BIOINFORMATICS

Vol. 00 no. 00 2006
Pages 1–7

Variance Stabilization and Normalization for One-Color Microarray Data Using a Data-Driven Multiscale Approach

E.S. Motakis [a], G.P. Nason [a*], P. Fryzlewicz [a] and G.A. Rutter [b]

[a]Department of Mathematics, [b]Department of Biochemistry, University of Bristol, UK.

ABSTRACT

Motivation: Many standard statistical techniques are effective on data that are normally distributed with constant variance. Microarray data typically violate these assumptions since they come from non-Gaussian distributions with a non-trivial mean-variance relationship. Several methods have been proposed that transform microarray data to stabilize variance and draw its distribution towards the Gaussian. Some methods, such as log or generalized log, rely on an underlying model for the data. Others, such as the spread-versus-level plot, do not. We propose an alternative data-driven multiscale approach, called the Data-Driven Haar-Fisz for microarrays (DDHFm) with replicates. DDHFm has the advantage of being "distribution-free" in the sense that no parametric model for the underlying microarray data is required to be specified nor estimated and hence DDHFm can be applied very generally, not just to microarray data.

Results: DDHFm achieves very good variance stabilization of microarray data with replicates and produces transformed intensities that are approximately normally distributed. Simulation studies show that it performs better than other existing methods. Application of DDHFm to real one-color cDNA data validates these results.

Availability: The R package of the Data-Driven Haar-Fisz transform (`DDHFm`) for microarrays is available in Bioconductor and CRAN.

Contact: g.p.nason@bristol.ac.uk

1 INTRODUCTION

Microarrays, in principle and in practice, are extensions of hybridization-based methods (Southern Blots, Northern Blots, SAGE etc), which have been used for decades to identify and locate mRNA and DNA sequences that are complementary to a segment of DNA (Alwin *et al*., 1977 and Velculescu *et al*., 1995). Microarray technology, in the form of either cDNA or High-Density Oligonucleotide arrays enables molecular biologists to measure simultaneously the expression level of thousands of genes. In a typical microarray experiment the aim is to compare different cell types, e.g. normal versus diseased cells, in order to identify genes that are differentially expressed in the two cell types.

Typically, microarray data analyses consist of several steps ranging from experimental design to the identification of important genes (for a review on the whole process see Sebastiani and Ramoni, 2003). Gene replication is a crucial design feature as it increases the precision of estimation and permits estimation of measurement variance which enables the significance of the final results to be judged.

Rocke and Durbin (2001) identified that the variance of the raw spot intensities increased with their mean and they modelled those intensities in terms of the two-component model:

$$Y_i = \alpha + \mu_i \times e^{\eta_i} + \epsilon_i, \qquad i = 1, ..., n \tag{1}$$

Here, $(Y_i)_{i=1}^n$ are the raw single-color intensities for the *n* genes, each assumed to be replicated p times. Sometimes we will write $Y_{r,i}$ when we are referring to the rth replicate on the ith gene ($r = 1, \ldots, p$). The α term represents the (common) mean background noise of the *n* genes on the array, μ_i is the true expression level for gene i, and η_i and ϵ_i are the normally distributed error terms with zero mean and variances σ_η^2 and σ_ϵ^2, respectively. In this way, $\mathbf{Y} = (Y_i)_{i=1}^n$ can be considered as coming from an *inhomogeneous process* that produces the *n* gene intensities with finite but different μ_i's and finite but different variances.

At low expression levels (i.e. μ_i close to 0) the measured expression Y_i in (1) can be written as $Y_i \approx \alpha + \epsilon_i$ so that Y_i is approximately distributed as N(α, σ_ϵ^2). On the other hand, for large μ_i's, the middle term in (1) dominates and Y_i can be modelled as:

$$Y_i \approx \mu_i e^{\eta_i} \tag{2}$$

with approximate variance

$$\mathrm{Var}(Y_i) \approx \mu_i^2 S_\eta^2 \tag{3}$$

where $S_\eta^2 = e^{\sigma_\eta^2}(e^{\sigma_\eta^2} - 1)$. For moderate values of μ_i, Y_i is modelled as in (1) with variance:

$$\mathrm{Var}(Y_i) = \mu_i^2 S_\eta^2 + \sigma_\epsilon^2 \tag{4}$$

From (3) and (4), we observe that the standard deviation (sd) of the Y_i increases linearly with their mean. Such mean-variance dependence, implying the presence of heteroscedastic intensities, is a major problem in the statistical analysis of microarrays.

Two methodological approaches have been followed to account for the heteroscedasticity. The first approach involves estimation of differentially expressed genes directly from the heteroscedastic data by means of penalized t-statistics (e.g. SAM method of Tusher *et al*., 2001), mixed or hierarchical Bayesian modelling (e.g. Baird *et al*., 2004 and Hsiao *et al*., 2004), appropriate Maximum Likelihood tests (e.g. Wang and Ethier, 2004) and, recently, gene grouping schemes (e.g. Comander *et al*., 2004 and Delmar *et al*., 2005a,2005b). The second approach, which we follow in this article, involves finding appropriate transformations that stabilize the variance of the data. After variance stabilization the data can be analyzed by standard, simple and universally accepted tools, like ANOVA models.

Section 2 outlines some existing variance stabilizing transforms that have been applied to microarray data. Section 3 proposes a new

*to whom correspondence should be addressed

method called the Data-Driven Haar-Fisz transform for microarrays (DDHFm) and compares its performance with existing methods by means of simulated and real cDNA data in Section 4. We show that DDHFm is superior to existing methods in terms of variance stabilization and Gaussianization of the transformed intensities.

2 ESTABLISHED VARIANCE STABILIZATION METHODS

For brevity we discuss and compare the performance of different variance stabilization techniques without, at this stage, worrying about differential expression. For this reason we consider data obtained from one-color microarrays. Generalization to two-color experiments will be considered in future work.

2.1 Log-based Transformations

Smyth *et al*. (2003) suggest using the log transform for microarray intensities. By assuming that the "lognormal distribution is an extremely good approximation to the bulk of the data" (Hoyle *et al*., 2002) as in model (2), the log transform $\log(Y_i)$ should stabilize the variance of the gene intensities and bring their distribution closer to the Gaussian. An extension of this approach then considers background corrected intensities, $\hat{Z}_i = Y_i - \hat{\alpha}$, which may be negative and cannot be handled by the simple $\log$ function. Based on this notion, several authors have studied alternative logarithmic-based transformations for microarray data.

Tukey (1977) defines the 'Started Log' transformation as: $\mathrm{sLog}(\hat{Z}) = \log(\hat{Z} + k)$ where k is a positive constant estimated via $\hat{k} = \hat{\sigma}_\epsilon^2 / 2^{1/4} \hat{\sigma}_\eta^2$, so that it minimizes the deviation from variance constancy. Alternatively, Holder *et al*. (2001) developed the Log-Linear Hybrid transformation as: $\mathrm{Hyb}_k(\hat{Z}) = \hat{Z}/k + \log(k) - 1$, for $\hat{Z} \leq k$ and $\mathrm{Hyb}_k(\hat{Z}) = \log(\hat{Z})$, for $\hat{Z} > k$. This transformation has also been called Linlog by Cui *et al*. (2003). As with sLog, the optimal k is estimated by $\hat{k} = \sqrt{2}\hat{\sigma}_\epsilon / \hat{\sigma}_\eta$.

2.2 The Generalized Logarithm Transformation (glog)

Munson (2001), Durbin *et al*. (2002) and Huber *et al*. (2002) independently developed the Generalized Logarithm transformation (referred to as glog in Rocke and Durbin, 2003). For data that come from model (1) with the mean-variance dependence (4), glog is assumed to produce symmetric transformed gene intensities with stabilized variance. The glog formula is:

$$\hat{Z} = \log\{(Y - \hat{\alpha}) + \sqrt{(Y - \hat{\alpha}) + \hat{c}}\} \tag{5}$$

where c is estimated by $\hat{c} = \hat{\sigma}_\epsilon^2 / \hat{S}_\eta^2$. Rocke and Durbin (2001) described algorithms to estimate α and c from one-color cDNA data. While estimation of α can be conducted without replicated genes, estimation of c involves estimation of S_η^2, which requires replication. Maximum Likelihood methods for c estimation only, based on Box and Cox (1964), were also developed by Durbin and Rocke (2003) for the case of two-colors microarrays and thus it is not relevant to the present work.

2.3 Spread-versus-Level Plot Transformation (SVL)

Archer *et al*. (2004) describes a different variance stabilization approach based on plotting the log-median of the replicated intensities on the x-axis (level) against the log of their fourth-spread (a variant of the interquantile range) on the y-axis (spread). Then the estimated slope of the subsequent linear regression model fit indicates the appropriate Box-Cox power transformation.

3 DATA-DRIVEN HAAR-FISZ TRANSFORMATION FOR MICROARRAYS

This section describes how the recent Data-Driven Haar-Fisz (DDHF) transform can be adapted for use with microarray data. Our adaption requires a subtle organization of microarray intensities into a form acceptable for the DDHF transform. We call our adaption the *DDHF transform for microarray data*, or DDHFm.

Recently, a new class of variance stabilization transforms, generically known as Haar-Fisz (HF) transforms, were introduced by Fryzlewicz and Nason (2004). In that work the HF transform used a multiscale technique to take sequences of Poisson random variables with unknown intensities into a sequence of random variables with near constant variance and a distribution closer to normality. Later Fryzlewicz *et al*. (2005) introduced the *Data-Driven* Haar-Fisz (DDHF) transform which used a similar multiscale transform but additionally estimated the mean-variance relation as part of the process of stabilization and bringing the distribution closer to normality. See also Fryzlewicz and Delouille (2005). Hence the DDHF transform can be used where there is a monotone mean-variance relationship but the precise form of the relationship is not known. In other words, DDHFm is "distribution-free" in that the precise data distribution, such as model (1), need not be known nor specified. See the Appendix for further details on the HF and DDHF transforms.

Both the HF and DDHF transforms rely on an input *sequence* of positive random variables X_i with mean μ_i and a variance σ_i^2 with some monotone (non-decreasing) relation between the mean and variance $\sigma_i^2 = h(\mu_i)$. Both HF and DDHF transforms work best when the underlying μ_i form a piecewise constant sequence. In other words, when consecutive μ are often very close or actually identical in value but large jumps in value are also permitted. However, microarray data are usually not organized in this sequential form. Microarray intensities Y_i usually come in replicated blocks: i.e. $Y_{r,i}$ is the rth replicate for the ith gene.

For the ith gene what we do know is that the underlying intensity $\mu_{r,i}$ for $Y_{r,i}$ is *identical* for each replicate r (this is the reason for replication). So, if the intensities for all replicates for a given gene i were laid out into a consecutive sequence we would *know* that their underlying μ_i sequence was constant.

To be able to make efficient use of the DDHF transform we would need to sort our intensities in order of increasing $\mu_{r,i}$ so that the sequence would be as near piecewise constant as possible. In actuality as we do not know the μ_i (since that is what we are trying to estimate) we cannot sort the sequence into increasing μ order. So, we do the next best thing in that we order the replicate sets according to their increasing mean observed value where the mean is taken across replicates. The idea is that the observed mean estimates the $\mu_{r,i}$ and observed mean ordering estimates the 'correct' true mean ordering. For example, suppose there were 4 replicates and 4 genes with observed (raw) intensities

	Rep 1	Rep 2	Rep 3	Rep 4	Means
Gene 1	13	12	13	14	13
Gene 2	10	11	12	11	11
Gene 3	100	102	99	103	101
Gene 4	73	74	74	75	74

Then ordering these replicates according to the means of replicates for each gene (indicated in the last column), and concatenating gives a sequence of:

10 11 12 11 13 12 13 14 73 74 74 75 100 102 99 103.

This ordered sequence of intensities within replicate blocks forms the input, denoted $(X_i)_{i=1}^n$ in the Appendix, to the DDHF transform. After transformation any further technique that has previously been applied to variance stabilized and normalized data may be applied here.

4 RESULTS

Durbin *et al*. (2002) and Rocke and Durbin (2003) compared the performance of glog with the background-uncorrected $\log$ (Log) and the background-corrected $\log$ (bcLog) transforms. By considering 18 deterministic μ values, each corresponding to a gene, they simulated $Y_{r,i}$ with $r = 1, \ldots, 1000$ and $i = 1, \ldots, 18$ intensities from the two-component model (1) with parameters $(\alpha, \sigma_\eta, \sigma_\epsilon) = (24800, 0.227, 4800)$ and assessed the performance of the methods in terms of the resulting transformed gene intensity variances and skewness coefficients. The two major results of Durbin *et al*. (2002) state that glog "stabilizes the asymptotic variance of microarray data across the full range of the data, as well as making the data more symmetric" than the other methods under comparison.

In Durbin *et al*. (2002) though, after simulating the intensities with the parameters mentioned above, the data were subsequently transformed using (5), with the *known* model parameters $(\alpha, \sigma_\eta, \sigma_\epsilon)$. This procedure is biased. In practice, the true parameters are not known and have to be estimated, which results in inferior overall variance stabilization performance. Below, we demonstrate this by simulating data from the two-component model *and* estimating the parameters.

Additionally, in our simulations described next, we also transform our data with the background uncorrected $\log$ (Log) method, the Log-Linear Hybrid transform, the Spread-Versus-Level transform and our new DDHFm method. We do not use background corrected Log and the Started Log, because both of them produce negative background corrected intensities, especially for small μ's, and we have observed that they result in highly asymmetric data.

4.1 One Color cDNA Data Acquisition

We simulate from the two component model (1) with parameters estimated from real cDNA data, obtained from the Stanford Microarray Database (http://smd.stanford.edu/). Two sets of data are considered. The first one comes from McCaffrey *et al*. (2004) study on mouse cDNA microarrays to investigate gene expression triggered by infection of bone marrow-derived macrophages with cytosol- and vacuole-localized Listeria monocytogenes (Lm). Each gene was replicated 4 times. The data set numbers were 40430, 40571, 34905 and 34912.

The second set comes from Pauli *et al*. (2006) work to identify genes expressed in the intestine of C.elegans using cDNA microarrays. Student t-tests for differential expression were conducted with 8 replicates for each gene. The data set numbers were 36590, 38262, 38265, 39215, 40157, 41833, 41834, 41886.

4.2 Simulations based on McCaffrey *et al*. (2004) data

We wish to simulate a likely μ_i signal using our real cDNA data. As in the example of Section 3, we estimate the mean of replicates for each gene from our two datasets. These means are ordered and concatenated in a single vector from which we sample 1024 equispaced values. This sequence of sample means, shown in Figure 1, forms

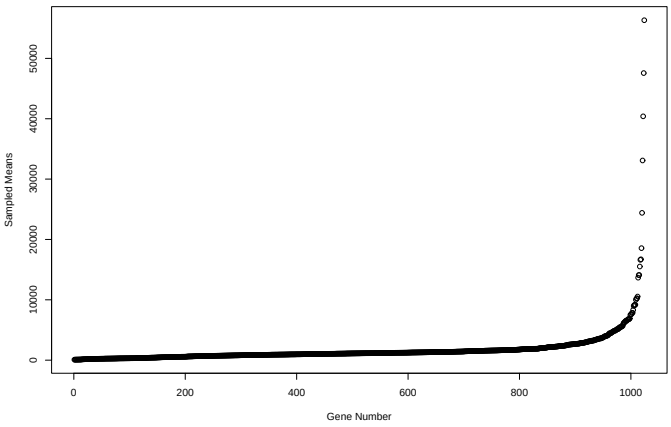

Fig. 1. Simulated μ signal of 1024 genes.

our simulated μ_i signal ("the truth"). This procedure is repeated for both real data sets.

From each of the 1024 μ_i levels we simulate $p = 4$ replicated raw intensities $Y_{r,i}$, where $r = 1, ..., 4$ and $i = 1, 2, ..., 1024$, using the `simdurbin2()` function from the `DDHFm` package which simulates from model (1). To obtain $Y_{r,i}$, model (1) was considered with parameters $\alpha = 340$, $\sigma_\eta = 0.9$ and $\sigma_\epsilon = 95$ as estimated (and rounded) from the McCaffrey *et al*. (2004) data set. These parameters are re-estimated as in Rocke and Durbin (2001), then applied to the transformation methods that require their estimation ($glog$ and Hyb) and the data are subsequently transformed. We iterate the above procedure $k = 1000$ times, and produce $Y_{r_k,i}$ raw intensities, where r_k denotes the r^{th} replicate of the k^{th} iterated sequence. Finally, we concatenate the transformed $Y_{r_k,i}$ into a single "output" vector for each *i*, from which we will derive our results. In other words, our output consists of 1024 output vectors $\underline{\mathrm{y}}_i$ of length $p \times k = 4000$ transformed observations.

The effectiveness of the methods is assessed in terms of adjusted sds ($\tilde{\sigma}_i$) of the replicated transformed intensities of each μ_i. Each $\tilde{\sigma}_i$ is computed as follows. The sd, σ_i, of the stabilized sample of 4000 values is computed for each μ_i. We noticed that each method stabilizes the variance to a different value. So, for each method we compute the mean of σ_i's over the whole μ_i set, denoted as $\bar{\sigma}$, and adjust each σ_i by computing $\tilde{\sigma}_i = \sigma_i / \bar{\sigma}$. In this way the different stabilization methods can be compared directly.

Additionally, we evaluate the Gaussianization properties of each transform by means of D'Agostino-Pearson K^2 test for normality (D'Agostino, 1971): the test is appropriate for detecting deviations from normality due to either "abnormal" skewness or kurtosis. Hence, when we subsequently write (not) normal we mean relative to this test. In contrast to the analysis of Durbin *et al*. (2002) on the means of skewness coefficients over 1000 samples for each μ, we choose this more comprehensive, distribution-based approach.

Figures 2–4 show the variance stabilization results of the transformation methods. Note that "$glog^i$" stands for the generalized logarithm transform with the known (optimal) parameters α, σ_η and σ_ϵ, while "$glog^e$" is the glog transform with all parameters being estimated. Additionally, "Hyb"=the Log-Linear Hybrid method,

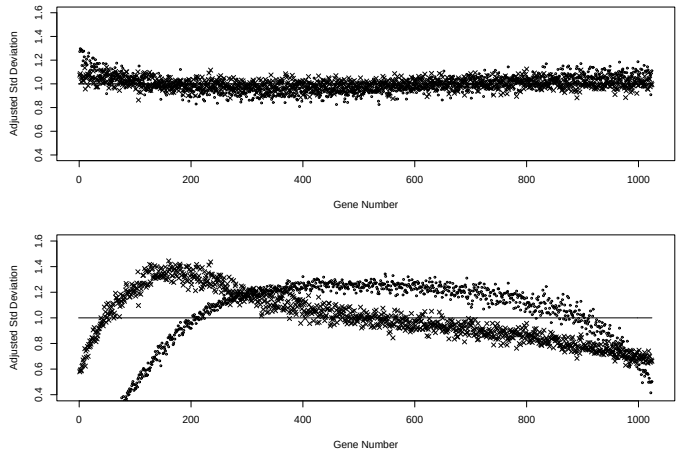

Fig. 2. Variance Stabilization of $glog^i$ (top) and $glog^e$ (bottom) transforms. Dots: $\sigma_\eta = 0.9$; Crosses: $\sigma_\eta = 0.3$. Horizontal line at 1. Each gene is replicated 4 times.

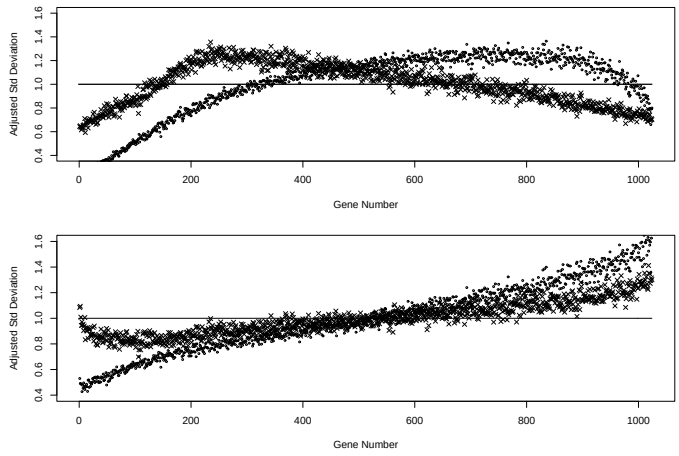

Fig. 3. Variance Stabilization of Hyb (top) and Log (bottom) transforms. Dots: $\sigma_\eta = 0.9$; Crosses: $\sigma_\eta = 0.3$. Horizontal line at 1. Each gene is replicated 4 times.

"Log"=the background uncorrected log transform, "SVL"=the Spread-Versus-Level transform and, finally, "DDHFm".

We plot the $\tilde{\sigma}_i$'s against the 1024 "mean-sorted" genes of data simulated first from $\sigma_\eta = 0.9$ (estimated from McCaffrey *et al.* (2004) data) and then from $\sigma_\eta = 0.3$ in order to show the performance of the methods with different choices of the model parameters. Varying α and σ_ϵ individually in the simulations did not yield different variance stabilization results from the ones reported here.

The more concentrated the $\tilde{\sigma}_i$'s are around 1 (the straight line in the figures), the better the stabilization has been performed. Figure 2 evidently shows the superiority of $glog^i$ over $glog^e$ for both σ_η values, indicating the direct effect on variance stabilization when the glog parameters are being estimated. The means of the estimated parameters over the $k = 1000$ sequences were estimated as $\bar{\alpha} = 430.22$, $\bar{\sigma}_\eta = 0.85$ and $\bar{\sigma}_\epsilon = 104.5$. Further analysis has showed that the large differences of the estimate $\hat{\alpha}$ from α, frequently observed over the k iterations, is the main cause of the degradation in $glog^e$ performance.

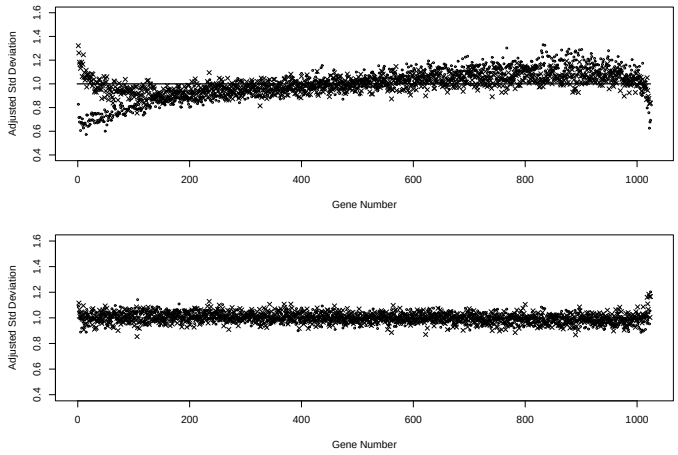

Fig. 4. Variance Stabilization of SVL (top) and DDHFm (bottom) transforms. Dots: $\sigma_\eta = 0.9$; Crosses: $\sigma_\eta = 0.3$. Horizontal line at 1. Each gene is replicated 4 times.

Figure 3 shows Hyb and Log variance stabilization results. Notice that both methods fail to stabilize the adjusted sds of the transformed intensities and, similarly to $glog^e$, their performance depends on the σ_η value: the smaller the σ_η gets, the better variance stabilization is achieved. For small σ_η though, Log seems to work better than the other two methods.

In Figure 4 we notice that SVL seems to perform well, especially for small σ_η, but its performance is still inferior to DDHFm. DDHFm clearly outperforms every other method and its variance stabilization results are very similar with those of $glog^i$ (but, of course, $glog^i$ uses known parameters and can not be used in practice).

Figures 5–6 show the Gaussianization results of SVL and DDHFm, which had the best variance stabilization performances. To produce the respective dotplots, we have estimated the D'Agostino-Pearson K^2 p-value for each set of transformed intensities. In the figures we present these 1024 p-values (dots) over the 1024 "mean-sorted" genes. We interpret p-values over 0.05 to indicate good Gaussianization and have plotted a horizontal line in the plots to aid interpretation.

We notice that SVL fails to normalize most of the transformed intensities for any σ_η. At $\sigma_\eta = 0.9$, DDHFm normalizes 55% percent of the transformed intensities but a slight downward trend is apparent, indicating that DDHFm normalization performance degrades as μ gets larger. For $\sigma_\eta = 0.3$, though, DDHFm normalizes the 91% of the transformed data with inexistence of a particular trend. DDHF normalizes better than SVL and outperforms the other transforms, due to its superior variance stabilization properties.

4.3 Simulations based on Pauli *et al*. (2006) data

We simulate, as before, $k = 1000$ sequences from $n = 1024$ genes. Here we replicate each gene $p = 8$ times in order to show the performance of selected methods when more replicates are available.

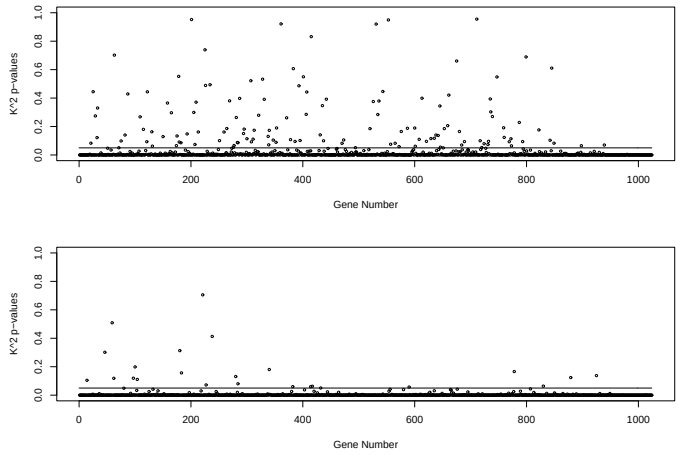

Fig. 5. Gaussianization of SVL transform. Top: $\sigma_\eta = 0.9$; Bottom: $\sigma_\eta = 0.3$. Horizontal line at 5%. Each gene is replicated 4 times.

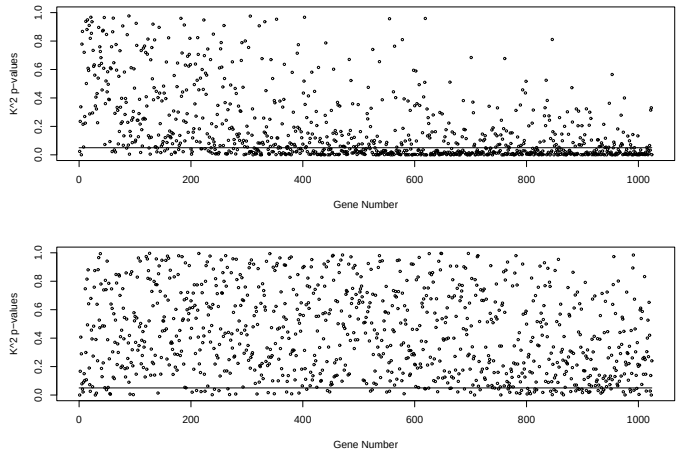

Fig. 6. Gaussianization of DDHFm transform. Top: $\sigma_\eta = 0.9$; Bottom: $\sigma_\eta = 0.3$. Horizontal line at 5%. Each gene is replicated 4 times.

We generate the μ signal and then simulate raw intensities from the two component model with parameters $\alpha = 900$, $\sigma_\epsilon = 196$ and $\sigma_\eta = 0.3$ derived from Pauli *et al*. (2006) cDNA data analysis. We compare $glog^e$, Log, SVL and DDHFm transforms, which for small σ_η produced the best results in the previous section.

The top section of Table 1 shows the summary statistics of the adjusted sds $\tilde{\sigma}_i$ of the transformed data for each method. Better concentration of the $\tilde{\sigma}_i$ around 1 suggests better variance stabilization. We observe that the best performance is achieved by DDHFm with approximately 3.5 times lower range and 4 times lower sd from the best competitor (Log transform).

The bottom section of Table 1 shows the K^2 p-value summary statistics. Again, DDHFm performs better than any other method. DDHFm also has the 1st Quantile (Q1) of its p-values distribution above 0.05.

Table 1. Summary statistics of the adjusted sds ($\tilde{\sigma}_i$) and K^2 p-values (K^2) for the various transforms.

		Min	Q1	Med	Q3	Max	SD
$\tilde{\sigma}_i$	$glog^e$	0.248	0.666	1.120	1.335	1.473	0.364
	Log	0.770	0.961	0.992	1.020	1.475	0.120
	SVL	0.830	0.915	0.967	1.040	1.512	0.121
	DDHFm	0.907	0.979	1.000	1.020	1.090	0.030
K^2	$glog^e$	0.000	0.000	0.000	0.000	0.981	$--$
	Log	0.000	0.007	0.165	0.503	0.998	$--$
	SVL	0.000	0.000	0.000	0.150	0.996	$--$
	DDHFm	0.000	0.085	0.291	0.597	0.995	$--$

4.4 Application to Real cDNA Data

In this section, we transform the McCaffrey *et al*. (2004) real cDNA data. The need for data transformation is suggested by a preliminary analysis which indicates that the replicate sd increases with the replicate mean.

We apply DDHFm, Log, SVL, and glog transforms to the data set and compute the adjusted replicate sds. Ideally, the five sequences of $\tilde{\sigma}_i$ should be as closely concentrated around one as possible.

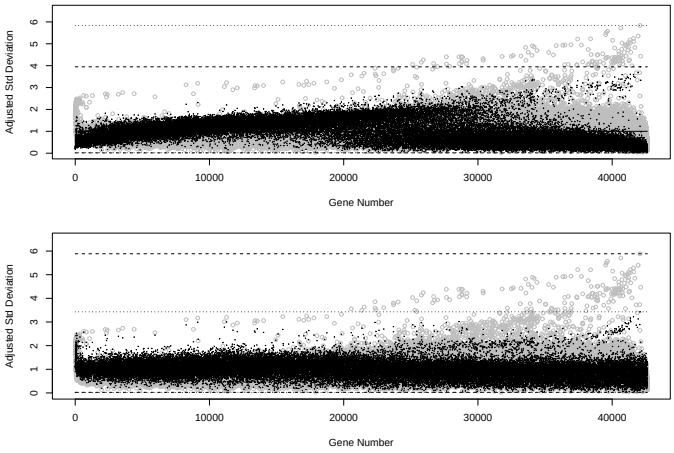

Fig. 7. Variance stabilization of glog (top/black), Log (top/grey), SVL (bottom/grey) and DDHFm (bottom/black) transforms. Dashed lines: range of glog (top) and SVL (bottom) adjusted sds; dotted lines: range of Log (top) and DDHFm (bottom) adjusted sds.

Figure 7 shows the variance stabilization results of the methods. Notice that DDHFm $\tilde{\sigma}_i$'s range approximately from 0 to 3.5 (the dotted lines in the bottom figure) with estimated sd of $\tilde{\sigma}_i$, $\hat{\sigma}_{\tilde{\sigma}_i} \simeq 0.35$, while the best competitor glog produces $\tilde{\sigma}_i$'s that range from 0 to 3.95 with $\hat{\sigma}_{\tilde{\sigma}_i} \simeq 0.51$. Log and SVL perform worse than glog (their $\tilde{\sigma}_i$'s range from 0 to 5.8 with $\hat{\sigma}_{\tilde{\sigma}_i} \simeq 0.46$). Since DDHFm produces $\tilde{\sigma}_i$'s that are more closely concentrated around 1 than of any of the competitors, we conclude that this is the best transformation for our data set.

5 CONCLUSIONS AND FURTHER RESEARCH

This article has introduced DDHFm, a new method for variance stabilization for replicated intensities that follow a non-decreasing mean-variance relationship. The DDHFm is self-contained and does not require any separate parameter estimation. The DDHFm is also "distribution-free" in the sense that a parametric model for intensities does not need to be pre-specified. Hence, it can be used in situations where there is uncertainty about the precise underlying intensity distribution.

Simulations have shown that DDHFm not only performs very good variance stabilization but also it produces intensities that have distribution much closer to the Gaussian when compared to other established methods.

The superior performance of DDHFm combined with its ability to adapt to a wide range of distributions with non-decreasing mean-variance relationship make it an ideal tool for variance stabilization for microarray data.

This paper has not addressed the separate, but related, issue of calibration (that is adapting to the over location and scale of separate slides). This is an issue for DDHFm but to judge from the results on stabilization not a significant issue. However, it would be possible to use DDHFm in conjunction with a calibration technique in a similar way to the combination of calibration and stabilization available in the vsn package described in Huber *et al*. (2003). We conjecture that stabilization would be again superior for DDHFm the use of DDHFm requires somewhat more computational effort than glog type methods. Our future aim is to investigate this more challenging problem as well as develop direct Haar-Fisz methods for calibration.

APPENDIX: THE DATA-DRIVEN HAAR-FISZ TRANSFORM

Let $\mathbf{X} = (X_i)_{i=1}^n$ denote an input vector to the Data-Driven Haar-Fisz Transform (DDHFT). The following list specifies the generic distributional properties of $\mathbf{X}$.

1. The length n of $\mathbf{X}$ must be a power of two. We denote $J = \log_2(n)$. In practice, if our data is not of length 2^J, then we reflect the end of our data set in a mirror-like fashion so that the "padded" sequence has a length which is a power of two.
2. $(X_i)_{i=1}^n$ must be a sequence of independent, nonnegative random variables with finite positive means $\rho_i = \mathbb{E}(X_i) > 0$ and finite positive variances $\sigma_i^2 = \mathrm{Var}(X_i) > 0$.
3. The variance σ_i^2 must be a non-decreasing function of the mean ρ_i: we must have $\sigma_i^2 = h(\rho_i)$, where the function h is independent of i.

For example, let $X_i \sim \mathrm{Pois}(\lambda_i)$. In this case, $\rho_i = \lambda_i$ and $\sigma_i^2 = \lambda_i$, which yields $h(x) = x$. Naturally, in many practical situations the exact form of h is unknown and needs to be estimated. Below, we describe the Haar-Fisz Transform (HFT) in the cases where h is known and unknown, respectively. (For microarrays the DDHF transform is modified and the ρ_i are sorted to minimize variation of the function ρ_i, see Section 3.)

We first recall the formula for the Haar Transform (HT). The HT is a linear orthogonal transform $\mathbb{R}^n \to \mathbb{R}^n$ where $n = 2^J$. Given an input vector $\mathbf{X} = (X_i)_{i=1}^n$, the HT is performed as follows:

1. Let $s_i^J = X_i$.
2. For each $j = J-1, J-2, \ldots, 0$, recursively form vectors $\mathbf{s}^j$ and $\mathbf{d}^j$:

$$s_k^j = \frac{s_{2k-1}^{j+1} + s_{2k}^{j+1}}{2}\,;\; d_k^j = \frac{s_{2k-1}^{j+1} - s_{2k}^{j+1}}{2},\; k = 1, \ldots, 2^j.$$

The operator H, where $H\mathbf{X} = (\mathbf{s}^0, \mathbf{d}^0, \ldots, \mathbf{d}^{J-1})$, defines the HT. The inverse HT is performed as follows:

1. For each $j = 0, 1, \ldots, J-1$, recursively form $\mathbf{s}^{j+1}$:

$$s_{2k-1}^{j+1} = s_k^j + d_k^j\,;\; s_{2k}^{j+1} = s_k^j - d_k^j,\; k = 1, \ldots, 2^j.$$

2. Set $X_i = s_i^J$.

The elements of $\mathbf{s}^j$ and $\mathbf{d}^j$ have a simple interpretation: they can be thought of as "smooth" and "detail" (respectively) of the original vector $\mathbf{X}$ at scale 2^j.

We now introduce the HFT: a multiscale algorithm for (approximately) stabilizing the variance of $\mathbf{X}$ and bringing its distribution closer to normality.

The main idea of the HFT is to decompose $\mathbf{X}$ using the HT, then "Gaussianise" the coefficients d_k^j and stabilize their variance, and then apply the inverse HT to obtain a vector which is closer to Gaussianity and has its variance approximately stabilized. We now describe the middle step: the variance stabilization and "Gaussianisation" of d_k^j.

Consider first $d_1^{J-1} = (X_1 - X_2)/2$. Suppose for now that X_1, X_2 are identically distributed (i.d.): indeed, this is likely if the underlying mean $\{\rho_i\}_i$ is e.g. piecewise constant. This implies that d_1^{J-1} is symmetric around zero. We want to stabilize the variance of d_1^{J-1} around $2^{(J-1)-J} = 1/2$. To do so, we divide d_1^{J-1} by $2^{1/2}$ times its own sd. Using the assumption of independence (item 2, first list of this section above) we have

$$\mathrm{Var}(d_1^{J-1}) = 1/4\,(\mathrm{Var}(X_1) + \mathrm{Var}(X_2)) = \sigma_1^2/2,$$

which gives $2^{1/2}\left(\mathrm{Var}(d_1^{J-1})\right)^{1/2} = \sigma_1 = h^{1/2}(\rho_1)$. In practice ρ_1 is unknown and we estimate it locally by $\hat{\rho}_1 = (X_1 + X_2)/2 = s_1^{J-1}$. The (approximately) variance-stabilized coefficient f_1^{J-1} is given by $f_1^{J-1} = d_1^{J-1}/h^{1/2}\left(s_1^{J-1}\right)$ (where the convention $0/0 = 0$ is used).

Turning now to $d_1^{J-2} = (X_1 + X_2 - X_3 - X_4)/4$, we also first assume that the X_1, X_2, X_3, X_4 are i.d. In order to stabilize the variance of d_1^{J-2} around $2^{j-J} = 2^{J-2-J} = 1/4$, we divide d_1^{J-2} by 2 times its sd. We have $2\left(\mathrm{Var}(d_1^{J-2})\right)^{1/2} = \sigma_1 = h^{1/2}(\rho_1)$ as before, and we estimate ρ_1 locally by s_1^{J-2}, which yields an approximately variance-stabilized coefficient $f_1^{J-2} = d_1^{J-2}/h^{1/2}\left(s_1^{J-2}\right)$. Asymptotic Gaussianity and variance stabilization of random variables of a form similar to f_k^j were studied by Fisz (1955): hence we label f_k^j the *Fisz coefficients* of $\mathbf{X}$, and the whole procedure — the *Haar-Fisz transform* of $\mathbf{X}$.

We now give the general algorithm for the Haar-Fisz transform when the function h is known.

1. Let $s_i^J = X_i$.
2. For each $j = J-1, J-2, \ldots, 0$, recursively form vectors $\mathbf{s}^j$ and $\mathbf{f}^j$:

$$s_k^j = \frac{s_{2k-1}^{j+1} + s_{2k}^{j+1}}{2}\,;\; f_k^j = \frac{s_{2k-1}^{j+1} - s_{2k}^{j+1}}{2h^{1/2}\left(s_k^j\right)},\; k = 1, \ldots, 2^j\,.$$

3. For each $j = 0, 1, \ldots, J-1$, recursively modify $\mathbf{s}^{j+1}$:

$$s^{j+1}_{2k-1} = s^j_k + f^j_k \; ; \; s^{j+1}_{2k} = s^j_k - f^j_k, \; k = 1, \ldots, 2^j \; .$$

4. Set $\mathbf{Y} = \mathbf{s}^J$.

The relation $\mathbf{Y} = F_h \, \mathbf{X}$ defines a nonlinear, invertible operator F_h which we call *the Haar-Fisz transform (of* $\mathbf{X}$*) with link function* h.

In practice h is often unknown and needs to be estimated. Since $\sigma_i^2 = h(\rho_i)$, ideally we would wish to estimate h by computing the empirical variances of $X_1, X_2, \ldots$ at points $\rho_1, \rho_2, \ldots$, respectively, and then smoothing the observations to obtain an estimate of h. Suppose for the time being that the ρ_i's are known and, as an illustrative example, consider $\rho_i = \rho_{i+1}$. The empirical variance of X_i can be pre-estimated, for example, as $\hat{\sigma}_i^2 = (X_i - X_{i+1})^2/2$. Note that on any piecewise constant stretch, our pre-estimate is exactly unbiased. The above discussion motivates the following regression setup:

$$\hat{\sigma}_i^2 = h(\rho_i) + \varepsilon_i,$$

where $\varepsilon_i = \hat{\sigma}_i^2 - \sigma_i^2 = (X_i - X_{i+1})^2/2 - \sigma_i^2$ and "in most cases" $\mathbb{E}(\varepsilon_i) = 0$. Of course, in practice, the ρ_i's are not known and, since we pre-estimate the variance of X_i using X_i and X_{i+1}, it also makes sense to pre-estimate ρ_i by $\hat{\rho}_i = (X_i + X_{i+1})/2$. Note that for each $k = 1, \ldots, 2^{J-1}$, we have $\hat{\rho}_{2k-1} = s_k^{J-1}$ and $\hat{\sigma}^2_{2k-1} = 2(d_k^{J-1})^2$, which leads to our final regression setup

$$2(d_k^{J-1})^2 = h(s_k^{J-1}) + \varepsilon_k. \qquad (6)$$

In other words, we estimate h from the finest-scale Haar smooth and detail coefficients of $(X_i)_{i=1}^n$, where the smooth coefficients serve as pre-estimates of ρ_i and the squared detail coefficients serve as pre-estimates of σ_i^2.

As we restrict h to be a non-decreasing function of ρ, we choose to estimate it from the regression problem (6) via least-squares isotone regression, using the "pool-adjacent-violators" algorithm described in detail in Johnstone and Silverman (2005), Section 6.3. The resulting estimate, denoted here by $\hat{h}$, is a non-decreasing, piecewise constant function of ρ.

The DDHFT is performed as above except that $\hat{h}$ replaces h.

ACKNOWLEDGEMENTS

ESM is the grateful recipient of a Wellcome Prize Studentship awarded to GAR and GPN. GPN was partially supported by an EPSRC Advanced Research Fellowship.

REFERENCES

Alwin, J.C., Kemp, D.J. and Stark, G.R. (1977) Methods for detection of specific RNAs in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with DNA probes. *Proc. Natl. Acad. Sci. USA*, **74**, 5350–5354.

Archer, K.J., Dumur, C.I. and Ramakrishnan, V. (2004) Graphical technique for identifying a monotonic variance stabilizing transformation for absolute gene intensity signals. *BMC Bioinformatics*, **5**:60.

Baird, D., Johnstone, P. and Wilson, T. (2004) Normalization of microarray data using a spatial mixed model analysis which includes splines. *Bioinformatics*, **20**, 3196–3205.

Box, G.E.P. and Cox, D.R. (1964) An analysis of transformations. *J. Roy. Statist. Soc.* B, **26**, 211–252.

Comander, J., Sripriya, N., Gimbrone, M.A. and García-Cardeña, G. (2004) Improving the statistical detection of regulated genes from microarray data using intensity-based variance estimation. *BMC Genomics*, **5**:17.

Cui, X., Kerr, M.K. and Churchill, G.A. (2003) Transformations for cDNA microarray data. *Statist. App. Gen. Mol. Biol.*, **2**:4.

D'Agostino, R.B. (1971) An omnibus test of normality for moderate and large size samples. *Biometrika*, **58**, 341–348.

Delmar, P., Robin, S., Tronik-Le Roux D. and Daudin J.J. (2005a) Mixture model on the variance for the differential analysis of gene expression data. *J. Roy. Statist. Soc.* C, **54**, 31–50.

Delmar, P., Robin, S. and Daudin, J.J. (2005b) VarMixt: efficient variance modelling for the differential analysis of replicated gene expression data. *Bioinformatics*, **21**, 502–508.

Durbin, B.P., Hardin, J.S., Hawkins, D.M. and Rocke, D.M. (2002) A variance-stabilizing transformation for gene expression microarray data. *Bioinformatics*, **18**, S105–S110.

Durbin, B.P. and Rocke, D.M. (2003) Estimation of transformation parameters for microarray data, Bioinformatics, **19**, 1360–1367.

Fisz, M. (1955) The limiting distribution of a function of two independent random variables and its statistical application. *Colloquium Mathematicum*, **3**, 138–146.

Fryzlewicz, P. and Delouille, V. (2005) A data-driven Haar-Fisz transform for multiscale variance stabilization. To appear in *Proc. of the 13th IEEE Workshop on Statistical Signal Processing.*

Fryzlewicz, P., Delouille, V. and Nason, G.P. (2005) GOES-8 X-ray sensor variance stabilization using the multiscale data-driven Haar-Fisz transform. *Tech. Rep.* 05:06, Statistics Group, Department of Mathematics, University of Bristol, UK.

Fryzlewicz, P. and Nason, G.P. (2004) A Haar-Fisz algorithm for Poisson intensity estimation. *J. Comp. Graph. Stat.*, **13**, 621–638.

Holder, D., Raubertas, R.F., Pikounis, V.B., Svetnik, V. and Soper, K. (2001) Statistical analysis of high density oligonucleotide arrays: a SAFER approach. GeneLogic Workshop on low level analysis of Affymetrix GeneChip data, Nov. 19, Bethesda, Maryland.

Hoyle, D.C., Rattray, M., Jupp, R. and Brass, A. (2002) Making sense of microarray data distributions. *Bioinformatics*, **18**, 576–584.

Hsiao, A., Worall, D.S., Olefsky, J.M. and Subramaniam, S. (2004) Variance-modelled posterior inference of microarray data: detecting gene-expression changes in 3T3-L1 adipocytes. *Bioinformatics*, **20**, 3108–3127.

Huber, W., Von Heydebreck, A., Sultmann, H., Poustka, A. and Vingron, M. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**, S96-S104.

Huber, W., Von Heydebreck, A., Sultmann, H., Poustka, A. and Vingron, M. (2003) Parameter estimation for the calibration and variance stabilization of microarray data *Statist. App. Gen. Mol. Biol.*, **2**, Issue 1, Article 3.

Johnstone, I.M. and Silverman, B.W. (2005) EbayesThresh: R programs for empirical Bayes thresholding, *J. Statist. Soft.*, **12**, 1–38.

McCaffrey, R.L., Fawcett, P., O'Riordan, M. Lee, K., Havell, E.A. Brown, P.O. and Portnoy, D.A. (2004) A specific gene expression program triggered by Gram-positive bacteria in the cytocol. *Proc. Nat. Acad. Sci.*, **101**, 11386–11391.

Munson, P. (2001) A "consistency" test for determining the significance of gene expression changes on replicate samples and two-convenient variance-stabilizing transformations. GeneLogic Workshop on Low Level Analysis of Affymetrix GeneChip Data, Nov. 19, Bethesda, Maryland.

Pauli, F., Liu, Y., Kim, A.Y., Chen, P. and Kim, S.K. (2006) Chromosomal clustering and GATA transcriptional regulation of intestine-expressed genes in C. elegans. *Development*, **133**, 287–295

Rocke, D.M. and Durbin, B.P. (2001) A model for measurement error for gene expression arrays. *J. Comp. Biol.*, **8**, 557–569.

Rocke, D.M. and Durbin, B.P. (2003) Approximate variance-stabilizing transformations for gene expression microarray data. *Bioinformatics*, **19**, 966–972.

Sebastiani, P. and Ramoni, M. (2003) Statistical Challenges in Functional Genomics. *Statist. Sci*, **18**, 33–70.

Smyth, G.K., Yang, Y.H. and Speed, T. (2003) Statistical issues in cDNA Microarray data analysis. In Brownstein, M.J. and Khodursky, A. (eds), *Functional Genomics: Methods and Protocols*, *Methods of Molecular Biology*, **224**, 111–136. Humana Press: Totowa, NJ.

Tukey, J.W. (1977) *Exploratory data analysis*, Addison-Wesley, Reading, MA.

Tusher, V., Tibshirani,R. and Chu, G. (2001) Significance analysis of microarrays applied to ionizing radiation response. *Proc. Nat. Acad. Sci.*, **98**, 5116–5121.

Velculescu, V.E., Zhang, L., Vogelstein, B. and Kinzler, K.W. (1995) Serial Analysis of Gene Expression, *Science*, **270**, 484–487.

Wang, S. and Ethier, S. (2004) A generalized likelihood ratio test to identify differentially expressed genes from microarray data, *Bioinformatics*, **20**, 100–104.