Hindawi Publishing Corporation
EURASIP Journal on Bioinformatics and Systems Biology
Volume 2006, Article ID 43056, Pages 1–13
DOI 10.1155/BSB/2006/43056

Normalization Benefits Microarray-Based Classification

Jianping Hua,[1] Yoganand Balagurunathan,[1] Yidong Chen,[2] James Lowey,[1] Michael L. Bittner,[1] Zixiang Xiong,[3] Edward Suh,[1] and Edward R. Dougherty[1,3]

[1] *Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004, USA*
[2] *Genetics Branch, Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda, MD 20892-2152, USA*
[3] *Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843, USA*

Received 11 December 2005; Revised 19 April 2006; Accepted 18 May 2006

Recommended for Publication by Paola Sebastiani

When using cDNA microarrays, normalization to correct labeling bias is a common preliminary step before further data analysis is applied, its objective being to reduce the variation between arrays. To date, assessment of the effectiveness of normalization has mainly been confined to the ability to detect differentially expressed genes. Since a major use of microarrays is the expression-based phenotype classification, it is important to evaluate microarray normalization procedures relative to classification. Using a model-based approach, we model the systemic-error process to generate synthetic gene-expression values with known ground truth. These synthetic expression values are subjected to typical normalization methods and passed through a set of classification rules, the objective being to carry out a systematic study of the effect of normalization on classification. Three normalization methods are considered: offset, linear regression, and Lowess regression. Seven classification rules are considered: 3-nearest neighbor, linear support vector machine, linear discriminant analysis, regular histogram, Gaussian kernel, perceptron, and multiple perceptron with majority voting. The results of the first three are presented in the paper, with the full results being given on a complementary website. The conclusion from the different experiment models considered in the study is that normalization can have a significant benefit for classification under difficult experimental conditions, with linear and Lowess regression slightly outperforming the offset method.

1. INTRODUCTION

Microarray technologies are widely used for assessing expression profiles, DNA copy number alteration, and other profiling tasks with thousands of genes simultaneously probed in a single experiment. Beside variation due to random effects, such as biochemical and scanner noise, simultaneous measurement of mRNA expression levels via cDNA microarrays involves variation owing to system sources, including labelling bias, imperfections due to spot extraction, and cross hybridization. Given the development of good extraction algorithms and the use of control probes at the array printing stage to aid in accounting for cross hybridization, we are primarily left with labelling bias via the fluors used to tag the two channels as the systemic error with which we are concerned. Although different experimental designs target different profiling objectives, be it global cancer tissue profiling or a single induction experiment with one gene perturbed, normalization to correct labelling bias is a common preliminary step before further statistical or computational analysis is applied, its objective being to reduce the variation between arrays [1, 2]. Normalization is usually implemented for an individual array and is then called *intra-array* normalization, which is what we consider here. Assessment of the effectiveness of normalization has mainly been confined to the ability to detect differentially expressed genes.

A major use of microarrays is phenotype classification via expression-based classifiers. Since some systematic errors may have minimal impact on classification accuracy, where only changes between two groups, rather than absolute values, are important, one might conjecture that normalization procedures do not benefit classification accuracy. This would not be paradoxical because it is well known in image processing that filtering an image prior to classification can result in increased classification error, especially in the case of textures, where fine details beneficial to classification can be lost in the filtering process. Thus, it is necessary to evaluate microarray normalization procedures relative to classification.

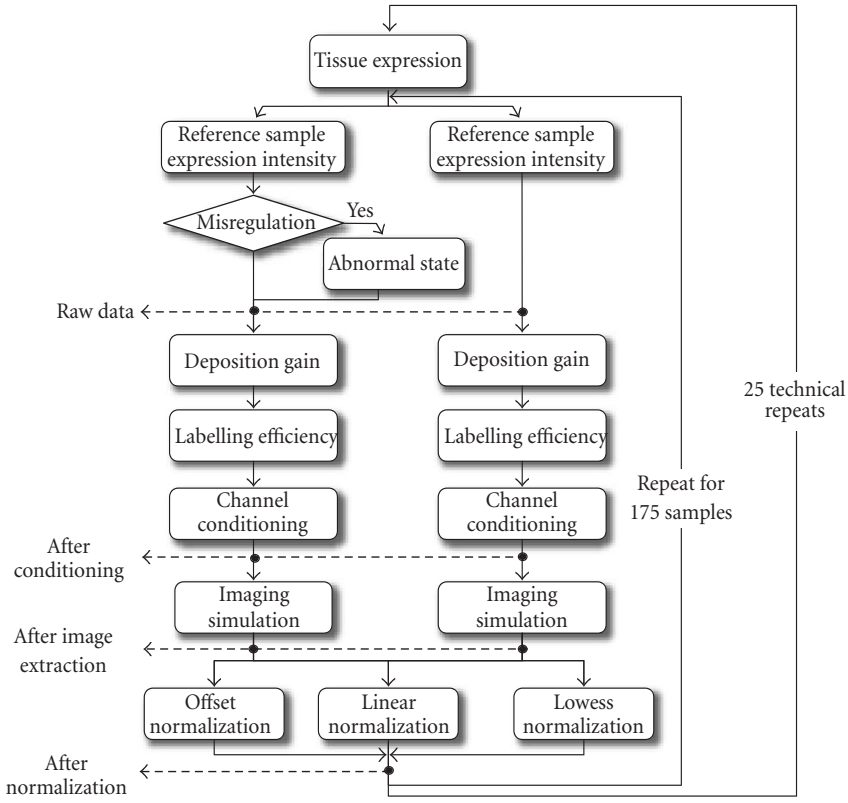


Figure 1: Simulation flow chart.

Using a model-based approach, we model the systemic-error process to generate synthetic gene-expression values. A model-based approach is employed because it gives us ground truth for the differentially expressed genes, the systemic-error process, and the evaluation of classifier error. Once generated, the synthetic expression values are subjected to typical normalization methods and passed through a set of classification rules, the objective being to carry out a systematic study of the effect of normalization on classification. Classification errors are computed at different stages of the processing so as to quantify the influence of each processing stage on the downstream analysis. As illustrated in Figure 1 by the pointers, for each classification rule, we measure accuracy at various stages of the system: (a) on the raw intensities; (b) on the conditioned intensities; (c) on the conditioned intensities following an imaging simulation; and (d) on three normalizations of the data, which can be considered as providing the practical measure of the normalization schemes. By conditioned intensities we mean the raw intensities subject to dye-scanner effects. Fluorescent dyes for microarray experiments can show nonlinear response characteristics, and different dyes give different responses, due to mismatches of fluorescent excitation strength and scanner dynamic range. These dye-scanner effects need to be simulated and, as we will see, they affect the impact of normalization.

2. MODEL GENERATION

Following the model proposed in [3], the gene-expression intensity, v_{ij}, for the ith gene in the jth sample is given by

$$v_{ij} = (r_{ij}\rho)^{m_i} d_i l_j u_{ij} + n_{ij}, \tag{1}$$

where u_{ij} is the reference intensity for each cell system, l_j is the labelling and hybridization efficiency, d_i is the printing deposition gain, ρ is a constant representing fold change (for any misregulated gene), r_{ij} is the variation of the fold change, n_{ij} is additive noise due to fluorescent background, and m_i takes the value 1 (up-regulated), 0 (normal), or -1 (down-regulated) for the gene i. The expression intensity given in (1) will be further subject to a scan-conditioning effect for both fluorescent dyes and other imaging simulations, as illustrated in Figure 1.

Prior to describing the parameters in the following subsections, we would like to comment on our approach to model development. The parameters for the simulation have been drawn from our experience at the National Institutes of Health with thousands of good and bad cDNA chips. The parameters chosen represent behaviors in the chips found to be worth analyzing. We have modeled variance sources, and their dependent and independent interactions, in a realistic way. In this paper, we also testunder different overall levels of

severity, again empirically derived from data from our own lab and many other labs that produce printed chips and have shared data with us. The behavior on poor chips would certainly lie outside the boundaries chosen; however, we believe that with such poor quality chips, one would not be able to reliably analyze the data, so we would not accept them. The noise levels and interactions seen in these simulations are worse than those that one gets with the best currently available technologies, but are representative of what one would typically face with reasonable to good quality homemade chips. The simulation presents the types and levels of problems one faces in real data from cDNA microarrays. The choice of most model parameters is discussed in the following sections, while the appendix discusses several parameters which are too complicated to be addressed in the main text. The data set (50 prostate cancer samples) used to estimate the parameters is provided on the complementary website.

2.1. Probe intensity simulation

In the basic model of [3], there are N genes, $g_1, g_2, \ldots, g_N$, in the model array. In the reference state, which we assume to be the normal state, the expression-intensity mean of the genes is distributed according to an exponential distribution with mean β, the amount of the shift representing the minimal detectable expression level above background noise. Hence, there are N mean expression levels $I_1, I_2, \ldots, I_N$ with $I_i \sim \text{Exp}[\beta]$. In many practical microarray experiments, there exist some higher-intensity probes and some extremely low-intensity probes due to various probe design artifacts. To simulate this effect, we mix some random intensities derived from a uniform distribution. This is done by choosing a probability q_0 and defining

$$I_i \sim \begin{cases} \text{Exp}[\beta], & \text{with probability } q_0, \\ U[0, A_{\max}], & \text{with probability } 1 - q_0. \end{cases} \tag{2}$$

For our simulations, β has been estimated from a set of microarray experiments, and the parameters are set at $\beta = 3000$, $A_{\max} = 65535$, and $q_0 = 0.9$. The intensity u_{ij} of the gene g_i in the jth sample, for the reference state, is drawn from a normal distribution with mean I_i and standard deviation αI_i, where α is a model parameter controlling signal variability,

$$u_{ij} \sim \text{Normal}(I_i, \alpha I_i), \tag{3}$$

I_i represents the true gene-expression level drawn according to (2) and α is the coefficient of variation of the cell system, varying from 5% to 15% (self-self experiment). The sample index j is not on the right-hand side of (3) because the normal expression state does not change. The simulation is randomly seeded at the start of each technical repeat and remains fixed throughout that repeat.

2.2. Intensity simulation for reference and test states

For an abnormal state (e.g., cancer state), a nominal (mean) fold change ρ is assumed for the model. The actual fold change for the gene i on the jth array is $r_{ij}\rho$, where r_{ij} is drawn from a beta distribution over the interval $[1/p, p]$ with mean 1, so that

$$r_{ij} \sim \text{beta}_{[1/p,p]}(2, 2p), \tag{4}$$

where $1 \le p \le \rho$. When the model parameter $p = 1$, there is no variation in the fold change, so that it is fixed at ρ; when $p = \rho$, the fold change lies between 1 and ρ^2. As suggested in [4], we set $\rho = 1.5$, as this is a level of fold change that can be reliably detected, while making the task of classification neither too easy nor too difficult under practical choices for the other model parameters. Misregulated genes, defined by $+1$ (up-regulated) and -1 (down-regulated) in m_i, are randomly selected at the beginning of each technical repeat, and fixed for all samples in the repeat.

2.3. Array printing and hybridization simulation

cDNA deposition results in a gain (or loss) in measured expression intensity. The signal gain is related to each immobilized detector and therefore each observation, independent of the sample. It is distributed according to a beta distribution,

$$d_i \sim \text{beta}_{[1/c,c]}(2, 2c). \tag{5}$$

There is also a gain/loss, l_j, of expression level owing to the RNA labelling and hybridization protocol. Related to each RNA, l_j is a constant scale factor for all genes for a given channel of an array, and is distributed according to

$$l_j \sim \text{beta}_{[1/h,h]}(2, 2h). \tag{6}$$

Then the final gene-expression intensity is generated by adding the background noise n_{ij}. The value of n_{ij} is drawn from a normal distribution with mean I_{bg} and standard deviation $\alpha_{bg} I_{bg}$, which are fixed through out each technical repeat.

2.4. Channel conditioning

Having completed the expression intensity generation, for a sample j with N genes, for the *normal* and the *abnormal* classes we have two channel intensities: $R_j = \{v_{1j}, \ldots, v_{Nj}\}$ and $G_j = \{v'_{1j}, \ldots, v'_{Nj}\}$, respectively. Given the intensities, dye-scanner effects need to be simulated. We model this effect by a nonlinear detection-system-response characteristic function,

$$f(x) = a_0 + x^{a_3}\left(1 - e^{-x/a_1}\right)^{a_2}; \tag{7}$$

R and G are transformed by this function, according to $f_R(x)$ and $f_G(x)$, to obtain the realistic fluorescent intensities. The resulting observed fluorescent intensities, $R'_j = f_R(R_j)$ and $G'_j = f_G(G_j)$, are the simulated mean intensities of the jth sample for all N genes.

Common effects are modeled by appropriate choice of the parameters in (7). Turning tails are modeled by $(a_0, a_1, a_2, a_3) = (0, a_1, -1, 1)$ for one channel, where the intensity will maintain a constant of a_1 at the lower-tail end,

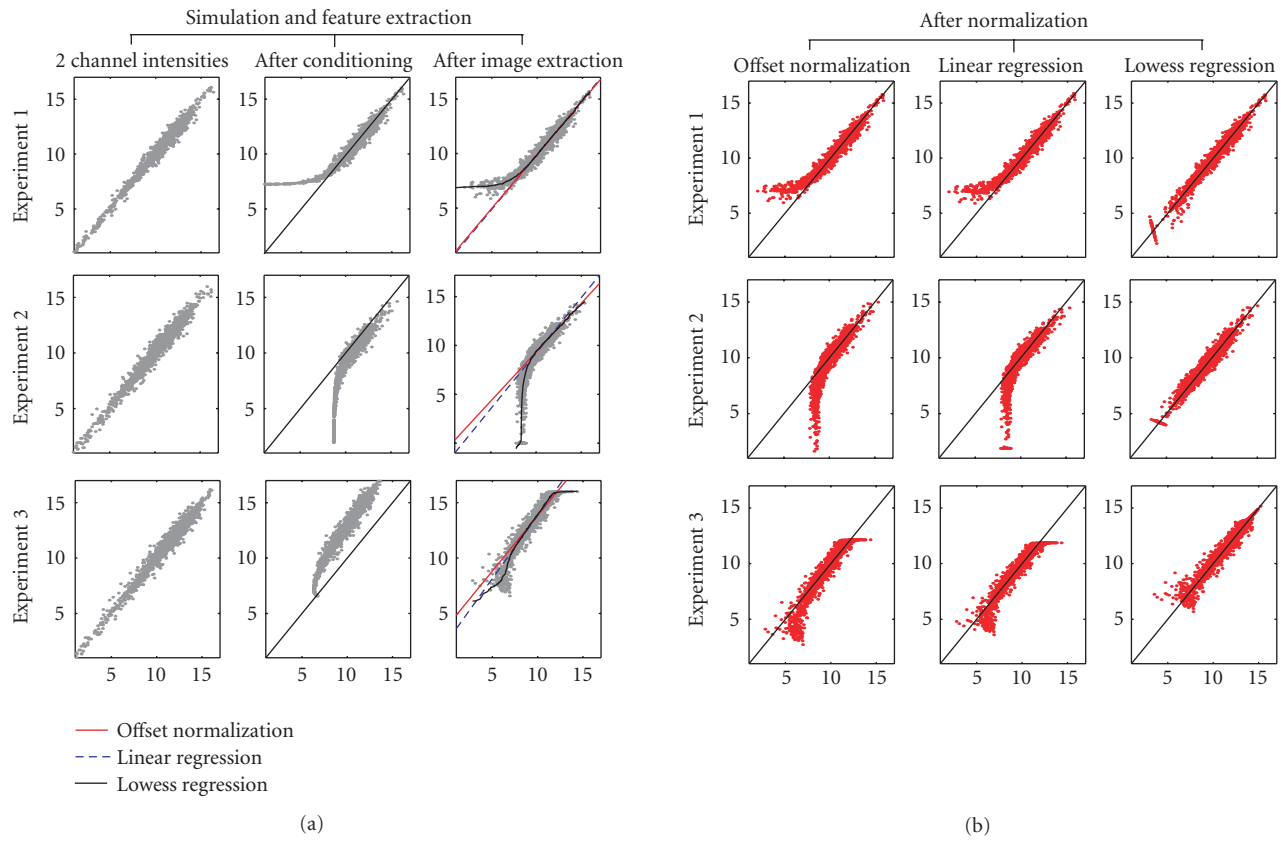


Figure 2: Scatter plots showing the effects of normalization.

as shown in Figure 2. Rotation of the normalization line is achieved by using an a_3 value other than 1.0. Setting the conditioning function parameters to $(0, 1, -1, 1)$ reduces transform function to $f(x) \approx x$, for $x \gg 1$, or no transforming effect at all.

Channel-conditioning functions are applied to each detection channel in two ways.

Method 1. Generate uniformly random parameters between the ideal setting $(0, 1, -1, 1)$ and a specific alternative setting.

Method 2. There is 0.5 probability that a given parameter setting will be used and a 0.5 probability that Method 1 will be used.

2.5. Microarray spot imaging simulation and data extraction

Upon obtaining each gene's intensity, a 1D Gaussian spot shape of size 100 with mean of given intensity is generated, and background noise is also added. To further differentiate the two-color system, we introduce a multiplicative *dot gain* parameter for each Gaussian shape, to enforce possible fluorescent dye bias. All pixels with intensity higher than $A_{\max} = 65\,535$ are set to $A_{\max}$ to simulate the effect of saturation. Measured expression intensity is calculated by averaging all pixel values, since we only simulate the target area. We subtract the mean background from the measured expression intensity and then report it. The measurement quality is calculated using the signal-to-noise ratio according to the definition given in [4].

2.6. Simulation conditions

Each experiment has 2000 genes per array and 175 samples per data set, with 87 normal samples and 88 abnormal samples. Of the 2000 genes, 200 (10%) are differentially expressed. These 200 genes are randomly selected at the beginning of each run and then fixed for all 175 samples (actually only 88 of them use differentially expressed genes). They are the true markers for classification. We then select another 100 (5%) genes randomly for each sample as differentially expressed genes, whereby it is the task of classifier training to (hopefully) eliminate these genes. In sum, for each sample, we have 15% differentially expressed genes, with 10% at fixed locations for all 175 samples, and 5% at random positions for each sample. Array spot size is preset to 100 pixels (1D only).

For each simulation condition, 25 technical repeats are generated, with different random parameters reinitialized. All other parameters are listed in Table 1. Simulation parameters for each experiment have been selected according to laboratory experience.

Table 1: Simulation parameters for each experimental condition.

Parameters	Experiment 1	Experiment 2	Experiment 3
Expression intensity mean, β	3000	3000	5000
Expression intensity coefficient of variation, α	0.10	0.15	0.15
Deposition gain, c	1.1	2	2
Labelling efficiency, h	1.1	4	4
Fold change, ρ	1.5	1.5	1.5
Fold change variation, p	1.5	1.5	1.5
Background noise mean, I_{bg}	100	400	400
Background noise coefficient of variation, α_{bg}	0.10	0.15	0.15

Experiment 1. It simulates a well-controlled lab protocol (small labelling efficiency variation, small expression variation, and background noise), along with high-quality arrays (very small deposition gain variation), and equal print dot gain. Channel conditioning parameters are selected consistently and relatively low: red channel, $(a_0, a_1, a_2, a_3) = (0, 1, -1, 1)$; green channel, $(a_0, a_1, a_2, a_3) = (0, 500, -1, 1)$. Channel-conditioning functions are applied to each channel according to Method 1; however, by setting the channel conditioning parameters identical to the ideal setting, there is no randomization in the channel-conditioning function of the red channel, and hence only the green channel changes randomly.

Experiment 2. It simulates a much less-controlled lab protocol (large labelling efficiency variation between the two channels, large expression variation, and large background noise), lower quality arrays (higher deposition gain variation), and equal print dot gain. Channel conditioning parameters are larger for both channels, so there is greater possibility of having nonlinear characteristics for each hybridization result: red channel, $(a_0, a_1, a_2, a_3) = (0, 500, -1, 1)$; green channel, $(a_0, a_1, a_2, a_3) = (0, 500, -1, 1)$. Channel-conditioning functions are applied to each channel according to Method 1, so both channels are allowed to be randomly selected. This setup creates conditionings that contain no turning tails (similar conditioning setting) and tails turning in either direction (one near the ideal setting and the other near the given setting).

Experiment 3. It is a similar simulation to Experiment 2, but with higher expression intensity (mean of 5000, instead of 3000) and uneven print dot gain (2× for green channel), so that greater saturation effect is observed. Different linear rotation parameters are used in the channel conditioning function, resulting in a more linear, rather than nonlinear, rotated effect (less dependency for Lowess normalization). For the red channel, $(a_0, a_1, a_2, a_3) = (0, 100, -1, 9)$; for the green channel $(a_0, a_1, a_2, a_3) = (0, 100, -1, 1.1)$. Channel-conditioning functions are applied to each channel according to Method 2, which requires 50% chance of one specific parameter setting (tail-turning and rotating scatter plot) to be used such that some extreme conditions will be reached with small sampling rate, while preserving some randomness of the direction and the degree of tail-turning and rotation.

There are several rationales behind the three simulated cases: dye-flipping commonly observed as tail-turning in different directions, various regression curve rotations due to uneven dynamic range of fluorescent signal on account of labelling efficiency or RNA loading, and, of course, various background effects and noise level.

3. NORMALIZATION PROCEDURES

In this study, we have implemented three normalization procedures: the offset method, linear regression, and the Lowess method. It is typically assumed that normalization methods are applied under the condition that most genes are not differentially expressed [5]. This assumption is fulfilled by our simulation setup. The effects of three normalization procedures on all three experiments are illustrated in Figure 2.

3.1. Offset normalization

The simplest and most commonly used normalization is the offset method [6]. To describe it, let the red and green channel intensities of the kth gene be r_k and g_k, respectively. In many cases these are background-subtracted intensities. In an ideal case where two identical biological samples are labeled and cohybridized to the array, we expect the log-transformed ratios, and therefore the sum of the log-transformed ratios, to be 0; however, due to various reasons (dye efficiency, scanner PMT control, etc.), this assumption may not be true. If we assume that the two channels are equivalent, except for a signal amplification factor, then the ratio of the kth gene, t_k, can be calculated by

$$\log t_k = \log\left(\frac{r_k}{g_k}\right) - \frac{1}{N_q}\sum_{i=1}^{N_q}\log\left(\frac{r_i}{g_i}\right), \tag{8}$$

where the second term in is a constant offset that simply shifts the r_k versus g_k scatter plot to a 45° diagonal line intersecting the origin and N_q is the number of probes that have measurement quality score of 1.0.

3.2. Linear regression

In some cases the R-G scatter plot may not be perfectly at a 45° diagonal line (or flat line for an A-M plot) due to the difference when the scanner's two channels may operate at

different linear characteristic regions. In this case, full linear regression, instead of requiring the line to intersect at the origin, may be necessary. In this study, the coefficients of a first-degree polynomial equation are obtained in via least-squares minimization, namely, minimizing

$$E\Big[(g_k - y_k)^2\Big] = E\Big[(g_k - (ar_k + b))^2\Big], \tag{9}$$

where a and b are the two coefficients of the first-degree polynomial. For expectation calculation, we only use intensity data that have measurement quality score of 1.0.

3.3. Lowess regression

Some microarray expression levels may have large dynamic range that will cause scanner systematic deviations such as nonlinear response at lower intensity range and saturation at higher intensity. Although data falling into these ranges are commonly discarded for further analysis, the transition range, without proper handling, may still cause some significant error in differential expression gene detection. To account for this deviation, locally weighted linear regression (*Lowess*) is regularly employed as a normalization method for such intensity-dependent effects [5, 6]:

$$\hat{y} = \text{Lowess}\,(X, Y), \tag{10}$$

where the components of X and Y are

$$x_k = \frac{\log_2 r_k + \log_2 g_k}{2}, \qquad y_k = (\log_2 r_k - \log_2 g_k) \tag{11}$$

and $\hat{y}$ is the regression center at each sample. The normalized ratio is

$$t'_k = y_k - \hat{y}_k \tag{12}$$

and the normalized channel intensities are

$$r'_k = 2^{x_k + (t'_k/2)}, \qquad g'_k = 2^{x_k - (t'_k/2)}. \tag{13}$$

In this study, we utilize Matlab's native implementation of Lowess.

4. EXPERIMENTAL DESIGN

Seven classifiers are considered in this study: 3-nearest-neighbor (3NN) [7], Gaussian kernel [7], linear support vector machine (linear SVM) [8], perceptron [9], regular histogram [7], classification and regression trees (CART) [10], linear discriminant analysis (LDA) [10], and multiple-perceptron majority-voting classifier. For linear SVM, we use the codes from LIBSVM 2.4 [11] with suggested default settings. For the Gaussian kernel, the smoothing factor h is set to 0.2. For the regular histogram classifier, the cell number along each dimension is set to 2. For CART, the Gini impurity criterion is used. To improve the performance and prevent overfitting, the tree is not fully grown, and the splitting stops when there are six samples or fewer in a node, without further pruning. For the perceptron, the learning rate is set to 0.1, and the algorithm stops once convergence is achieved or a maximum iteration time of 100 is reached. The same settings are used for the multiple-perceptron majority-vote classifier. All classifiers use the log-ratio of expression levels for classification. Results for three of the classifiers, 3NN, linear SVM, and LDA, are presented in the paper and results for the others are given on the complementary website.

The combination of various situations listed in the previous sections results in a significant number of different conditions to be considered. Altogether we have 3 conditioning functions, with each function generating $M = 25$ experiment repeats. In each experiment, six ratios are used: true value, conditioned value, direct ratio, offset normalization, linear regression, and Lowess regression. True values are the ratios between $G = \{v_{1j}, \ldots, v_{Nj}\}$ and $R = \{v'_{1j}, \ldots, v'_{Nj}\}$, which are the ground truths of expression levels. The conditioned values are the ratios between conditioned expression levels R'_k and G'_k. Direct ratios are the ratios using the channel values following imaging simulation and before normalization. Offset normalization, linear regression, and Lowess regression are the ratios obtained by the respective normalization methods. Hence we have altogether 450 sets of data, each set containing 175 samples, with each sample consisting of 2000 gene-expression ratios.

Each classification rule is independently applied to each of the 450 data sets and we estimate the corresponding classification error using cross-validation, which is applied in a nested fashion by holding out some samples, applying feature selection to arrive at a feature set, classifier, and error, and then repeating the process in loop. Specifically, we have the following.

(1) Given a data set, to estimate performance at training sample size n, each time n samples are randomly drawn from the 175 samples in the data set. Since the observations are drawn without replacement, they are actually not independent, and therefore a large training sample size would induce inaccuracy in the error estimation (see [12] for a discussion of this issue in the context of microarray data). Hence, we set $n = 30$ in our study to reduce the impact of observation correlation.

(2) After eliminating any gene with quality score below 0.3 in any of the n samples, feature selection is conducted on the n samples composed from the remaining genes. Optimal feature sets of size 1 to 20 are obtained, except for the regular histogram classifier, which is from 1 to 10, owing to the exponential increase in the cell numbers with feature size. Three feature-selection schemes are used.

(a) Sequential floating forward selection (SFFS) [13] with leave-one-out (LOO) error estimation is used to find the optimal feature subsets at various sizes based on the n samples. Studies have shown the superiority of SFFS for feature selection [14, 15].

(b) SFFS is used with bolstered resubstitution error estimation [16] instead of LOO error estimation within the SFFS algorithm. A previous study has demonstrated better performance using bolstering within the SFFS algorithm [17].

(c) The third scheme uses random selection from the 200 true markers (10% differentially expressed genes at fixed locations). Since we know all the true markers in the 2000 available genes, we can randomly pick genes without replacement from the true markers using the same feature set sizes. Obviously this is not a practical scheme, but one for comparison only.

(3) For every optimal feature subset obtained in the previous step, construct the corresponding classifier and test it on the remaining $175 - n$ samples.

(4) Repeat the steps (1) through (3) a total of 250 times, and average the obtained error rates and true markers found. There are three error curves for the three feature selection schemes, respectively, and there are two curves showing the numbers of true markers found by the two SFFS-based feature selection algorithms, respectively.

Lastly, the results of the 450 data sets with the same conditioning function and ratio type are averaged.

5. CLASSIFICATION RESULTS

Selected classification results for Experiments 1, 2, and 3 are presented in Figures 3, 4, and 5, respectively, for 3NN, linear SVM, and LDA, with the full classification results being given on the complementary website www.tgen.org/research/index.cfm?pageid=644. The figures in the paper provide error curves relative to the number of features for SFFS using leave-one-out and SFFS using bolstered resubstitution. Although our concern in this paper is with comparative performance among the normalization methods, we begin with a few comments regarding general trends.

As expected from a previous study, SFFS with bolstering significantly outperforms SFFS with leave-one-out [13]. In accordance with a different study, owing to uncorrelated features and the Gaussian-like nature of the label distributions, LDA, 3NN, and linear SVM do not peak early if features are selected properly, even for sample sizes as low as 30 [18]. Hence, we see no peaking for feature size $d \leq 20$ for SFFS with bolstering; however, we do see very early peaking for LDA when using SFFS with leave-one-out, owing to poor feature selection on account of leave-one-out. This is in accord with the early study that shows linear SVM and 3NN less prone to peaking than LDA with uncorrelated features [18]. This proneness to peaking for LDA is also visible when the true markers are selected randomly, which is akin to using equivalent features when the results are averaged over a large number of cases. In particular, we see that for the true values, peaking with normalization is around $d = 14$, which is in agreement with a previous study that predicts peaking at $n/2 - 1$ for equivalent features [19]. Finally, in regard to peaking, on the complementary website we see early peaking for the regular-histogram rule, a rule whose use is certainly not advisable in this context.

Focusing now on the main issue, the effect of normalization, we see a general trend across the classifiers: in the case of the easy one (Experiment 1), there is very slight improvement using normalization, the particular normalization used not being consequential; and for the difficult ones (Experiments 2 and 3), there is major improvement using normalization, with linear and Lowess regression being slightly better than offset normalization, but not substantially so. As expected, in all cases, the true values give the best results. The actual quantitative results we have obtained depend on the various parametric settings of the classifiers. Certainly some changes would occur with different selections. Owing to the consistency of the results across all classifiers studied, we believe the general trends will hold up for corresponding parametric choices; of course, one might find the parametric settings that give different results, but such settings would only be meaningful were they to result in synthetic data similar to that experienced in practice.

6. CONCLUSION

The standard normalization methods, offset normalization, linear regression, and Lowess regression, have been shown to be beneficial for classification for the conditions and classifiers considered in this study. Their benefit depends on the degree of conditioning and the randomness within the data, which is in agreement with intuition. While linear and Lowess regressions have performed slightly better than simple offset normalization in the cases studied, the improvement has not been consequential.

APPENDIX

A. PARAMETER ESTIMATION

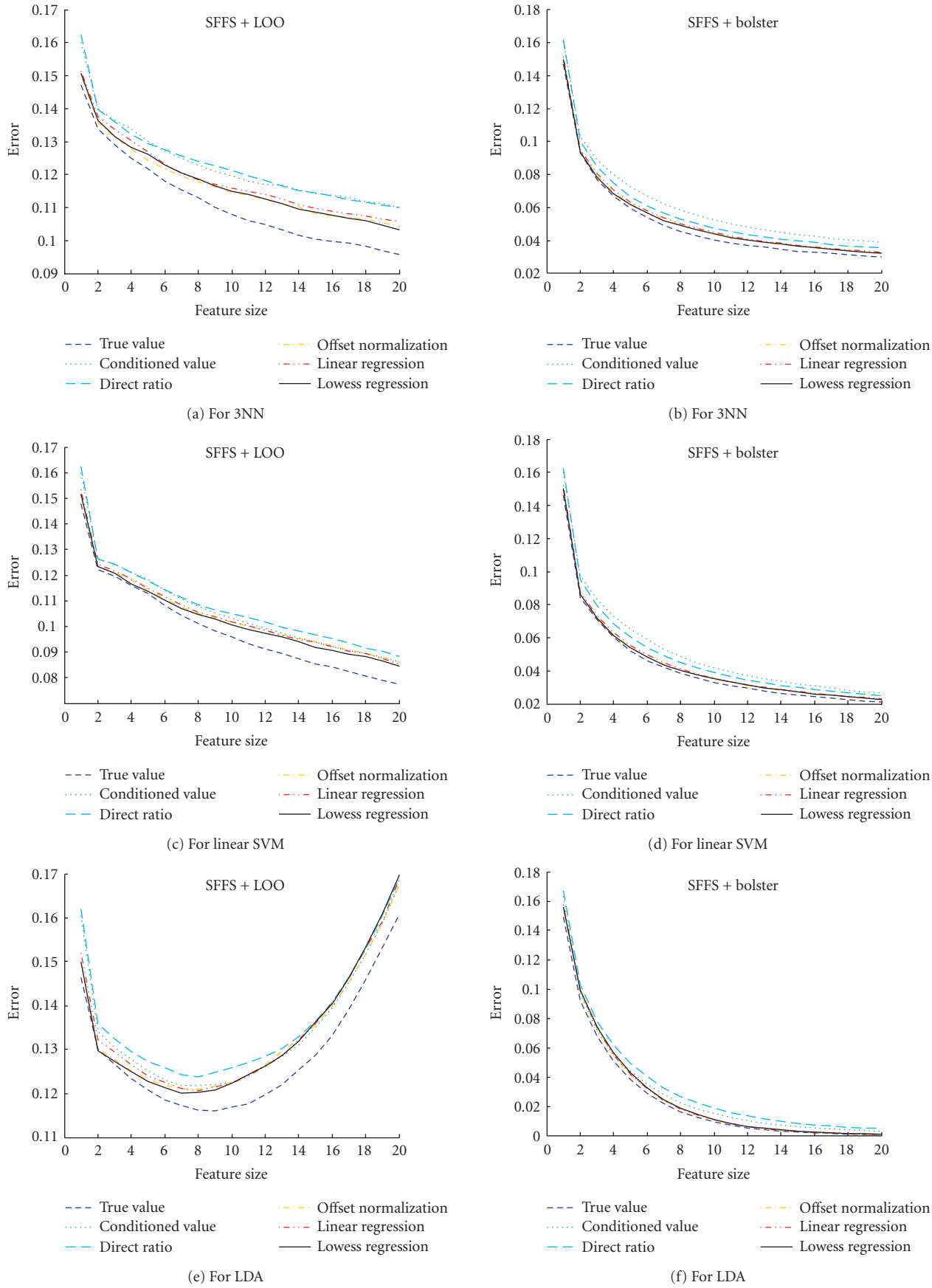
The appendix discusses estimation of several important parameters employed in the simulation model. The data set used to estimate the parameters is provided in the complementary website. It results from 50 prostate cancer samples whose gene-expression profiles have been obtained using cDNA microarrays (custom-manufactured by Agilent Technologies, Palo Alto, Calif). In particular, the parameter for the exponential distribution of (2) is estimated using the prostate cancer data set. Using only the Cy5 channel intensity data, β was spread from 1826 to 5023.

The coefficient of variation α of each microarray can be found by using a set of housekeeping genes that carry minimal biological variation between samples, or a set of duplicated spots on the same microarray, which has only assay variation plus spot-to-spot variation (or printing artifacts). The latter method typically produces a smaller α than that from housekeeping gene set, but it may not be available on every array. The calculation for α is given as follows.

(1) For a given set of housekeeping (HK) genes

   (a) get all normalized expression ratios t_i, for HK genes;
   (b) calculate α by [20]

$$\alpha = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{(t_i - 1)^2}{(t_i^2 + 1)}}. \tag{A.1}$$



(a) For 3NN

(b) For 3NN

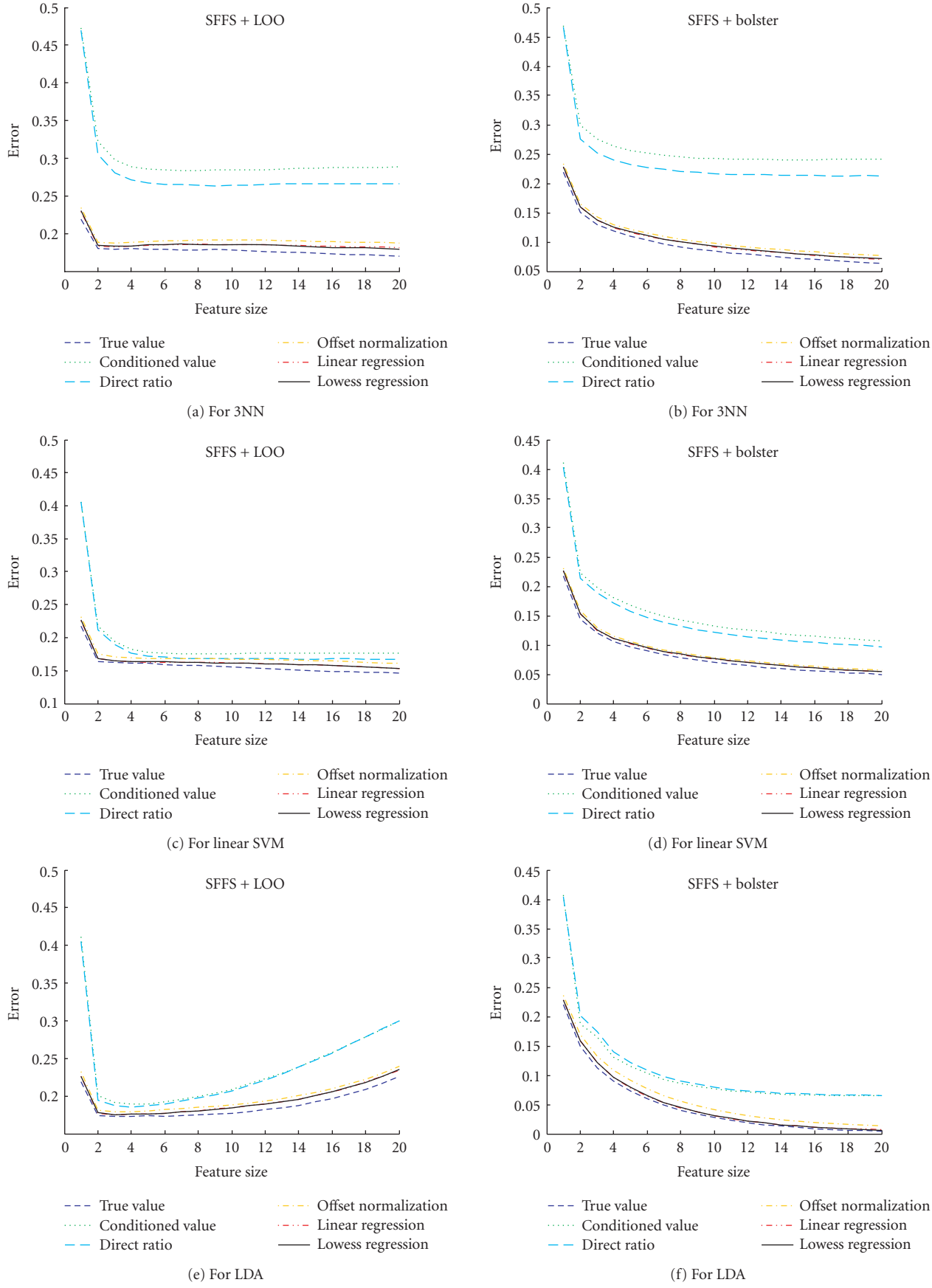
(c) For linear SVM

(d) For linear SVM

(e) For LDA

(f) For LDA

Figure 3: Classification results for Experiment 1.



(a) For 3NN

(b) For 3NN

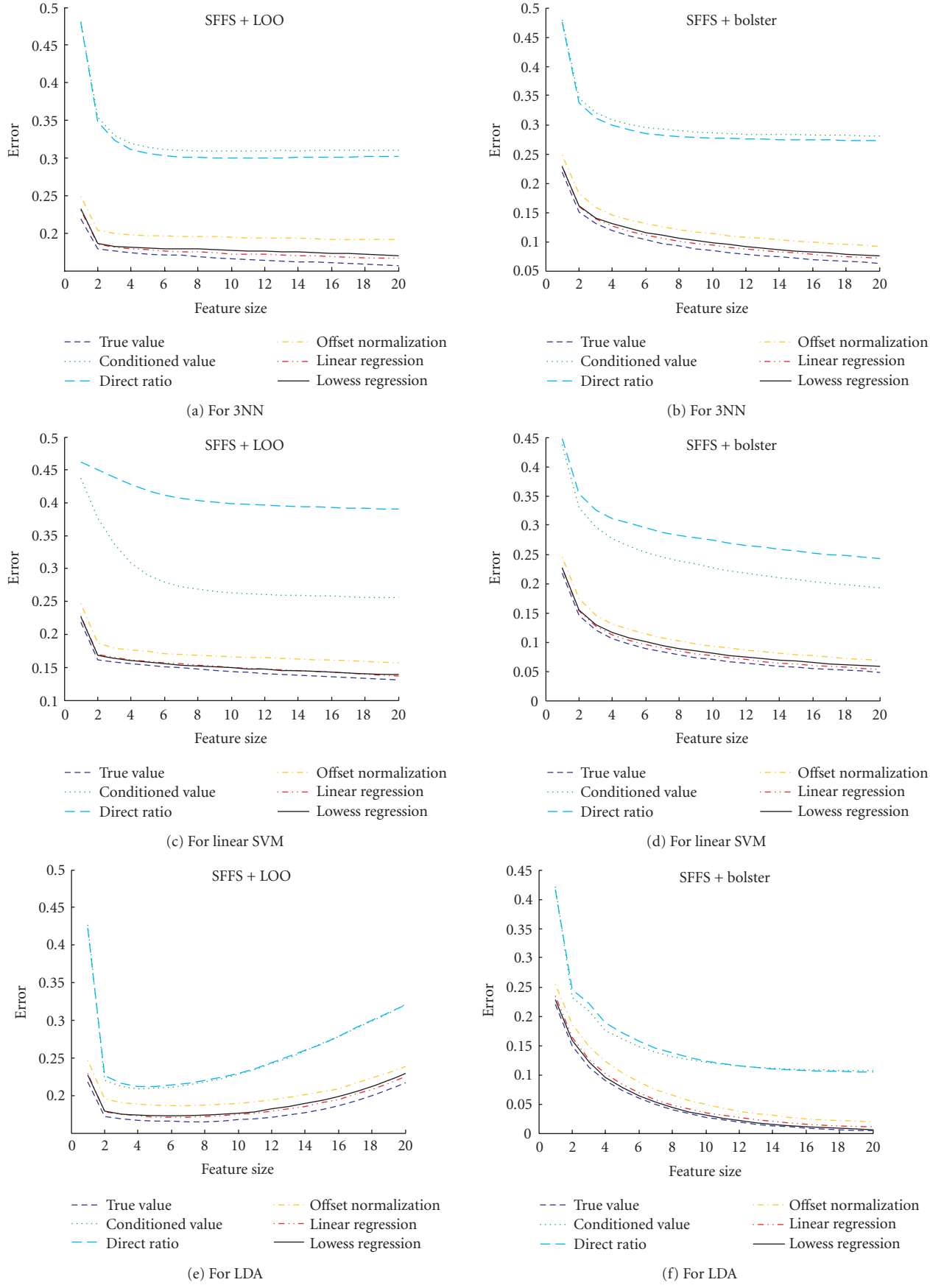
(c) For linear SVM

(d) For linear SVM

(e) For LDA

(f) For LDA

Figure 4: Classification results for Experiment 2.



(a) For 3NN

(b) For 3NN

(c) For linear SVM

(d) For linear SVM

(e) For LDA

(f) For LDA

Figure 5: Classification results for Experiment 3.

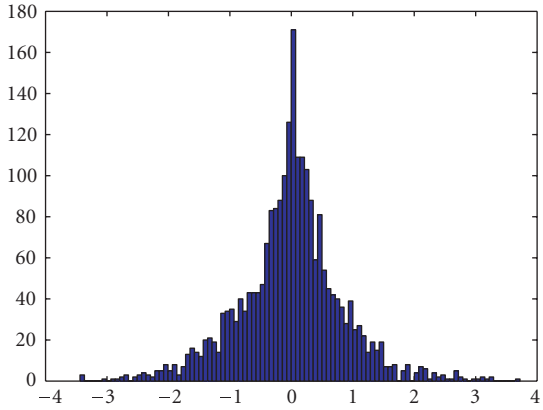


Figure 6: The histogram (binned in log-scale) of deposition gain (log-ratio greater than 0) or loss (log-ratio less than 0).

(2) For a given set of replicated genes (replicated K times for each gene)

   (a) get all normalized expression ratios t_{ij} for all duplicate locations, $j = 1, \ldots, K$, for gene i;
   (b) calculate ratio of ratios $t_{ij} = t_{ij}/t_{i1}$, for $j = 2, \ldots, K$;
   (c) calculate α by

$$\alpha = \sqrt{\frac{1}{2n(K-1)} \sum_{j=2}^{K} \sum_{i=1}^{n} \frac{(t_{ij}-1)^2}{(t_{ij}^2+1)}}. \tag{A.2}$$

Also, α can be estimated from a self-self (homotypic) experiment. It is normally around 0.05 to 0.15. To justify this observation, we have selected the same 50 aforementioned arrays. For these, the α of each experiment estimated from duplicate spots was spread from 0.067 to 0.073.

The deposition gain c is estimated according to the following procedures.

(1) Normalize each reference channel of N experiments by mean intensity within each microarray.
(2) Calculate mean intensity of each cDNA location across N experiments as the estimate of expression level of each gene.
(3) For each gene, calculate ratios (deposition gain or loss) by dividing mean intensity of the gene obtained in step (2). Repeat for every cDNA location.
(4) Pool all deposition gain ratios from genes and positions together. The histogram of all deposition gain ratios from the 50 microarray experiments is shown in Figure 6.
(5) To avoid some inaccurate intensity measurements that may still remain in the data set after measurement quality filtering, we estimate the 1-percentile sample and 99-percentile sample from the deposition gain ratios, $\tau_{1\%}$ and $\tau_{99\%}$. Estimate the range of deposition gain c by

$$c = \max\left\{\frac{1}{\tau_{1\%}}, \tau_{99\%}\right\}. \tag{A.3}$$

For the set of 50 experiments employed, we have $c = 2.28$. It is based on this value, we set c for our experiments. The labelling gain h is determined in a similar fashion.

REFERENCES

[1] J. Quackenbush, "Microarray data normalization and transformation," *Nature Genetics*, vol. 32, no. 5 supplement, pp. 496–501, 2002.

[2] M. Bilban, L. K. Buehler, S. Head, G. Desoye, and V. Quaranta, "Normalizing DNA microarray data," *Current Issues in Molecular Biology*, vol. 4, no. 2, pp. 57–64, 2002.

[3] S. Attoor, E. R. Dougherty, Y. Chen, M. L. Bittner, and J. M. Trent, "Which is better for cDNA-microarray-based classification: ratios or direct intensities," *Bioinformatics*, vol. 20, no. 16, pp. 2513–2520, 2004.

[4] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bittner, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, no. 9, pp. 1207–1215, 2002.

[5] Y. H. Yang, S. Dudoit, P. Luu, et al., "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation," *Nucleic Acids Research*, vol. 30, no. 4, p. e15, 2002.

[6] G. C. Tseng, M.-K. Oh, L. Rohlin, J. C. Liao, and W. H. Wong, "Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects," *Nucleic Acids Research*, vol. 29, no. 12, pp. 2549–2557, 2001.

[7] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer, New York, NY, USA, 1996.

[8] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[9] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, DC, USA, 1962.

[10] R. Duda and P. Hart, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2001.

[11] C.-C Chang and C.-J. Lin, "LIBSVM: introduction and benchmarks," Tech. Rep., Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2000.

[12] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[13] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[14] A. K. Jain and D. Zongker, "Feature selection: evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[15] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33, no. 1, pp. 25–41, 2000.

[16] U. Braga-Neto and E. R. Dougherty, "Bolstered error estimation," *Pattern Recognition*, vol. 37, no. 6, pp. 1267–1281, 2004.

[17] C. Sima, S. Attoor, U. Brag-Neto, J. Lowey, E. Suh, and E. R. Dougherty, "Impact of error estimation on feature selection algorithms," *Pattern Recognition*, vol. 38, no. 12, pp. 2472–2482, 2005.

[18] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty, "Optimal number of features as a function of sample size for various classification rules," *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, 2005.

[19] A. K. Jain and W. G. Waller, "On the optimal number of features in the classification of multivariate Gaussian data," *Pattern Recognition*, vol. 10, no. 5-6, pp. 365–374, 1978.

[20] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, no. 4, pp. 364–374, 1997.

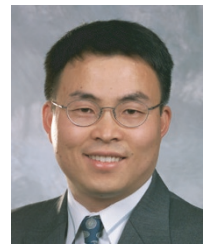
Jianping Hua received the B.S. and M.S. degrees in electrical engineering from the Tsinghua University, Beijing, China, in 1998 and 2000, respectively. He received the Ph.D. degree in electrical engineering from Texas A&M University in 2004. Currently, he is a senior postdoctoral fellow in Translational Genomics Research Institute (TGen) at Phoenix, Ariz. His main research interest lies in bioinformatics, genomic signal processing, signal and image processing, image and video coding, and statistic pattern recognition.

Yoganand Balagurunathan is a Senior Research Scientist at TGen. He received a Ph.D. degree in electrical engineering from Texas A&M University, College station, Tex, in 2001 and M.E. and B.E. degrees in electronics & communication engineering from Anna University and the University of Madras in India, respectively. He had worked in diverse research fields from imaging, soil science, and life sciences and published his findings in peer-reviewed journals. His research interests include computational biology, pattern recognition, and nonlinear image/signal processing.

Yidong Chen received his B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1983 and 1986, respectively, and Ph.D. degree in imaging science from Rochester Institute of Technology, Rochester, NY, in 1995. From 1986 to 1988, he joined the Department of Electronic Engineering of Fudan University as an Assistant Professor. During 1988 to 1989, he was a Visiting Scholar at the Department of Computer Engineering, Rochester Institute of Technology. From 1995 to 1996, He joined Hewlett Packard Company as a Research Engineer, specialized in digital halftoning and color image processing. Since 1996, he joined microarray technology development effort at the National Human Genome Research Institute, National Institutes of Health, Bethesda, Md, as a Staff Scientist for microarray image analysis and bioinformatics. Currently, he is an Associate Investigator at Cancer Genetics Branch of the National Human Genome Research Institute. His research interests include bioinformatics, biostatistics in genomic research, genetic data visualization, analysis and management, genetic network modeling, and biomedical image processing.

James Lowey is the Assistant Director of the High Performance Biocomputing Center at the Translational Genomics Research Institute (TGen). He is responsible for the architecture, management, and daily operation of TGens high performance computer systems that include a 512 node parallel cluster computer and various large SMP machines. He works closely with TGen scientists to implement and provide computational tools and data management systems to facilitate and accelerate translational genomics research. Prior to joining TGen, he worked as a Consultant at various Fortune 500 companies, implementing and managing large-scale computational systems.

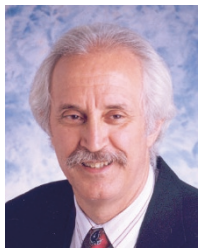
Michael L. Bittner has done research on the practical applications of biotechnology in the fields of bioproduction of proteins, peptides, and small molecules, and in development of clinical diagnostics and therapeutics. He has been a Researcher in both corporate (Monsanto, Amoco) and research institute (NIH, TGen) settings.



Zixiang Xiong received the Ph.D. degree in electrical engineering in 1996 from the University of Illinois at Urbana Champaign. From 1997 to 1999, he was with the University of Hawaii. Since 1999, he has been with the Department of Electrical and Computer Engineering at Texas A&M University, where he is an Associate Professor. He spent the summers of 1998 and 1999 at Microsoft Research, Redmond, Wash, and the summers of 2000 and 2001 at Microsoft Research in Beijing. His current research interests are network information theory and code designs, genomic signal processing and networked multimedia. He received an NSF Career Award in 1999, an ARO Young Investigator Award in 2000, and an ONR Young Investigator Award in 2001. He also received faculty fellow awards in 2001, 2002, and 2003 from Texas A&M University. He served as Associate Editor for the IEEE Trans. on Circuits and Systems for Video Technology (1999–2005) and the IEEE Trans. on Image Processing (2002–2005). He is currently an Associate Editor for the IEEE Trans. on Signal Processing and the IEEE Trans. on Systems, Man, and Cybernetics (Part B).

Edward Suh is the Chief Information Officer of the Translational Genomics Research Institute (TGen), where he leads and manages Biomedical Informatics, Information Technology, and High Performance Biocomputing Programs. He and his team develop and provide data mining and data management systems, computational algorithms and application software, and high-performance biocomputing and secure information technology infrastructure for rapid collection, integration, analysis, and dissemination of biomedical data for the discovery of novel biomarkers, diagnostics, and prognostics, leading to the treatment of diseases. He has served multiple NIH grants in the

capacity of an IT Director and an Investigator. He joined TGen after 15 years at NIH, where he held increasingly important positions in the Division of Computational Bioscience (DCB) of the Center for Information Technology, finally serving as its Associate Director. He began his career in electrical engineering. After earning an S.D. degree in computer science from George Washington University, he married the two career fields and now specializes in the application of computational science and engineering methodologies to biomedical data mining, systems biology, and high-performance biocomputing. He authored and coauthored numerous articles in journals such as Science, Journal of Computational Biology, Bioinformatics, and Cancer Research.



Edward R. Dougherty is a Professor in the Department of Electrical Engineering at Texas A&M University in College Station, Tex, Director of the Genomic Signal Processing Laboratory at Texas A&M University, and Director of the Computational Biology Division of the Translational Genomics Research Institute in Phoenix, Ariz. He holds a Ph.D. degree in mathematics from Rutgers University and an M.S. degree in Computer Science from Stevens Institute of Technology. He is the author of twelve books, Editor of five others, and author of more than one hundred and ninety journal papers. He is an SPIE Fellow, is a Recipient of the SPIE Presidents Award, and has served as an Editor of the Journal of Electronic Imaging for six years. He has contributed extensively to the statistical design of nonlinear operators for image processing and the consequent application of pattern recognition theory to nonlinear image processing. His current research is focused in genomic signal processing, with the central goal being to model genomic regulatory mechanisms for the purposes of diagnosis and therapy.