(x_A) f_{G_1}(x_B))}{4} \\ &= \frac{1}{2}(A'_1 + A'_2). \end{aligned} \qquad (20)$$

Thus, the result of a dye-swap can be written as the average of two "virtual" hybridizations (A'_1 , M'_1) and (A'_2 , M'_2). Moreover, if (and only if) the measurement functions are equal for each array, that is, $f_{R_1} = f_{R_2}$ and $f_{G_1} = f_{G_2}$, then the observed ratios will be identical to the true ratios *for non-differentially expressed genes*. For this to be true for differentially expressed genes we know that they also have to be linear, that is, affine with zero intercept.

Several authors [28,29] have reported that dye-swap normalization does remove curvature, but less successful results have also been reported [30]. To better understand the reasons why and when dye-swap normalization works or not, we dissect the measurement functions f_c of the four channels $c = R_1, G_1, R_2, G_2$ into $(v_c \circ u_c \circ t_c \circ s_c)$ where s_c models the process of all steps up to the step where the (not yet labeled) cDNA sample is obtained, t_c models the labeling, u_c models the following steps including the hybridization, and v_c models the scanning etc. As channel R_1 and G_2 are from sample A and the other two are from sample B, we know that $s_{R_1} = s_{G_2} = s_A$ and $s_{R_2} = s_{G_1} = s_B$. Furthermore, if the labeling process is well controlled, we can assume that $t_{R_1} \approx t_{R_2} \approx t_R$ and $t_{G_1} \approx t_{G_2} \approx t_G$. When channel R_1 and G_1 are hybridized to array 1 and the other two to array 2 we have that $u_{R_1} \approx u_{G_1} \approx u_1$ and $u_{R_2} \approx u_{G_2} \approx u_2$. Moreover, if the same scanner settings are used for both arrays and everything else is equal, we have that $v_{R_1} \approx v_{R_2} \approx v_R$ and $v_{G_1} \approx v_{G_2} \approx v_G$. The overall measurement functions for the channels are then approximately

$$\begin{aligned} f_{R_1} &\approx v_R \circ u_1 \circ t_R \circ s_A \\ f_{G_1} &\approx v_G \circ u_1 \circ t_G \circ s_B \\ f_{R_2} &\approx v_R \circ u_2 \circ t_R \circ s_B \\ f_{G_2} &\approx v_G \circ u_2 \circ t_G \circ s_A. \end{aligned} \qquad (21)$$

For the dye-swap normalization to be efficient, we conclude that we must control the process of extracting the RNA etc. to an extent such that we can expect $s_A \approx s_B$. Moreover, we must also be able to reproduce hybridizations well enough such that $u_1 \approx u_2$. If these requirements are met, data will be self-normalized. Turning to the affine model, from (19) we have, if $f_{R_1} = f_{R_2}$ and $f_{G_1} = f_{G_2}$. that a dye-swap normalization of affine transformation data gives

$$\begin{aligned} M_1' &= \log_2 \frac{a_R + b_R x_A}{a_R + b_R x_B}, \\ M_2' &= \log_2 \frac{a_G + b_G x_A}{a_G + b_G x_B}, \end{aligned} \qquad (22)$$

and similar for A_1' and A_2'. For both virtual arrays, the signals in both channels have undergone identical affine transformations. We know from before that identical transformation in both channels does not introduce curvature for the non-differentially expressed genes and that symmetry between up- and down-regulated genes is preserved, cf. perpendicular and parallel shift normalization. If the offsets in any of the two replicated channels are not equal ($a_{R_1} \neq a_{R_2}$ or $a_{G_1} \neq a_{G_2}$), the dye-swap normalization will not work.

The above discussion assumed that the same cell samples have been replicated. If biological replicates are used, an additional source of variability is introduced. However, as long as it is possible to assume that for most genes $x_{A_1} \approx x_{A_2}$ and $x_{B_1} \approx x_{B_2}$. dye-swap normalization should still perform well.

In [11] we observed that scanners can introduce channel-specific offsets that are stable over time, i.e. $a_{R_1} = a_{R_2}$ and $a_{G_1} = a_{G_2}$. Assume that everything else is perfect, but the PMT is adjusted separately for each array resulting in $b_{R_1} / b_{R_2} \neq b_{R_2} / b_{R_1}$ so that (22) is not obtained. This may be a reason why dye-swap normalization sometimes fails.

Alternative dye-swap normalization

An alternative dye-swap normalization method is to average the observed expression levels *before* taking the logarithm

$$\begin{aligned} M &= \log_2 \frac{(f_{R_1}(x_A) + f_{G_2}(x_A))/2}{(f_{R_2}(x_B) + f_{G_1}(x_B))/2} \\ &= \log_2 \frac{f_{R_1}(x_A) + f_{G_2}(x_A)}{f_{R_2}(x_B) + f_{G_1}(x_B)}, \end{aligned} \qquad (23)$$

and analogously for A. This approach uses the *arithmetic mean* of the observed signals whereas the previous dye-swap method used the *geometric mean*. To be able to say more about the difference between the two approaches, we turn to the affine transformation for which we have

$$\begin{aligned} M &= \log_2 \frac{a' + b' x_A}{a' + b' x_B} \\ A &= \frac{\log_2 (a' + b' x_A)(a' + b' x_B)}{2} \end{aligned} \qquad (24)$$

where $a' = a_R + a_G$ and $b' = b_R + b_G$. Again, we note that the dye-swap method makes the transforms in the resulting two virtual channels equal. Comparing the bias in log-intensities between the geometrical and the arithmetical approaches, for the latter we have

$$A_0 = \log_2 \frac{a_R + a_G}{2} \qquad (25)$$

whereas for the former we have

$$A_0 = \log_2 \sqrt{a_R a_G}. \qquad (26)$$

Because $(a_R + a_G)/2 \geq \sqrt{a_R a_G}$, we conclude that the log-ratio biases are always larger for arithmetic than geometric

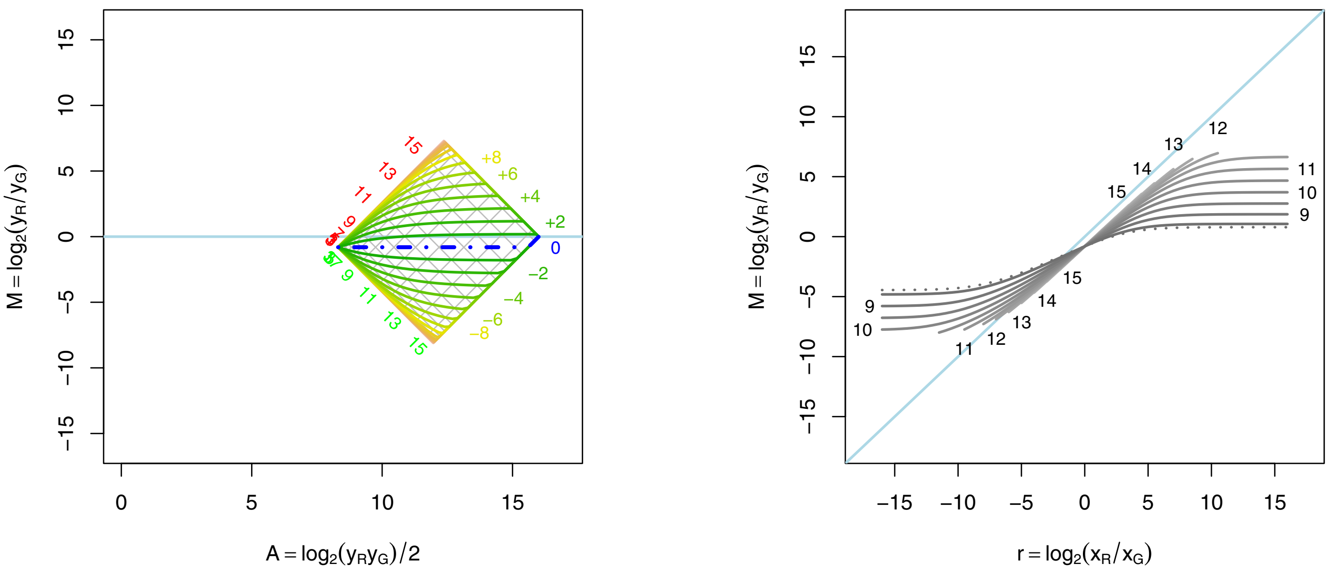

Figure 5

Parallel translation normalization of affine transformed data. Parallel translation normalization of $\mathcal{A}_1$ transformed data. The optimal amount of normalization shift in the raw data is *a* = 220, which corresponds to an effective shift of (a'_G, a'_R) = (420, 240). *Left*: Log-ratios as a function of log-intensities for certain fold changes. The *r* = 1 curve (dot-dash blue) is horizontal, that is, for this specific value of *r* and *a* the log-ratios are independent of the log-intensities. *Right*: Normalized log-ratios versus true log-ratios. From this graph it is clear that we obtain the minimum error in log-ratios at zero-fold change.

dye swap. However, there are other differences too. For instance, if each microarray glass array (the u_c functions above) introduces the same offset to both channels and this offset is different between arrays, but otherwise everything else is the same, that is, $a_{R_2} = a_{R_1} + a$ and $a_{G_2} = a_{G_1} + a$, then geometric dye-swap fails whereas arithmetic dye-swap succeeds to remove curvature.

Two-channel quantile normalization

Two-channel or in general multi-channel *quantile normalization* [31,32] is based on and relies on the *assumption that the true gene-expression levels in the two biological samples are approximately equally distributed*. If the measurement functions in the two channels, say f_R and f_G, are different, then the distributions of the measured signals in the two channels are different even if underlying distributions of true expression levels are identical. By estimating the distributions of the two channels and making them equal, for instance to an average distribution, the log-ratios for the *non-differentially* expressed genes will be unbiased and independent of the intensities. Thus, making the density functions of measured data equal for the two channels is the same as making their transformation functions equal, say to f_{RG}, which makes *M* independent of *A* for non-differentially expressed genes. If f_{RG} could be made linear too, this would be true for all fold changes.

For affine transformations, two-channel quantile normalization removes intensity-dependent effects, because the offsets $a_R$and a_G are identical after normalization. In addition, the constant log-ratio bias $\log_2\beta$ is also removed. Hence, two-channel quantile normalization can be considered to be both a method that corrects for differences in offset between two channels, but also a method that corrects for biases in the expression ratios. In Figure 6, the quantile normalization of $\mathcal{A}_1$ transformed data is depicted. The curvature for non-differentially expressed genes is removed.

Background subtraction as a normalization method

We have observed that log-ratios of *background signals* show the same intensity-dependent effects as ditto for *foreground signals* do, which suggests that background signals undergo the same transformation as foreground signals. An example of this is shown in Figure 7, where background and foreground estimates are plotted in the same *M* versus *A* scatter plots. A probable reason for this is the existence of scanner biases [11]. A widely adopted rationale for background correction is the assumption that the region that defines the spot is contaminated with the same physical noise that can be observed in the surrounding regions. Background noise is believed to be due to dust particles, DNA contaminated buffers, failed wash-

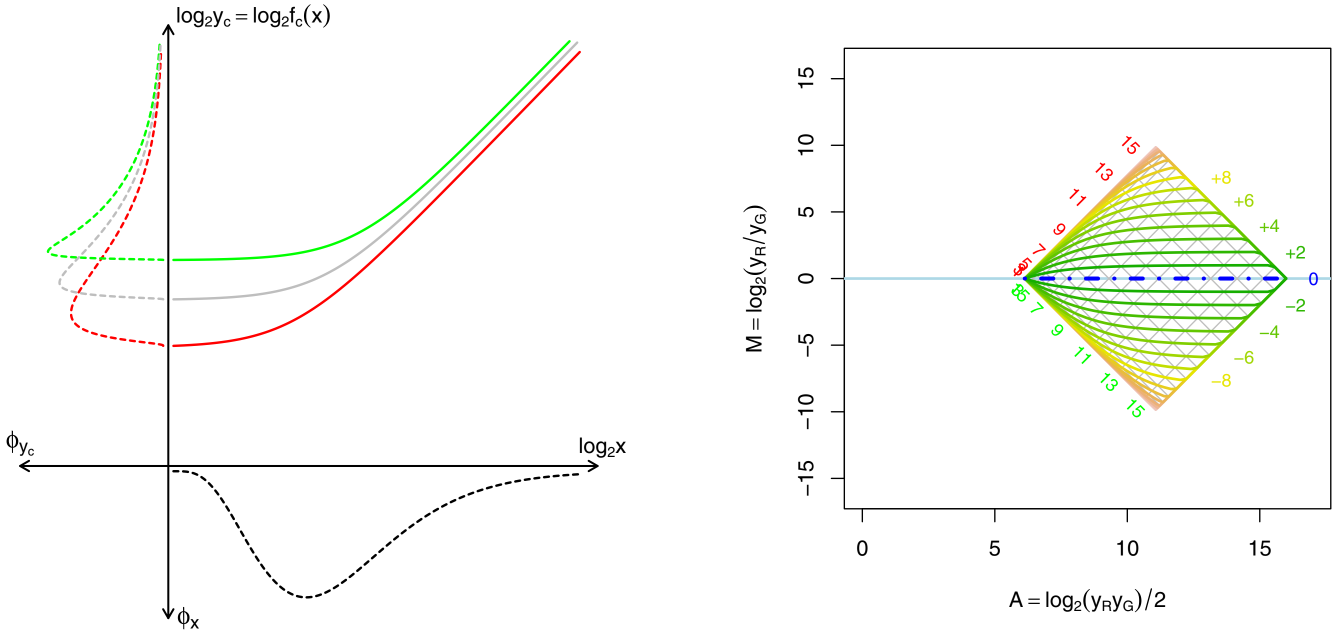

Figure 6

Equalizing the signal densities of the two channels removes the intensity dependency of the log-ratios for non-differentially expressed genes. Equalizing the signal densities of the two channels remove the intensity dependency of the log-ratios of non-differentially expressed genes. *Left*: Equal gene-expression distributions in both channels will under the non-channel balanced affine transform $\mathcal{A}_1$ turn into two different densities for the measured data. The (upside-down and dashed) curve at the bottom shows a hypothetical density function, $\phi_x(\cdot)$, of the true (log) gene-expression levels expected to be equal in both samples. The distributions of the affine transformed signals are shown in the (rotated and dashed) density functions, $\{\phi_{y_c}(\cdot)\}_c$, at the left (red and green curves). The average signal density (middle gray curve) to be normalized toward corresponds to a common measurement function (gray function in the main plot). *Right*: Normalizing the non-equal densities of the two channels makes the log-ratios of the non-differentially expressed genes zero for all intensities.

ing during printing or hybridization, cross hybridization etc. [20,33]. This type of background noise is often assumed to add to the foreground signal. Thus, in order to obtain true signals, background is subtracted from foreground signal as

$$y_{c,i} \leftarrow y_{c,i}^{(\mathrm{fg})} - y_{c,i}^{(\mathrm{bg})} \qquad (27)$$

where $y_{c,i}^{(\mathrm{fg})}$ is the estimated foreground signal and $y_{c,i}^{(\mathrm{bg})}$ is the estimated background signal for channel c and spot i. Under a transformation that is dominated by an affine function at lower intensities (of the same level as the background), subtracting the background from the foreground will shift the biases toward zero and background subtracted signals will have less curvature in the (A,M) plane than non-background subtracted signals (not shown). In this sense we can consider background subtraction to be a normalization method. However, just because the log-ratios as a function of the log-intensities become more flat, it does not imply that foreground regions are contaminated by the same noise as in background regions; unnecessary noise may be introduced. Instead, it may be that the background estimates from the image analysis *happen* to be close to a non-image-related offset in the foreground signals. Moreover, different image analysis software estimate the background signal differently based on different algorithms such as fixed-size circles, adaptive circles, morphological estimates, and pixel intensity distributions. Although comparative studies have been conducted [10,34], it is still not clear which background estimate is most correct. Some methods give higher background estimates than others, which means that they all correct for channel biases by different amounts, which by the way is another argument for why there exist channel offsets. makes use of this is [20], which emphasizes that the true signal can *not* be negative and uses a Bayesian approach to correct for this.

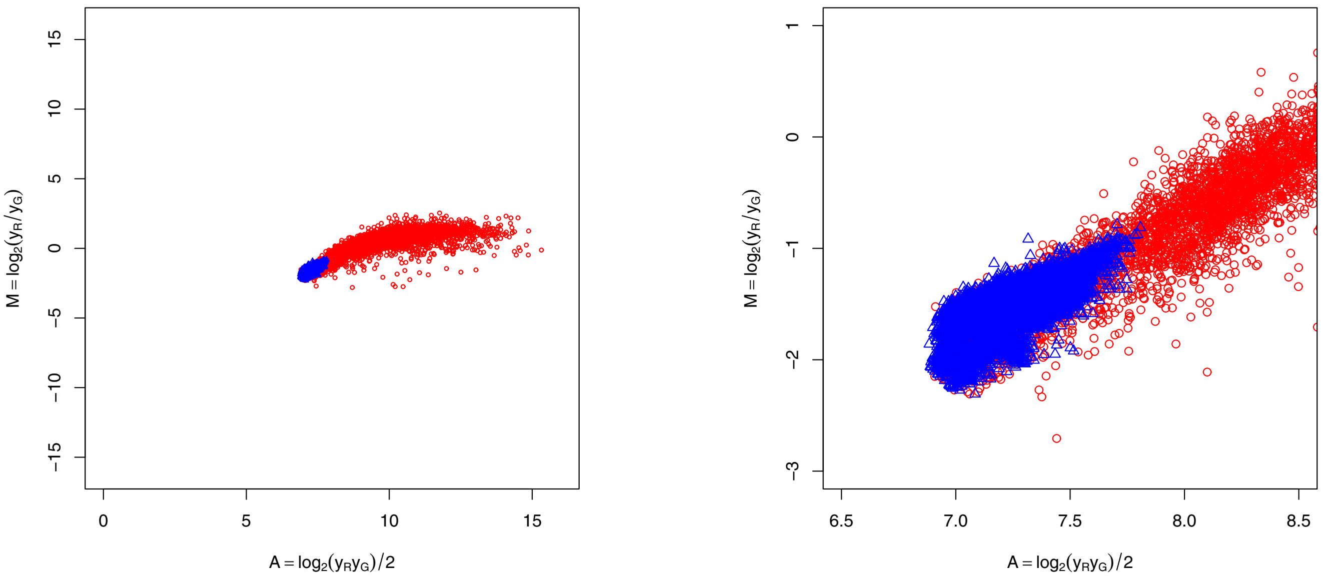

Figure 7

Transformation of background signal. Transformation of background signal. *Left*: An *M* versus *A* scatter plot where background signals (blue triangles) and foreground signals (red circles) lye along the same curve, which is evidence that both have been transformed identically. *Right*: A zoom-in of the left graph. Data is from [50].

Result of a (relative) negative translation

If too much background is subtracted, or a threshold has to be passed before the reverse transcription takes place, one can imagine that a_G, $a_R < 0$. Negative bias also applies if the observed signals are compared, not to the true signals, but to the signals obtained by another measuring technique that has a larger bias. Examples of such comparisons can be two-color microarray data compared to oligonucleotide (Affymetrix) data or two-color microarray data compared to QRT-PCR data. Negative bias may also be observed when control clones, spike-ins, negative and positive controls etc. are compared to the genes/ESTs of interest. The effect of a negative translation is depicted in Figure 8. The fan-out effect in the fold-change curves for the lower intensities is due to the negative translation. Note that this should not be mistaken for the fan-out effect due to decreasing signal-to-noise levels in the same way as lack of a fan-out effect due to a positive offset should not be mistaken for low noise.

Robust affine normalization

From the above discussion, it is clear that it is essential to correct for channel offsets when normalizing gene expression data. For two-channel data, we can obtain estimates of a_R, a_G and β as follows. For non-differentially expressed genes (without noise) we have that

$$y_{R,i} = \alpha + \beta y_{G,i};\ \forall i \qquad (28)$$

with $\alpha = a_R - \beta a_G$ and $\beta = b_R/b_G$. Define $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^{I}$ where $\mathbf{y}_i = (y_{G,i}, y_{R,i})$ and let

$$Q(\alpha,\beta;\mathbf{y}) = \sum_{i=1}^{I} w_i d_i(\alpha,\beta;\mathbf{y}_i)^2 \qquad (29)$$

be our objective function where $d_i(\alpha, \beta;\mathbf{y}_i) > 0$ is the orthogonal Euclidean distance between $\mathbf{y}_i$ and the line $L(\alpha, \beta)$ with intercept α and slope β. The estimates of α and β are then

$$(\alpha,\beta) = \arg\min_{(\alpha,\beta)} Q(\alpha,\beta;\mathbf{y}). \qquad (30)$$

With $w_i = 1$ for all observations we obtain standard principal component analysis (PCA), which minimizes the orthogonal distances in the L_2 norm [35]. With $w_i \neq 1$, (sample-) weighted PCA (WPCA), a special case of generalized PCA, is obtained [35,36]. With weights $w_i = 1/(d_i(\hat{\alpha}, \hat{\beta};\mathbf{y}_i) + \delta)$ we can minimize the distances in the L_1 norm, if we let $\delta \to 0^+$. The distance $d_i(\hat{\alpha}, \hat{\beta};\mathbf{y}_i)$, which equals the sum of squares of the values of all but the first principal component, was first suggested by [37]. Thus, our choice of weight function down-weigh outliers as defined by [37] in order to obtain a robust estimate of $L(\alpha, \beta)$ corresponding to the first principal component.

Our procedure is related to principal component analysis applied to an M-estimator of the covariance (scatter) matrix of data. The main difference is that we use weights $w = w(d_i) = 1/ (\delta + d_i)$ based on the orthogonal distance d_i from $\mathbf{y}_i$ to $L(\alpha, \beta)$ whereas for M-estimation one uses weights $w = w(d_i)$ based on a robustified Mahalanobis distance of $\mathbf{y}_i$, which is computed from an M-estimator of the covariance matrix of data. M-estimation of location and scatter was first defined by [38], and subsequently applied to principal component analysis by [39]. For other more recent papers on robust multivariate analysis, see [40,41] and the references therein. Alternative robust estimators can be obtained by choosing other weight functions $w(d_i)$, but we choose to optimize in L_1. Moreover, if one suspects a non-symmetric distribution of data points around the line, a trimmed version of the weight function may be considered. In practice, the above optimization can be performed by an *iterative reweighted principal component analysis* (IWPCA) scheme. For iteration $l = 1,2,\ldots,$ minimize (29) using WPCA where $w_i^{(1)} = 1$ and $w_i^{(l+1)} = 1/ (d_i(\alpha^{(l)},\beta^{(l)};\mathbf{y}_i) + \delta)$ with δ being a small positive number to avoid infinite weights.

As a last step, in order to get estimates of the four parameters a_R, a_G, b_R, and b_G from the two parameter estimates $\hat{\alpha}$ and $\hat{\beta}$, we introduce additional constraints. Let $\gamma_{c,(1)} = \min_i \gamma_{c,i}$ for $c = \{R, G\}$ and choose

$$\hat{b}_G = 1$$
$$\hat{b}_R = \hat{\beta} \qquad (31)$$

$$\hat{a}_G = \max\{a_G; a_G < \gamma_{G,(1)} \wedge \hat{\alpha} + \hat{\beta} a_G < \gamma_{R,(1)}\} \\ \hat{\alpha}_R = \hat{\alpha} + \hat{\beta}\hat{a}_G \qquad (32)$$

to be the estimates of the bias and the scale parameters in model (3). Constraint (32) is only correct in the noise-free case. If we allow noise, say

$$\gamma_{c,i} = a_c + b_c x_{c,i} + \varepsilon_{c,i}, \qquad (33)$$

where $E[\varepsilon_{c,i}] = 0$ and $V[\varepsilon_{c,i}] = \sigma_{c,i}^2$ for $c = \{R, G\}$, it is possible that the bias terms a_R and a_G are larger than the smallest observed value in the respective channel. This is especially important if the distributions of $\varepsilon_{c,i}$ for $c = \{R, G\}$ have heavy negative tails. An alternative, which introduces negative estimates, is to replace $\gamma_{c,(1)}$ in (32) with $\gamma_{c,(j)}$ for some order index (j) such that j - 1 non-positive signals are obtained in channel c. Choosing an optimal value on j is currently investigated by the authors, but beyond this article. Furthermore, it has been observed that the noise in each channel is roughly proportional to the signal strength, that is, $\sigma_{c,i} \propto x_{c,i}$. Thus, a positive side effect of the above estimation algorithm is that, contrary to have equal weights for all spots ($w_i = 1$), more weight will be given to low-intensity spots compared to high-intensity ones. This makes the method more robust to saturation and other non-linear effects that might occur at high intensities, effects for which classical line fits, which rely on homoscedasticity, would fail. Finally, with backward transformation (4) based on estimates ($\hat{a}_G$, $\hat{a}_R$, $\hat{b}_G$, $\hat{b}_R$), data is translated and rotated such that it falls around the diagonal line that goes through (0, 0) and (1, 1).

To illustrate the affine normalization method we have applied it to six two-color microarray data sets each containing 240 spike-in controls designed to have $\log_2 r = (-2, 0, +2)$ at various intensities. See also Methods. These controls were not used to estimate the normalization parameters. As shown in Figure 9, which is for one of the arrays, there is a small curvature for non-differentially expressed genes (and spike-ins) before normalization, a curvature that corresponds to $-\hat{\alpha} \approx +7 > 0$ (small positive derivative) at $\log_2 r = 0$, cf. (10). More importantly, the intensity dependent effect is profound for the $\log_2 r = \pm 2$ controls. Affine normalization allowing no negative signals removes curvature ($\alpha \approx 0$) for $\log_2 r = 0$, but not for the $\log_2 r = \pm 2$ controls, which indicates equal affine transformation in both channels, cf. right graph of Figure 6. If 5% negative signals is allowed, the log-ratios of all controls become roughly independent of intensity, which indicates that the observed signals are proportional to the concentrations of the spike-ins. All six arrays in this study show very similar properties.

Generalization to multiple channels and multiple arrays

A multi-dimensional version of the above algorithm can be summarized as follows. Say there are N arrays each hybridized with K samples (colors) such that there is in total $C = NK$ channels. Let $\mathbf{y}_i = (\gamma_{1,i},\ldots, \gamma_{K,i},\ldots, \gamma_{(N-1)K+1,i},\ldots, \gamma_{NK,i})$ be the NK observations for gene i. Thus, $\{\mathbf{y}_i\}_i$ spans an NK-dimensional space. Analogously to the above two-dimensional procedure, we can fit a robust line L through data in $\mathbb{R}^{NK}$ and constrain the estimate of $\mathbf{a} = (a_1,\ldots, a_{NK})$ by enforcing that $a < \mathbf{y}_i; \forall i$, where < is the component-wise inequality. Backward transformation (4) translates and

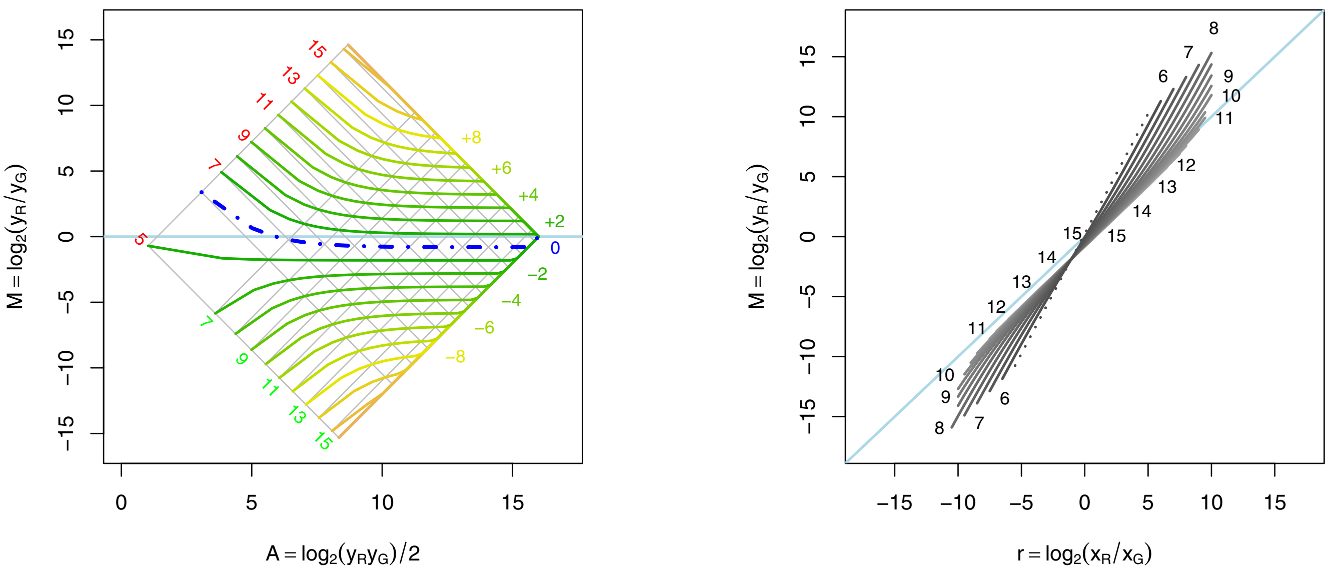

Figure 8
Affine transformation with negative translation. Affine transformation of the red and the green signals with negative translation where (a_G, a_R) = (-87, -24), (b_G,b_R) = (1.4,0.8). *Left*: Log-ratios as a function of log-intensities for certain fold changes. *Right*: Translated log-ratios versus true log-ratios. The slope of a line fitted in the *M* versus *M* plot will be *larger* than one, which is due to the negative translation. The grid and the fold-change curves in the left graph, and the intensity curves in the right graph have been truncated such that x_R,$x_G \geq 1$.

rotates data such that it lies along the diagonal line. By normalizing all arrays at once, signals from all hybridizations are brought onto the same scale and no further, so called, between-slide scale normalization is needed.

To apply the multi-dimensional normalization, the assumption that most genes are non-differentially expressed for *all* possible hybridization/channel pairs must be added. For most experimental setups this is not a problem. For instance, in two-channel microarrays experiments it is common to hybridize one test sample and one reference sample, which is selected such that it does not differ too much from the test sample, to the same array. The same reference is then used between arrays (in either channel). Thus, since each test-reference pair is "close" to each other, all test-test pairs should be approximately "close" to each other too. Alternatively, all reference channels can be normalized together. Then, keeping the reference signals fixed, each test channel is normalized toward the corresponding reference channel.

An implementation of the above algorithm is made available in the R [42] package named *aroma* [43], which is platform independent. In addition, the methods are available as an R plugin [44] for BASE [45]. A typically call is normalizeAffine(rg), which will normalize all arrays and all channels in the microarray object rg at once. The first parameter that has to be specified in the above algorithm is δ. However, its value is not critical and we have found that for instance δ = 0.02 works well in general and is therefore the default value. The second parameter to be specified is the number of negative signals allowed after normalization. By default the method allows 5% negative signals, but any fraction (or absolute number) of negative signals can be specified. Moreover, the method can be applied to any subsets of genes separately such as print-tip groups, clone groups and spike-ins. Finally, support for datapoint weights has been implemented so that the influence each spot has in the estimation procedure can be specified (not to be mistaken for the iterative weights above). Such weights may for instance be calculated from spot quality measures obtained by image analysis methods.

Discussion

If we compare the robust affine normalization method with the perpendicular and the parallel translation normalization methods optimized by minimizing the curvature, we find that there are similarities, because minimizing the curvature is identical to finding estimates of the bias parameters along the line $L(\alpha, \beta; \mathbf{y})$. Assuming a pure affine transformation, there are also similarities to the curve-fit method, which fits approximately the same line (curve) through data. The difference is how data is transformed to meet the assumptions. The affine method translates and rescales data in the original domain

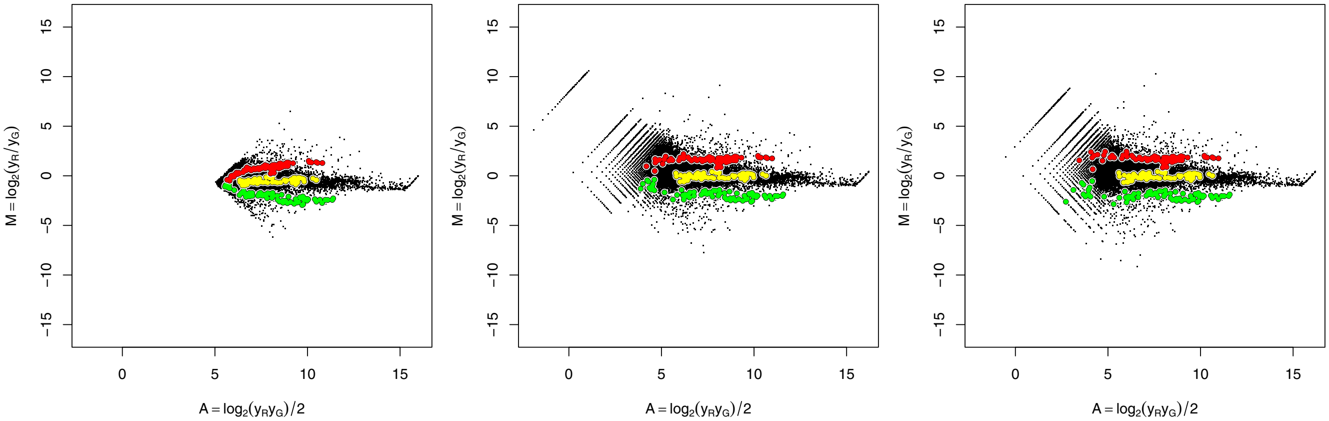

Figure 9

Log-ratios versus log-intensities before and after a robust affine normalization. Log-ratios versus log-intensities before and after a robust affine normalization. *Left*: Non-normalized data. Spike-ins designed to have $\log_2 r$ = +2, 0, and -2 are highlighted in red, yellow and green, respectively. *Middle*: Affine normalization utilizing constraint (32) resulting in no negative signals. Parameter estimates used in back transformation are ($\hat{a}_G$, $\hat{a}_R$, $\log_2 \hat{\beta}$) = (39.0, 22.0, -0.418). *Right*: Affine normalization where 5% (default) negative signals has been allowed; Parameter estimates used in back transformation are ($\hat{a}_G$, $\hat{a}_R$, $\log_2 \hat{\beta}$) = (45.7, 27.0, -0.418). The rotated binning effects of data points at low intensities are due to (unnecessary) rounding of average spot pixel intensity to nearest integer by the image analysis software.

whereas the curve-fit method operates in a rotated and log-transformed domain.

Moreover, the translation and the curve-fit methods rely on two-dimensional data (log-ratios) and it is not clear how to generalize them to multi-dimensional data, although re-iterative versions such as the cyclic loess [31] and the (multi-dimensional) contrast based method [46] have been suggested. Our affine normalization method is not limited to two-dimensional data, but can be applied to any number of channels, which means that three and four-color microarray data can be normalized as easily as two-color data.

It is interesting to note the close relationship between the quantile and the affine normalization method. In quantile normalization data points are shifted such that the sample densities of both channels are made identical. This results in new measurement functions, which may not be linear (or affine), but for which log-ratios for non-differentially expressed genes are zero. The affine normalization method can be though of as a quantile normalization method with special constraints on the underlying densities. An interesting continuation of the affine method and quantile normalization method is to relax the affine constraint by using other parametric or semi-parametric models. One possibility is to add smoothness constraints to the transformation functions using smoothing splines [25].

In previous sections, we did not discuss the variance stabilizing methods suggested by [12,47,48], which are based on error models that also contain channel-specific bias terms. Thus, those methods do indeed correct for intensity-dependent effects. Because they are based on specific error models and target hypothesis testing of non-differentially expressed genes, but also because they stabilize the log-ratio variances, they do not fit well into the above deterministic discussion. In addition, stabilizing the variance introduces bias for *differentially* expressed genes, which is not useful if absolute expression levels are of interest. However, we do believe that the directions drawn up by their underlying error models are promising.

Moreover, in the spirit of [20], it would be interesting to incorporate an empirical Bayes component to allow for non-positive signals more naturally.

An interesting study on microarray scanner calibration curves was published while submitting this article [19]. From their results on under-estimated log-ratios and propeller-shaped log-ratio versus log-ratio scatter plots, we suspect that they observe nothing but affine transformed signals. It would be of great interest to redo their analysis with affine normalization.

Finally, offset and scale parameters in (3) can be extended to incorporate, say, spatial structures by replacing them with $a_c(\mathbf{u}_i)$ and $b_c(\mathbf{u}_i)$ where $\mathbf{u}_i = (u_{i,x}, u_{i,y})$ is the spatial position of spot i.

Conclusion

We have proposed a robust non-parametric normalization method for affine transformed gene-expression data, which centers and symmetrizes log-ratios at all intensities. Symmetric log-ratios are fundamental for statistical tests on non-differentially expressed genes, typically utilizing t-tests or similar. In addition and contrary to other normalization methods (except quantile normalization), which are exclusively for paired channels, the method applies equally well to multi-array and multi-channel data. We believe that normalization based on affine transformations, such as our proposed IWPCA method, is very promising and has the potential of being used for many microarray applications. However, more comparison with other normalization methods is needed to fully understand its advantages and disadvantages.

Methods

Log-ratios as a function of log-intensities

Let $x_g = b_G x_G \geq 0$. Equation (6) for affine transformations (3) can then be written as

$$A = \frac{1}{2}\log_2[(a_R + r\beta x_g)(a_G + x_g)]$$

with $\beta = b_R/b_G$ and $r = x_R/x_G$. After a few steps, one gets that

$$x_g = (r\beta)^{-1}\left(-\frac{1}{2}(a_R + r\beta a_G) + \sqrt{\frac{1}{4}(a_R - r\beta a_G)^2 + r\beta 2^{2A}} \right).$$

It follows that

$$\begin{aligned} a_G + b_G x_G &= a_G + x_g \\ &= (r\beta)^{-1}\left(-\frac{1}{2}\alpha(r) + \sqrt{\frac{1}{4}[\alpha(r)]^2 + r\beta 2^{2A}} \right) \\ a_R + b_R x_R &= a_R + r\beta x_g \\ &= \frac{1}{2}\alpha(r) + \sqrt{\frac{1}{4}[\alpha(r)]^2 + r\beta 2^{2A}} \end{aligned}$$

with $\alpha(r) = a_R - r\beta a_G$. Equation (9) follows immediately.

Data

Arrays and hybridization

Six arrays were used in this study. The arrays contain Operon's Human Array-Ready Oligo Sets™ and 240 Stratagene SpotReport™ (Alien and Alien Oligo) control spots with layout of 12-by-4 print-tip groups each containing 25-by-25 spots. In total there are 30000 spots on each array. The arrays were produced by the SWEGENE DNA Microarray Resource Centre, Department of Oncology at Lund University using a MicroGrid II 600R arrayer fitted with MicroSpot 10 K pins (BioRobotics). Arrays were spotted on UltraGAPS™ coated slides (Corning Incorporated). Printing was performed in a temperature (18–20°C) and humidity (44–49% RH) controlled area. After printing was completed, arrays were left in a desiccator to dry for 48 hours, rehydrated for 1 second over steaming water, snap dried on a hot plate (98°C), UV-cross-linked (800 mJ/cm^2) and subsequently hybridized with various test and reference RNA samples. Samples and Stratagene RNA spikes were labeled, purified and hybridized using Pronto!™ Plus System 6 (Corning Incorporated) according to manufacturer's instructions.

Scanning and Image analysis

The arrays were scanned on an Agilent G2505A DNA microarray scanner (Agilent Technologies) at laser power and PMT gain both at 100% and scan resolution 10 μm/pixel. The so called *dark offset* intentionally added to all signals by the Agilent scanner [[49], p. 18] has been uninstalled. Multiscan calibration [11] was not used for this study.

The scanned images (65536 gray scales) were analyzed using the Axon GenePix Pro v4.1.1.40 software (Axon Instruments). The median spot pixel intensity was used for the foreground signal. Background estimates were not considered in this analysis. No spot signals were discarded.

Authors' contributions

HB drafted the first version of the manuscript. Both authors contributed equally to the study and the final version of the manuscript.

Acknowledgements

This work would not have been achieved without scientific support from Terry Speed at UC Berkeley and Walter and Eliza Hall Institute of Medical Research (WEHI), Patyaksha Wirapati (at the time at WEHI), Gordon Smyth (WEHI), and Halfdan Grage (at the time at Lund University). While at UC Berkeley (2000) and WEHI (2002), HB was financially supported by The Swedish Foundation for International Cooperation in Research and Higher Education (STINT), The Fulbright Commission, The Foundation Blanceflor Boncompagni-Ludovisi née Bildt, The Royal Swedish Academy of Sciences, and The Royal Physiographic Society in Lund. OH was financially supported by the Swedish Research Council. Microarray data was kindly provided by the SWEGENE DNA Microarray Resource Center at the BioMedical Center B10 in Lund, supported by the Knut and Alice Wallenberg foundation through the SWEGENE consortium. We also wish to thank the reviewers for feedback improving this manuscript.

References

1. Schena M, Shalon D, Davis RW, Brown PO: **Quantitative monitoring of gene expression patterns with a complementary DNA microarray.** *Science* 1995, **270(5235):**467-470.
2. Duggan DJ, Bittner M, Chen Y, Meltzer P, Trent JM: **Expression profiling using cDNA microarrays.** *Nature Genetics* 1999, **21(1 Supplement):**10-14.

3. Rocke DM, Durbin B: **A Model for Measurement Error for Gene Expression Arrays.** *Journal of Computational Biology* 2001, **8(6):**557-569.
4. Yang YH, Dudoit S, Luu P, Lin DM, Peng V, Ngai J, Speed TP: **Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation.** *Nucelic Acids Research* 2002, **30(4):**e15.
5. Bengtsson H: **Identification and normalization of plate effects in cDNA microarray data.** Preprints in Mathematical Sciences 2002:28, Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Sweden; 2002.
6. Ramdas L, Coombes KR, Baggerly K, Abruzzo L, Highsmith WE, Krogmann T, Hamilton SR, Zhang W: **Sources of nonlinearity in cDNA microarray expression measurements.** *Genome Biology* 2001, **2(11):**research0047.1-0047.7.
7. Li X, Gu W, Mohan S, Baylink DJ: **DNA microarrays: their use and misuse.** *Microcirculation* 2002, **9:**13-22.
8. Burle Industries Inc: **Photomultiplier Handbook.** Lancaster, PA, U.S.A.; 1980.
9. Handran S, Wang C, Aziz D: **Assessing Slide Flatness.** 2001.
10. Bengtsson A, Bengtsson H: **Microarray Image Analysis: Background Estimation using Quantile and Morphological Filters.** *BMC Bioinformatics* 2006, **7(1):**96.
11. Bengtsson H, Jönsson G, Vallon-Christersson J: **Calibration and assessment of channel-specific biases in microarray data with extended dynamical range.** *BMC Bioinformatics* 2004, **5(177):**.
12. Huber W, von Heydebreck A, Sültmann H, Poustka A, Vingron M: **Variance stabilization applied to microarray data calibration and to the quantification of differential expression.** *Bioinformatics* 2002, **18(Suppl 1):**S96-104.
13. Kerr MK, Afshari CA, Bennett L, Bushel P, Martinez J, Walker NJ, Churchill GA: **Statistical analysis of a gene expression microarray experiment with replication.** In *Tech rep* The Jackson Laboratory, Bar Harbor, Maine; 2001.
14. Cui X, Kerr MK, Churchill GA: **Data Transformations for cDNA Microarray Data.** In *Tech rep* The Jackson Laboratory, USA; 2002.
15. Callow M, Dudoit S, Gong E, Speed T, Rubin E: **Microarray Expression Profiling Identifies Genes with Altered Expression in HDL-Deficient Mice.** *Genome Research* 2000, **10(12):**2022-9.
16. Yue H, Eastman P, Wang B, Minor J, Doctolero M, Nuttall R, Stack R, Becker J, Montgomery J, Vainer M, Johnston R: **An evaluation of the performance of cDNA microarrays for detecting changes in global mRNA expression.** *Nucelic Acids Research* 2001, **29(8):**E41-1.
17. Yuen T, Wurmbach E, Pfeffer RL, Ebersole BJ, Sealfon SC: **Accuracy and calibration of commercial oligonucleotide and custom cDNA microarrays.** *Nucleic Acids Research* 2002, **30:**.
18. Barczak A, Rodriguez MW, Hanspers K, Koth LL, Tai YC, Bolstad BM, Speed TP, Erie DJ: **Spotted long oligonucleotide arrays for human gene expression analysis.** *Genome Research* 2003, **13(7):**1775-85.
19. Shi L, Tong W, Su Z, Han T, Han J, Puri RK, Fang H, Frueh FW, Goodsaid FM, Guo L, Branham WS, Chen JJ, Xu ZA, Harris SC, Hong H, Xie Q, Perkins RG, Fuscoe JC: **Microarray scanner calibration curves: characteristics and implications.** *BMC Bioinformatics* 2005, **6(Suppl 2):**S11.
20. Kooperberg C, Fazzio TG, Delrow JJ, Tsukiyama T: **Improved background correction for spotted DNA microarrays.** *Journal of Computational Biology* 2002, **9:**55-66.
21. Bengtsson H: **Low-level analysis of microarray data.** In *PhD thesis* Centre for Mathematical Sciences, Division of Mathematical Statistics, Lund University; 2004.
22. Cleveland W: **Robust locally weighted regression and smoothing scatterplots.** *Journal of American Statistics Association* 1979, **74:**829-836.
23. Cleveland W: **LOWESS: A program for smoothing scatterplots by robust locally weighted regression.** *The American Statistician* 1981, **35:**54.
24. Cleveland W, Grosse E, Shyu W: *Local regression models* MIT Press/ McGraw-Hill; 1992.
25. Green P, Silverman B: *Nonparametric Regression and Generalized Linear Models – A roughness penalty approach* Chapman and Hall; 1994.
26. Newton MA, Kendziorski CM, Richmond CS, Blattner FR, Tsui KW: **On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data.** *Journal of Computational Biology* 2001, **8:**37-52.
27. Yang YH, Dudoit S, Luu P, Speed TP: **Normalization for cDNA microarray data.** Technical Report 589, Department of Statistics, University of California at Berkeley; 2000.
28. Marton MJ, DeRisi JL, Bennett HA, Iyer VR, Meyer MR, Roberts CJ, Stoughton R, Burchard J, Slade D, Dai H, Jr DEB, Hartwell LH, Brown PO, Friend SH: **Drug validation and identification of secondary drug target effects using DNA microarrays.** *Nature Medicine* 1998, **4(11):**1293-1301.
29. Kerr MK, Martin M, Churchill GA: **Analysis of variance for gene expression microarray data.** *Journal of Computational Biology* 2000, **7:**819-837.
30. Tseng GC, Oh MK, Rohlin L, Liao JC, Wong WH: **Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects.** *Nucelic Acids Research* 2001, **29(12):**2549-2557.
31. Bolstad B, Irizarry R, Astrand M, Speed T: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias.** *Bioinformatics* 2003, **19(2):**185-93.
32. Yang YH, Thome NP: **Normalization for Two-color cDNA Microarray Data.** In *Science and Statistics: A Festschrift for Terry Speed, Monograph Series Volume 40*. Edited by: Goldstein DR. IMS Lecture Notes; 2003:403-418.
33. Schena M: *Microarrays Analysis* Hoboken, New Jersey: John Wiley & Sons, Inc.; 2003.
34. Yang YH, Buckley M, Dudoit S, Speed T: **Comparison of methods for image analysis on cDNA microarray data.** *Journal of Computational and Graphical Statistics* 2002, **11:**108-136.
35. Jolliffe I: *Principal Component Analysis* Springer series in statistics, Springer-Verlag New York Inc.; 1986.
36. Greenacre M: *Theory and Applications of Correspondence Analysis* London and Orlando: Academic Press; 1984.
37. Rao CR: **The use and interpretation of principal component analysis in applied research.** *Sankhya Series A* 1964, **26:**329-358.
38. Maronna RA: **Robust M-Estimators of Multivariate Location and Scatter.** *The Annals of Statistics* 1976, **4:**51-67.
39. Campbell NA: **Robust procedures in multivariate analysis. I. Robust covariance estimation.** *Applied Statistics* 1980, **29(3):**231-237.
40. Croux C, Haesbroeck G: **Principal Component Analysis based on Robust Estimators of the Covariance or Correlation Matrix: Influence Functions and Efficiencies.** 2000, **87:**603-618.
41. Pison G, Rousseeuw PJ, Filzmoser P, Croux C: **Robust factor analysis.** *J Multivar Anal* 2003, **84:**145-172.
42. R Development Core Team: **R: A language and environment for statistical computing.** R Foundation for Statistical Computing, Vienna, Austria; 2005. [ISBN 3-900051-07-0]
43. Bengtsson H: **aroma – An R Object-oriented Microarray Analysis environment.** Preprint in Mathematical Sciences 2004:18, Mathematical Statistics, Centre for Mathematical Sciences, Lund University, Sweden; 2004.
44. Bengtsson H: **aroma.Base – A generic R plugin dispatcher for BASE.** *online* 2005 [http://www.maths.lth.se/bioinformatics/].
45. Saal LH, Troein C, Vallon-Christersson J, Gruvberger S, Borg Å, Peterson C: **BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data.** *Genome Biology* 2002, **3(8):**SOFTWARE0003.
46. Åstrand M: **Contrast Normalization of Oligonucleotide Arrays.** *Journal of Computational Biology* 2003, **10:**95-102.
47. Durbin B, Hardin J, Hawkins D, Rocke D: **A variance-stabilizing transformation for gene-expression microarray data.** *Bioinformatics* 2002, **18:**S105-S110.
48. Rocke DM, Durbin B: **Approximate variance-stabilizing transformations for gene-expression microarray data.** *Bioinformatics* 2003, **19(8):**966-72.
49. Agilent Technologies Inc.: **Agilent G2565AA and Agilent G2565BA Microarray Scanner System – User Manual.** third, Palo Alto, CA; 2002. [G2566-90007]
50. Jögi A, Vallon-Christersson J, Holmquist L, Åke Borg HA, Påhlman S: **Human neuroblastoma cells exposed to hypoxia: induction of genes associated with growth, survival, and aggressive behavior.** *Experimental Cell Research* 2004, **295(2):**469-87.