Journal of Data Science **2**(2004), 347-370

# Evaluation of Missing Value Estimation for Microarray Data

Danh V. Nguyen[1], Naisyin Wang[2] and Raymond J. Carroll[2]
[1] *University of California, Davis and*
[2] *Texas A&M University*

*Abstract*: Microarray gene expression data contains missing values (MVs). However, some methods for downstream analyses, including some prediction tools, require a complete expression data matrix. Current methods for estimating the MVs include sample mean and K-nearest neighbors (KNN). Whether the accuracy of estimation (imputation) methods depends on the actual gene expression has not been thoroughly investigated. Under this setting, we examine how the accuracy depends on the actual expression level and propose new methods that provide improvements in accuracy relative to the current methods in certain ranges of gene expression. In particular, we propose regression methods, namely multiple imputation via ordinary least squares (OLS) and missing value prediction using partial least squares (PLS).

Mean estimation of MVs ignores the observed correlation structure of the genes and is highly inaccurate. Estimating MVs using KNN, a method which incorporates pairwise gene expression information, provides substantial improvement in accuracy on average. However, the accuracy of KNN across the wide range of observed gene expression is unlikely to be uniform and this is revealed by evaluating accuracy as a function of the expression level.

*Key words:* Gene expression, imputation, microarray missing value estimation; K-nearest neighbors, partial least squares.

## 1. Introduction and Background

DNA microarrays, designed to monitor mRNA expression levels of thousands of genes in concert, are used to investigate various biological processes. Gene expression data obtained from microarray experiments, like other experimental data, often contain missing values (MVs). Reasons for MVs include insufficient resolution, image corruption, array fabrication error, and excessive background noise among others. However, some data analysis methods applied to gene expression data, including some classification and model-based clustering techniques, are not equipped to handle missing data. For methods that require a complete

expression matrix, the primary approaches to missing data include (1) removing data points with MVs before the analysis or (2) estimating the MVs and then proceeding to the analysis. We note that there are some methods, such as clustering, that utilize only available data (implicit imputation), although it is not the focus of this paper. Currently, some analyses of gene expression data adopt approach (1) to handle MVs, partly for its simplicity. Approach (2), estimating the MVs before analysis, is less common and only naive methods, such as replacing MVs with zeros or the sample means, have been used. Such methods are highly inaccurate. One of the earliest use of a more sophisticated MV estimation method is by Dudoit, Fridlyand and Speed (2002), where the method of K-nearest neighbors (KNN) was used to estimate MVs before applying various classification methods.

Recognizing the potential benefits of estimating MVs accurately in gene expression data before applying analysis methods, Troyanskaya *et al.* (2001) provided the first substantial evaluation of various MV estimation methods, including KNN. K-nearest neighbors was found to give accurate prediction of MVs on average. However, like other prediction methods, the accuracy of KNN is unlikely to be uniform across the wide range of observed gene expression. Thus, in this work, we evaluate the relative accuracy of imputation methods, including mean and KNN imputation, as a function of the observed gene expression level. Also, we propose and evaluate regression methods, OLS and PLS, for estimating MVs which provide improvement in accuracy over some ranges of expression values where KNN did not performed as well. Both cDNA and oligonucleotide microarray data sets are used in the study.

The paper is organized as follows. We first describe the general framework of the study design to evaluate imputation methods and also introduce the necessary notations in Section 2. Next, the estimation or imputation methods, which include mean, KNN, OLS regression, and PLS regression are described in Section 3. A brief description of the microarray gene expression data sets follows and a summary of the findings are given in the Results Section 4. In Section 5 we discuss various issues, including the selection of method parameters, sensitivity to initial values, and other missing data mechanisms. We conclude in Section 6 with a summary of some practical guidelines and issues for further investigation. All algorithms in the paper are described in sufficient details for implementation and are also made available at http://stat.tamu.edu/∼dnguyen/supplemental.html. Implementation codes in Matlab will be made available there as well.

## 2. Evaluation Procedure

Given $n$ microarray experiments, each of which contains the mRNA expressions of $p$ genes, the data can be organized into an $n \times p$ matrix of gene expression

values. Denote $x_{ij}$ to be the expression value of gene (column) $j$ in sample (row) $i$. To evaluate the accuracy of imputation methods we used the following evaluation design.

1. Given a real gene expression matrix, the (real) MVs are removed to form a *complete* gene expression matrix with no MVs. This is denoted by $\mathbf{X} = (x_{ij})_{n\times p}$.

2. Next, a proportion $q$, $0 < q < 1$, of MVs are intentionally introduced by randomly removing values in $\mathbf{X}$. We also examine the case where the missingness depends on the actual gene expression.

3. Imputation methods are applied to estimate the MVs (values removed in step 2).

4. The imputed or estimated values are compared to the true values to assess accuracy.

In step 2, the missing values are introduced by systematically removing expression values from the complete expression matrix. Let $\mathcal{L}$ be the set of genes with some MVs and denote the introduced MVs of gene $j \in \mathcal{L}$ by $\mathbf{y}_j' = (y_{1j}, y_{2j}, \ldots, y_{m_j j}), m_j \geq 1$. The set of all $M = \sum m_j$ MVs can be expressed as $\mathbf{y}' = (\mathbf{y}_1', \mathbf{y}_2', \ldots, \mathbf{y}_m') \equiv (y_1, y_2, \ldots, y_M)$, where $m = |\mathcal{L}|$ is the number of genes with some MVs. We refer to a particular gene with MVs to be estimated as the *target* gene. The set of genes with available information for estimating the MVs of the target gene is the set of *candidate* genes.

Although obvious, we note that the imputation methods applied to estimate the MVs in step 3 should not utilize, in any way, the true values that were removed from $\mathbf{X}$ in step 2. A notation, that is useful to track the MVs and non-MVs of $\mathbf{X}$, after step 2, is the missing indicator matrix, $\mathbf{R}$, introduced in Rubin (1976). The $ij$th element of $\mathbf{R}$ is $r_{ij} = 1$ if the expression value $x_{ij}$ is observed and $r_{ij} = 0$ if $x_{ij}$ is missing. Using this notation, all computations involved in any imputation method must be applied to only the available data, which is $\{x_{ij} : r_{ij} = 1\}$. This caution is relevant to even "preprocessing" algorithms applied to $\mathbf{X}$ (prior to step 3), because the MVs would not be available in practice.

In step 4 the accuracy of the imputation method is evaluated. Since the MVs were introduced intentionally, they are therefore known. Thus, the vector of estimates ($\hat{\mathbf{y}}$) can be compared to the vector of true values ($\mathbf{y}$) to assess the accuracy of an imputation method. For example, Figure 1 displays the normalized relative estimation error (RAE) curve ($|y - \hat{y}|/|y|$) as a function of the true expression value ($y$) in a cDNA microarray data set for mean, KNN, and regression (PLS and OLS) estimation methods. It is apparent from Figure 1 that the accuracy of

an estimation method is not uniform (flat) across the full range of gene expression levels. The vertical lines marks the mean, mean ± 1 standard deviation, and mean ± 3 standard deviations ($\mu$, $\mu \pm \sigma$, $\mu \pm 3\sigma$) of true expression values. Note that there is a range of expression where KNN performs better compared to other methods (e.g., near $\mu$), but other methods (e.g., PLS regression) outperform it in other ranges.

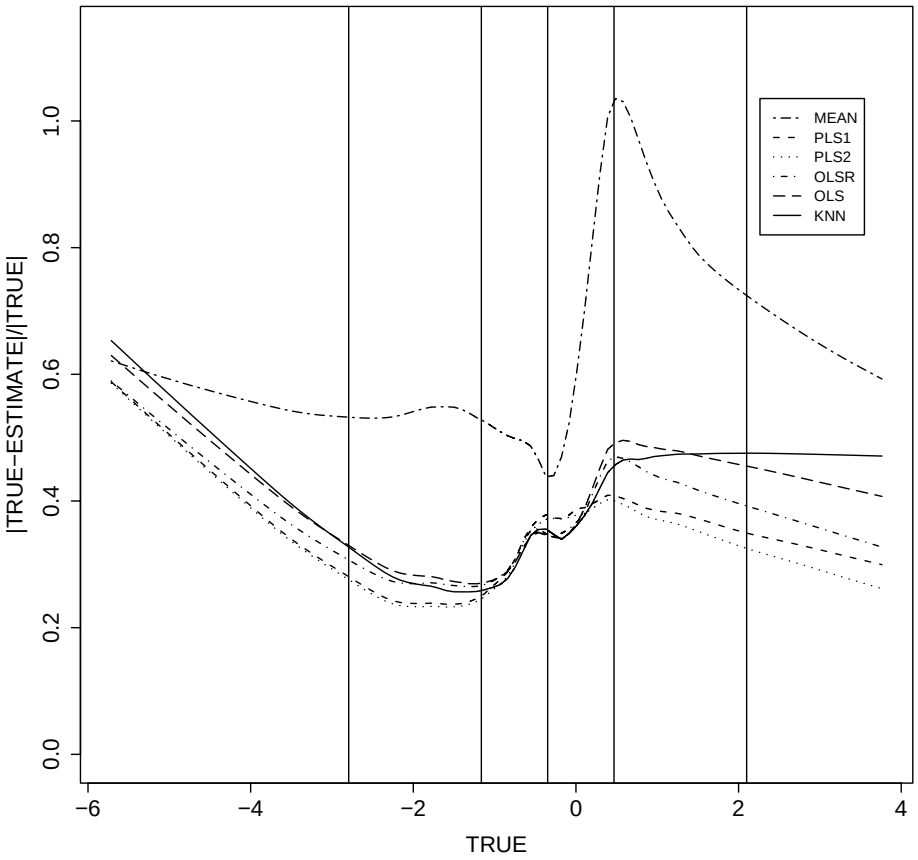


Figure 1: RAE error curves for BCLL data. Given are the relative absolute errors ($y$-axis) as a function of the true gene expression values ($x$-axis) for MEAN, PLS1, PLS2, OLSR, OLS, and KNN imputation. The lines plotted are the loess fits through the scatter plots of the $M$ true values ($y_i$) and the errors $e_i = |y_i - \hat{y}_i|/|y_i| = |\text{TRUE} - \text{ESTIMATE}|/|\text{TRUE}|$ for each imputation method. The mean true expression ($\mu$), $\mu \pm \sigma$, and $\mu \pm 3\sigma$, where $\sigma$ is the standard deviation of the true expression, are marked with vertical lines. See section 'Construction of the error curves' for details of computation. The data set used is the B-cell chronic lymphocytic leukemia (BCLL) cDNA data.

Table 1: Summary of some characteristics of the complete gene expression matrices, $\mathbf{X}$, used to test imputation methods. For cDNA arrays, each expression value, $x_{ij}$, is the log expression ratio of the experimental to reference, background corrected, expression intensity. A proportion of $q = 0.05,\ 0.10,\ 0.15$, and 0.20 MVs is introduced to each complete expression matrix, $\mathbf{X}$, of size $n \times p$.

| | **Lymphoma** | | **Leukemia** | | **Breast Cancer** |
|---|---|---|---|---|---|
| DATA: | DLBCL | BCLL | AML | ALL | *BRCA1*, *BRCA2*, sporadic |
| # arrays, $n$ | 45 | 29 | 47 | 25 | 22 |
| # genes, $p$ | 5,353 | 5,079 | 2,260 | 2,560 | 3,226 |
| type | cDNA | cDNA | oligo. | oligo. | cDNA |

To evaluate the accuracy of the imputation methods, we implemented the evaluation design described above to five complete expression matrices (summarized in Table 1). The percentage of induced missing data is 5%, 10%, 15% and 20%. For cDNA array data we examined estimation using ratio data as well as estimation based on data from each channel separately. In addition, we examined the estimation results under the setting where the rate of missing data is dependent on the expression level.

## 3. Imputation Methods for Microarray Data

### 3.1 Ignoring gene correlation structure: Mean imputation

One of the simplest imputation methods used for microarray data is mean imputation, wherein the MVs of target gene $j$ are estimated by the observed average expression of gene $j$. The average is taken over the available values of gene $j$ in the $n$ experiments. More precisely, the imputed values of gene $j \in \mathcal{L}$, are given by

$$\hat{y}_{vj} = \frac{\sum_{i=1}^{n} r_{ij} x_{ij}}{\sum_{i=1}^{n} r_{ij}} \qquad v = 1, \ldots, m_j \geq 1.$$

Note that mean imputation does not utilize any information between genes across the $n$ experiments. Although it is an improvement over replacing the MVs with zeros (or a positive constant), as is sometimes done with microarray data (e.g., Alizadeh *et al.*, 2000), there are other much more precise methods.

### 3.2 Incorporating pairwise gene information: KNN imputation

One approach to improve upon mean imputation is to incorporate the information between genes contained in the structure of the gene expression matrix.

To improve accuracy, the method of KNN imputation has been used in the estimation of MVs in microarray experiments (Dudoit, Fridlyand and Speed, 2002; Troyanskaya *et al.*, 2001). KNN uses *pairwise* information between the target gene with MVs to be estimated and the remaining candidate genes. A MV of target gene $j$ in experiment $v$ is imputed based on $K$ candidate genes with available values in experiment $v$ corresponding to $K$ genes "closest" or "nearest" to target gene $j$. The choice of a distance measure to select the $K$ genes giving good accuracy depends on various factors, some of which are data dependent. For example, it was found that a weighted Euclidean distance performed well when it was applied to cDNA experiments from the yeast *Saccharomyces cerevisiae* (Troyanskaya *et al.*, 2001). Others have used the Pearson correlation to select the $K$ nearest genes (Dudoit Fridlyand and Speed, 2002).

Without lost of generality, suppose that a target gene $j \in \mathcal{L}$ has a MV in experiment $v$. The set of available candidate genes for estimating the MV, $y_{vj}$, are all genes with available values in experiment $v$ corresponding to MV $y_{vj}$. Denote this set of candidate genes by $\mathcal{C}_v$. Next, $K$ genes among the candidate gene set, i.e., from $\mathcal{C}_v$, are selected so that they are closest to gene $j$. These $K$ selected candidate genes are referred to as the $K$ *nearest neighbors* of the target gene. The rationale underlying such a procedure is that candidate genes closer to target gene $j$ may provide better information for estimating MVs. Often the Euclidean distance or some variant of it is used. For example, the weighted Euclidean distance measure between target gene $\mathbf{x}_j$ and each candidate gene $\mathbf{x}_k$, $k \in \mathcal{C}_v$ based on the available data is

$$d_{jk} \equiv d(\mathbf{x}_j, \mathbf{x}_k) = \left\{ n_{jk}^{-1} \sum_{i=1}^{n} r_{ij} r_{ik} (x_{ik} - x_{ij})^2 \right\}^{1/2}, \qquad k \in \mathcal{C}_v \tag{3.1}$$

where $n_{jk} = \sum_{i=1}^{n} r_{ij} r_{ik}$ is the number of jointly available values between $\mathbf{x}_j$ and $\mathbf{x}_k$. Note that the distance (3.1) is weighted by the number of data points, $n_{jk}$. Furthermore, note that $\mathcal{C}_v$ depends on the target gene $j$, but the dependence on $j$ is suppressed in the notation throughout for simplicity.

Let $\mathcal{C}_v^*$ be the set of column labels corresponding to the *selected* $K$ nearest neighbors. The estimated value for MV $y_{vj}$ is the weighted average of expression values of the $K$ selected candidate genes in experiment $v$,

$$\hat{y}_{vj} = \sum_{k \in \mathcal{C}_v^*} w_k x_{vk}, \tag{3.2}$$

where $w_k = 1/(d_{jk} C)$, $k \in \mathcal{C}_v^*$ are the weights and $C = \sum_{k \in \mathcal{C}_v^*} d_{jk}^{-1}$ is the normalizing weight constant. Weights are inverse of the distances, thus, giving higher weights to expression values from candidate genes closer to target gene $j$. The

distances and weights described above were used in (Troyanskaya *et al.*, 2001) in the algorithm called KNNimpute.

### 3.3 Incorporating pairwise gene information: Imputation via repeated OLS regression

Rather than taking a weighted average of the $K$ available values, as is done in KNN, multiple estimation of each MV can be obtained by repeatedly regressing the target gene on each of the $K$ selected candidate genes. Consider the selection of $K$ nearest candidate genes based on the distance given by (3.1) used in KNN imputation. As before, denote the selected $K$ candidate genes for estimating the MV of target gene $j$ in experiment $v$ by $\mathcal{C}_v^*$. For each of the $K$ selected candidate genes, we can obtain an estimate of the MV of target gene $j$ based on ordinary least squares (OLS) regression. More precisely, the $k$th OLS imputation of a MV of target gene $j \in \mathcal{L}$ based on available data is given by

$$\hat{y}_{vj}^{(k)} = \bar{x}_j + b_j^{(k)}(x_{vk} - \bar{x}_k), \qquad k \in \mathcal{C}_v^*,$$

where $x_{vk}$ is the available value of candidate gene $k \in \mathcal{C}_v^*$, $\bar{x}_j$ and $\bar{x}_k$ are the sample means based on jointly available data, and $b_j^{(k)}$ is the regression slope coefficient using the available data. The final estimate of the MV of target gene $j$ in experiment $v$ is the weighted average of the $K$ separate estimates,

$$\hat{y}_{vj} = \sum_{k \in \mathcal{C}_v^*} w_k \hat{y}_{vj}^{(k)}. \tag{3.3}$$

The weights can be based on the distance used to select the $K$ nearest genes, as in (3.2). If equal weight is desired for each of the $K$ separate estimates, then $w_k = 1/K$.

### 3.4 Incorporating global gene structure: Imputation via PLS regression

In this section we introduce the method of partial least squares (PLS) imputation using PLS regression. PLS is a useful prediction and modelling tool in chemometrics (Helland, 1988; Höskuldsson, 1988) and has been applied to cancer classification problems based on gene expression data (Nguyen and Rocke, 2002a, 2002b, 2002c). Rather than *select* $K$ nearest candidate genes for imputation based on pairwise distances, as in KNN, PLS uses all the candidate gene expressions, as well as the available values from the target gene to estimate the MVs. Based on the candidate gene expression matrix and the available values of the target gene, PLS constructs a sequence of gene components. Next, to estimate MVs of target gene $j$, a regression model with the target gene as the

response variable and $K_P$ PLS gene components as predictors is fitted. The MVs are predicted using the fitted PLS regression model.

Suppose that target gene $j \in \mathcal{L}$ has MVs which are to be estimated. All genes that have available values corresponding to the MVs of gene $j$ comprise the set of candidate genes. Denote the expression matrix of the candidate genes, without column $j$, as $\mathbf{X}_{-j}$. This expression matrix, $\mathbf{X}_{-j}$, can be partitioned according to the available values (A) and MVs (M) of gene $j$ as follows,

$$\mathbf{x}_j = \begin{pmatrix} \mathbf{x}_j^A \\ \mathbf{x}_j^M \end{pmatrix}, \qquad \mathbf{X}_{-j} = \begin{pmatrix} \mathbf{X}_{-j}^A \\ \mathbf{X}_{-j}^* \end{pmatrix},$$

where $\mathbf{x}_j^A$ is a vector of available values of target gene $j$, $\mathbf{x}_j^M$ a vector of missing (empty) entries to be imputed (filled), $\mathbf{X}_{-j}^A$ is a $n_j \times p_j$ matrix of available values corresponding to $\mathbf{x}_j^A$, and $\mathbf{X}_{-j}^*$ consists of available values corresponding to the MVs of target gene $j$ ($\mathbf{x}_j^M$). In this setup the pair $(\mathbf{X}_{-j}^A, \mathbf{x}_j^A)$ is the *training* data and $\mathbf{X}_{-j}^*$ is the *test* data that will be used to predict the MVs $\mathbf{x}_j^M$.

Note the that the number of samples (rows) is much smaller than the number of available genes (columns) in the training data, i.e., $n_j << p_j$. Hence, dimension reduction is necessary. PLS is a dimension reduction method which extracts the gene components sequentially to maximize the sample covariance between the target gene and the linear combination of the set of candidate genes. More precisely, in this imputation context, the $k$th PLS step seeks a weight vector, $\mathbf{w}_k(j)$ $(p_j \times 1)$, satisfying the following objective criterion,

$$\mathbf{w}_k(j) = \underset{\mathbf{w}'\mathbf{w}=1}{\operatorname{argmax}} \operatorname{cov}^2(\mathbf{X}_{-j}^A \mathbf{w}, \mathbf{x}_j^A) \qquad j \in \mathcal{L}, \tag{3.4}$$

subject to the orthogonality constraints

$$\mathbf{w}_k'(j)\mathbf{S}\mathbf{w}_d(j) = 0, \qquad \text{for all } 1 \le d < k, \tag{3.5}$$

where $\mathbf{S} = \mathbf{X}_{-j}^{A'}\mathbf{X}_{-j}^A$. Equation (3.4) says that weights are assigned to each gene such that the covariance between the target gene and the linear combination of the candidate genes is maximized. The weights are non-linear functions of both the candidate and target expression values. Note that a sequence of weights, $\mathbf{w}_1(j), \mathbf{w}_2(j), \ldots$, is obtained via (3.4) for *each* gene $j \in \mathcal{L}$ with MVs. Furthermore, these weights depends on what is being predicted, namely target gene $j$.

The resulting linear combinations with maximum covariance with target gene $j$, namely $\mathbf{t}_k^A(j) = \mathbf{X}_{-j}^A \mathbf{w}_k(j)$, are the PLS gene components. One interpretation of the sequence of weight vectors (3.4) and PLS gene components is as follows. In the first step of PLS the most important mode of covariation exhibited between the candidate gene expressions and the target gene is captured. The second step

of PLS still seeks the most important mode of covariation between the candidate genes and the target, but, as stipulated by (3.5), it further requires the linear combination constructed to be orthogonal to the one constructed at the first step. Thus, the second PLS components captures a new mode of target and candidate gene covariation that is different from the first, i.e., $\text{cov}(\mathbf{t}_2^A, \mathbf{t}_1^A) = 0$. As in the second step of PLS, subsequent steps seek the strongest mode of candidate and target gene covariation and the $k-1$ orthogonality constraints imposed require that the $k$th linear combination identifies a mode of predictor and response covariation distinct from those previously identified (by the previous $k-1$ linear combinations).

A linear regression model based on the available values is fitted using the constructed PLS gene components as predictors. The fitted expression values of target gene $j$ is

$$\hat{\mathbf{x}}_j^A = \mathbf{T}^A(j)\hat{\boldsymbol{\beta}}_j, \tag{3.6}$$

where $\mathbf{T}^A(j)$ is a matrix of the $K_P$ PLS gene components and $\hat{\boldsymbol{\beta}}_j$ is the least squares regression coefficient estimates. The expression values of candidate genes, $\mathbf{X}_{-j}^*$, corresponding to missing entries, $\mathbf{x}_j^M$, namely the test data, are used to construct the test PLS components, $\mathbf{T}^*(j)$, using only the training information (3.4). The test components are substituted into the training PLS regression model (3.6) to predict the MVs,

$$\hat{\mathbf{x}}_j^M = \mathbf{T}^*(j)\hat{\boldsymbol{\beta}}_j.$$

Computations involved in (3.4) require $\mathbf{X}_{-j}^A$ to be complete. In practice we first form a complete matrix by replacing the MVs with "initial" estimates from KNN.

## 4. Results

### 4.1 Microarray test data sets

The estimation methods were tested on a variety of cDNA and oligonucleotide (Affymetrix) arrays. Table 1 summarizes some characteristics of the 5 complete expression matrices, $\mathbf{X}$, used to evaluate the imputation methods. The first complete expression matrix, $\mathbf{X}$, consists of $n = 45$ diffuse large B-cell lymphoma (DLBCL) arrays with $p = 5,353$ cDNA probes. We formed a second complete expression matrix with $n = 29$ B-cell chronic lymphocytic leukemia (BCLL) arrays with complete data for $p = 5,079$ cDNAs. Both lymphoma expression matrices, DLBCL and BCLL, are from the same set of experiments, using a combination of 9 lymphoma cell lines as the reference (Alizadeh *et al.*, 2000). The third and fourth complete expression matrices consists of $n = 47$ acute myeloid leukemia (ALL) samples and $n = 25$ acute lymphoblastic leukemia (ALL) (Golub

*et al.*, 1999). Complete AML and ALL expression matrices contain data for $p = 2,260$ and $p = 2,560$ oligonucleotide sets respectively. The fifth data set used to evaluate estimation accuracy is a cDNA data set consisting of 7 breast cancer (BC) samples with mutation in the *BRCA1* gene, 8 with mutation in the *BRCA2* gene, and 7 sporadic cases with neither mutations detected (Hedenfalk *et al.*, 2001). The complete expression matrix of the combined 22 BC samples consists of 3,226 cDNAs. The reference used for all 22 BC samples is a nontumorigenic breast cell line. More detailed protocols for both cDNA and oligonucleotide experiments are available in Nguyen *et al.*, (2002d). Prior to applying the imputation methods the data sets were log-transformed.

Note that the first four complete matrices were formed based on biological samples of one type, such as all ALL samples or all AML samples. This is a reasonable strategy since prediction generally can be poor for a collection of heterogenous samples. However, it may be of interest to see the performance of the methods under heterogenous or diverse biological samples. The complete matrix formed from the breast cancer data combined three types of biological samples, namely samples with mutation in the *BRCA1* gene, samples with mutation in the *BRCA2* gene, and sporadic samples with neither mutations detected. This serves as a test case for comparing the imputation methods with diverse biological samples.

### 4.2 Assessment of estimation accuracy as a function of gene expression

The two basic questions regarding evaluation of estimation accuracy being addressed in the current paper are:

1. Does estimation accuracy depend on the actual gene expression level? If accuracy depends on the expression level, then it is natural to also examine:

2. How does estimation accuracy depend on the gene expression level? For example, is the accuracy higher for some range of expresson values and lower for other ranges?

A cursory examination of the error curves in Figure 1, for example, indicates that the accuracy is not uniform across the range of observed gene expression. If the accuracy is independent of the gene expression levels then the error curves would all be flat horizontal lines. Therefore, we focus on question 2 above.

For the data in displayed in Figure 1, an *overall average* estimation accuracy can be measured by a summary statistics, such as $\{M^{-1}\sum_{i=1}^{M}|y_i - \hat{y}_i|^s/c\}^{1/2}$ $(s = 1, 2, \ldots)$, where $y$ is the true expression value, $\hat{y}$ is its estimated value, and $c$ is a normalization constant (e.g., the average of the data value in the data set). The root mean square (RMS) error (Troyanskaya *et al.*, 2001) and the root

Table 2: Overall average estimation error measured by RMS error, $\{M^{-1}\sum(y_i - \hat{y}_i)^2\}^{1/2}$. The values give an indication of the overall average estimation accuracy for the various estimation methods (A; MEAN, KNN, OLS, OLSR, PLS1 & PLS2). Also given (B) are the RMS errors when the methods are applied to the two channels of the BCLL cDNA data set separately and (C) the RMS errors for expression-dependent missing rate. See section 'Construction of the error curves' for details.

| | **Data** | **MEAN** | **KNN** | **OLS** | **OLSR** | **PLS1** | **PLS2** |
|---|---|---|---|---|---|---|---|
| **A.** | BCLL | 0.6755 | 0.4578 | 0.4615 | 0.4519 | 0.4520 | 0.4447 |
| | DLBCL | 0.7298 | 0.3990 | 0.4035 | 0.4013 | 0.4074 | 0.4009 |
| | BRCA | 0.4493 | 0.3886 | 0.3901 | 0.3907 | 0.3882 | 0.3861 |
| | ALL | 0.6490 | 0.5791 | 0.5871 | 0.5701 | 0.5640 | 0.5735 |
| | AML | 0.6186 | 0.5440 | 0.5380 | 0.5221 | 0.5221 | 0.5240 |
| Estimation applied to separate channel–BCLL data. | | | | | | | |
| **B.** | Cy5 | 1.8994 | 0.6098 | 0.6140 | 0.6078 | 0.5961 | 0.5819 |
| | Cy3 | 0.8460 | 0.4211 | 0.4226 | 0.4125 | 0.4275 | 0.4071 |
| Expression-dependent missing rate. | | | | | | | |
| **C.** | Cy5 | 1.2435 | 0.7749 | 0.7817 | 0.7645 | 0.7596 | 0.7440 |
| | Cy3 | 1.0056 | 0.5763 | 0.5779 | 0.5557 | 0.5774 | 0.5540 |

mean absolute (RMA) error is when $s = 2$ and $s = 1$ in the above formula, respectively. Such summary measures are essentially averages of the individual errors: $e_1 = |y_1 - \hat{y}_1|^s, e_2 = |y_2 - \hat{y}_2|^s, \ldots, e_M = |y_M - \hat{y}_M|^s$. They give an indication of the overall average estimation error, which can be roughly interpreted as averages of each error curves in Figure 1. Assessing the average estimation error, as measured by RMS for instance, is important and has been addressed in detail (Troyanskaya *et al.*, 2001) and we refer the interested reader there for a more in-depth treatment. We are investigating here a different issue, namely the dependence of the estimation accuracy on the gene expression and this cannot be adequately assessed by RMS error or any other summary measure. Therefore, to more fully address question 2 posed above, we need to examine the entire error curve, because there are systematic rises and falls in the error curve (accuracy) as a function of the expression levels that will, inevitably, be lost when averaging the errors over the entire range of gene expression values. Table 2 gives the relative *overall average performance* of the various methods using RMS. As will be detailed in the following sections, although the overall average estimation error, as measured by RMS, may be higher for K-nearest neighbors (KNN) than for

partial least squares regression (PLS), KNN performs better than PLS when the expression is near the mean.

### 4.3 Construction of the error curves

Accuracy of the estimation methods were evaluated with 5%, 10%, 15% and 20% missing data. For example, Figure 1 displays the error curves for MEAN, PLS1, PLS2, OLSR, OLS, and KNN imputation for the BCLL cDNA data with with 10% missing. PLS1 denotes imputation using PLS regression as described earlier and PLS2 is also a PLS regression method, but uses the genes with high sum of squares PLS weights (3.4) from the PLS1 fit (Wold, 1994; PROC PLS, SAS Institute, 1999). OLS is imputation via ordinary least squares regression with weights selected using the weighted Euclidean distance (3.1), as in KNN. OLSR denotes OLS imputation but with distances and weights based on the Pearson correlation coefficient.

The error curves, lines plotted in Figure 1, are the loess fits through the scatter plots of the true values, $y_1, \ldots, y_M$ ($x$-axis) versus the errors $e_1, \ldots e_M$ ($y$-axis). The same window size of 0.20 was used in the loess fits for all methods. The individual errors are given by $e_i = |y_i - \hat{y}_i|^s/c_i$, where $\hat{y}_i$ is the estimated value and $c_i$ is a normalization constant. Taking $s = 1$ and $c_i = 1$ gives the absolute error, $e_i = |y_i - \hat{y}_i|$, which is a direct and simple measure of error. An advantage of using absolute error is its simple and natural interpretation. However, it may be more appropriate to consider a relative measure of error which accounts for the magnitude of the true value. More specifically, taking $c_i = |y_i|$ gives relative absolute error (RAE), $e_i = |y_i - \hat{y}_i|/|y_i| = |\text{TRUE} - \text{ESTIMATE}|/|\text{TRUE}|$. Note that the RAE measure is appropriate, but has some minor drawbacks. It is undefined for $y_i = 0$ and for small $y_i$ values the ratio is unstable (e.g., could be artificially inflated). Thus, the error patterns of the various estimation methods can not be reliably compared in this region, say $|y_i| \leq \epsilon$, although RAE may be adequate for $|y_i| > \epsilon$. We first illustrate that the conclusions remain essentially the same, whether the error curves are constructed from individual RAE or absolute errors. We then propose an alternative construction of error curves that (1) resolves the aforementioned drawbacks of RAE, (2) allows for evaluation of the error patterns equally across the range of gene expression values, and (3) facilitates easy comparison and interpretation.

To examine the RAE measure discussed above, Figure 1 gives the RAE error curves for the various estimation methods using RAE $= |y_i - \hat{y}_i|/|y_i|$ for $|y_i| > \epsilon$ and $|y_i - \hat{y}_i|/\epsilon$ for $|y_i| \leq \epsilon$ ($\epsilon = 0.5$). The vertical lines marks the mean of the true expression values ($\mu$), $\mu \pm \sigma$, and $\mu \pm 3\sigma$, where $\sigma$ is the standard deviation of the true expression values. Note the range of expression levels where KNN estimation out-performed the other methods $\{y \approx (\mu - .75\sigma, \mu + .60\sigma)\}$ and vice

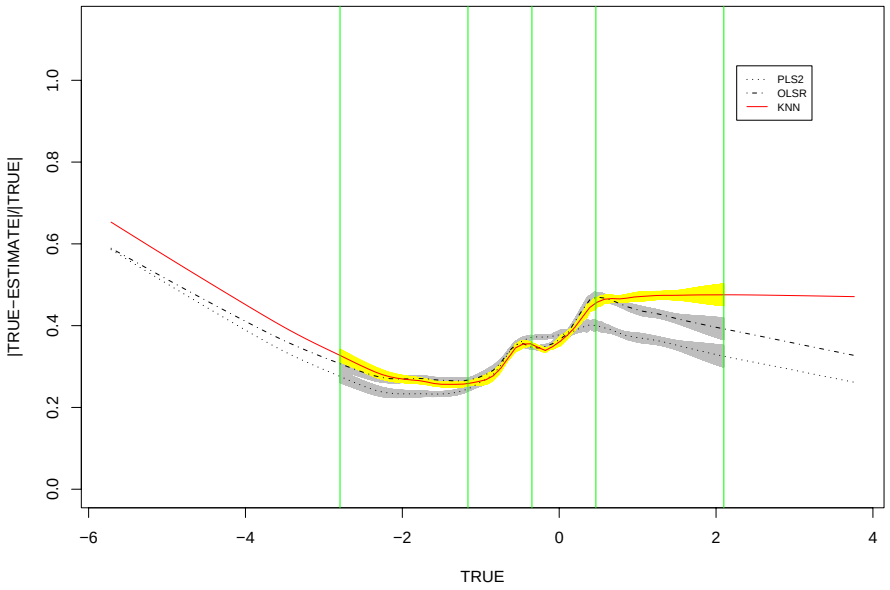


Figure 2: Error curves in Figure 1 with standard error bands. Given are the KNN, OLSR, and PLS2 error curves in Figure 1 with corresponding error bands added. Error bands are given for the $\pm 3$ standard deviation of true value ($x$-axis) range to avoid artifacts. Note that the OLS curve and the PLS1 curve which is very similar to the PLS2 curve are not displayed. Mean imputation performed extremely poorly and is not of particular interest. The standard error bands were obtained using the bootstrap method (Efron and Tibshirani, 1993) and with 500 replications.

versa $\{y \lesssim \mu - .75\sigma,\ y \gtrsim \mu + .60\sigma\}$ for this data set. The KNN, OLSR and PLS2 curves in Figure 1 are repeated in Figure 2 along with the standard error bands. (To avoid clutter in the plot, the OLS curve and the PLS1 curve which is very similar to the PLS2 curve are not displayed. Mean imputation performed very poorly and is not of particular interest as well.) Similar results were obtained from error curves based on individual absolute errors $|y_i - \hat{y}_i|$. However, due to wide range of expression values, the region where KNN performed well relative to the other methods is not apparently clear in the error curves constructed from absolute errors. In addition, since we are assessing the performance of the proposed new methods (PLS1, PLS2, OLSR, and OLS) relative to KNN estimation, the RAE curves in Figure 1 can be presented by dividing the RAE error curves for the new methods by the RAE error curve for KNN. For example, the new PLS2 error curve, *relative to KNN*, is obtained as

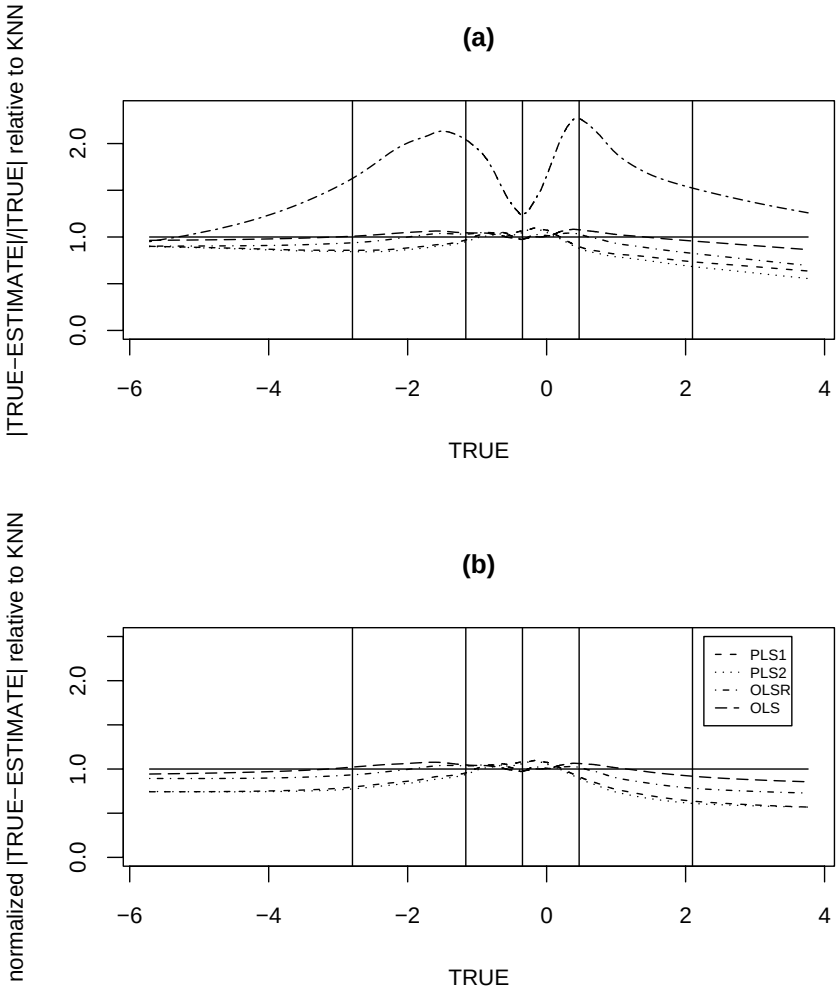


Figure 3: (a) RAE error curves presented relative to KNN for BCLL data and (b) Error curves relative to KNN for BCLL data.

$$\frac{\{\text{PLS RAE curve } (|y_i - \hat{y}_i^{(\text{PLS})}|)/|y_i|\}}{\{\text{KNN RAE curve } (|y_i - \hat{y}_i^{(\text{KNN})}|)/|y_i|\}}. \tag{4.1}$$

These relative RAE curves are given in Figure 3(a). Note that in Figure 3(a) the KNN error curve is represented by the flat horizontal line at one. We emphasize that the information in Figures 1 and 3(a) are *identical*. However, the relative comparison and interpretation of the error curves is straight-forward from Figure 3(a): (1) The region where KNN out-performs the other methods is clear — it is

the region where the error curves rise above the (KNN reference) horizontal line at 1. (2) The region where the regression methods out-perform KNN estimation is simply where the error curves fall below the horizontal reference line at 1.

In Figure 3(a) the RAE error curves are presented relative to KNN for BCLL data. The identical error curves in Figure 1 are presented here relative to KNN. These error curves were obtained by dividing each of the error curves by the KNN error curve in Figure 1. The error curves here contain the same information as the error curves in Figure 1, but they are relative to KNN. Thus, the relative comparison and interpretation is straight-forward: (1) The region where KNN performs better relative to the other methods is the region where the error curves rise above the (KNN reference) horizontal line at 1. (2) The region where the regression methods out-perform KNN estimation are simply where the error curves fall below the horizontal reference line at 1. In Figure 3(b), error curves relative to KNN for BCLL data are presented. Given are the error curves normalized relative to KNN as in Figure 3(a), but constructed based on individual errors $e_i = |y_i - \hat{y}_i| = |\text{TRUE} - \text{ESTIMATE}|$. These error curves approximates those in Figure 3(a) well, thus the conclusions remain the same. In addition, they eliminate some of the drawbacks associated with RAE (detailed in the section 'Construction of the error curves') and, like Figure 3(a), the relative comparison and interpretation is straight-forward. Note that error curve for mean estimation was removed (here and throughout) because it performed extremely poor in all cases.

The error curves in Figure 3(a) (based on indivdual RAEs) are easy to interpret, but still have the aforementioned drawbacks. Error curves that alleviate these technical problems, but still retain the properties of the relative RAE error curves (in Figure 3(a)) is desirable. Thus, we also examined the error curves constructed as

$$\{* \text{ error curve } (|y_i - \hat{y}_i^{(*)}|)\} \div \{\text{KNN error curve } (|y_i - \hat{y}_i^{(\text{KNN})}|)\} \tag{4.2}$$

where $* =$ PLS1, PLS2, OLSR, and OLS (see Figure 3(b)). Note that the error curves in Figure 3(b) are very similar to the ones in Figure 3(a). The conclusions drawn are the same: (1) KNN is superior relative to PLS2 regression method in the range $\{y \approx (\mu - .75\sigma, \mu + .60\sigma)\}$. (2) PLS2 Regression method out-performed KNN in the range $\{y \lesssim \mu - .75\sigma,\ y \gtrsim \mu + .60\sigma\}$. As in Figure 3(a), the new error curves in Figure 3(b) clearly show the range of gene expression where KNN performed well.

Also, the absolute error, $|y_i - \hat{y}_i^{(\text{KNN})}|$, was observed to be far above zero, so the aforementioned drawbacks associated with RAE were no longer issues. Note that the use of (4.2) would have the same problems as RAE if the KNN estimation was perfect ($\hat{y}_i^{(\text{KNN})} = y_i$). However, this was not the case.

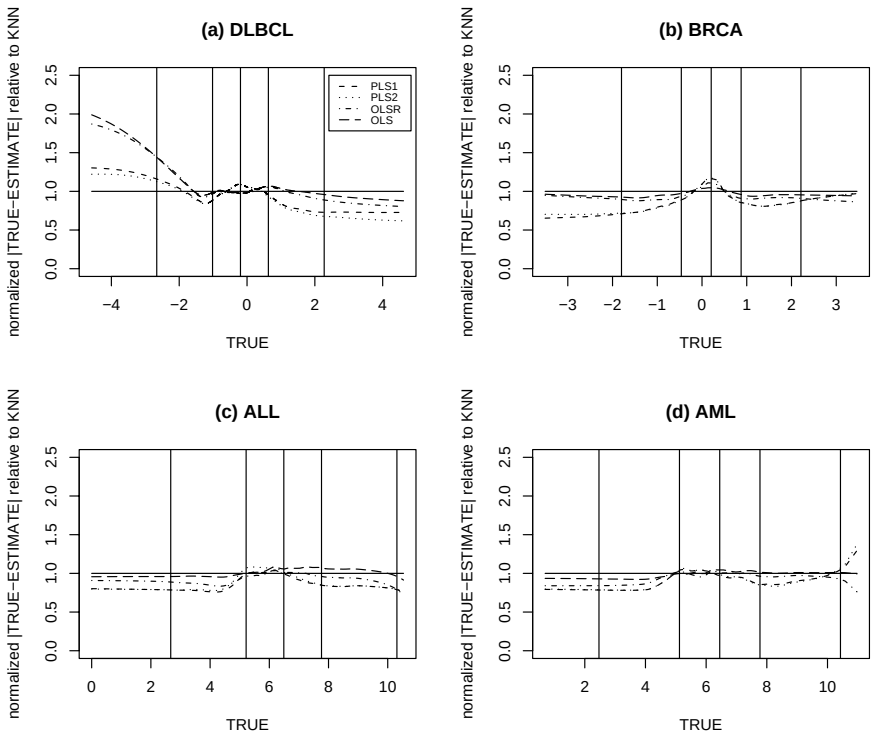


Figure 4: Error curves relative to KNN (a) DLBCL data. Given are the error curves normalized relative to KNN for the diffused large b-cell lymphoma (DLBCL) cDNA data. See Figure 3(b) caption for details. Similarly for (b)BRCA data, (c) ALL data, (d) AML data.

### 4.4 Observed estimation error pattern as a function of gene expression

For the reasons detailed in the previous section, we present the error curves given by (4.2) in this section, although the conclusions remain the same, whether using (4.1), absolute errors, or squared errors. The error curves using other error measures are made available at

http://stat.tamu.edu/∼dnguyen/supplemental.html.

The estimation error curves (4.2) for the data sets, summarized in Table 1 (labelled BCLL, DLBCL, AML, ALL, and BRCA), are given in Figures 3(b), 4(a)-(d). Error curves for all data sets are not flat horizontal lines, so the accuracy is not uniform across the observed expression levels. (Full-size figures are available at the above supplemental web site.) As can be seen from Figures 3 and 4 the estimation accuracy patterns fluctuates as a function of the gene expression

Table 3: Summary of estimation accuracy (error curves in Figures 3 and 4) over the observed range of expression values. The first column gives the range of expression where KNN was more accurate than PLS2 for the various data sets. The second column gives the range of expression where PLS2 was more accurate than KNN. The last column gives the range where OLSR is more accurate relative to KNN. To avoid potential artifacts at the very extreme ends, we conservatively restrict interpretations to a $\mu \pm 2.5\sigma$ range.

| | **Data** | **KNN** | **PLS2** | **OLSR** |
|---|---|---|---|---|
| **A.** | BCLL | $y \approx (\mu - .75\sigma, \mu + .60\sigma)$ | $y \lesssim \mu - .75\sigma$, $y \gtrsim \mu + .60\sigma$ | $y \lesssim \mu - 2.0\sigma$, $y \gtrsim \mu + 1.0\sigma$ |
| | DLBCL | $y \approx (\mu - .75\sigma, \mu + .75\sigma)$* | $y \lesssim \mu - .75\sigma$, $y \gtrsim \mu + .75\sigma$ | $y \gtrsim \mu + 2.0\sigma$ |
| | BRCA | $y \approx (\mu - .60\sigma, \mu + .50\sigma)$ | $y \lesssim \mu - .60\sigma$, $y \gtrsim \mu + .50\sigma$ | $y \lesssim \mu - .60\sigma$, $y \gtrsim \mu + .50\sigma$ |
| | ALL | $y \approx (\mu - 1.2\sigma, \mu)$ | $y \lesssim \mu - 1.2\sigma$, $y \gtrsim \mu$ | $y \lesssim \mu - .40\sigma$, $y \gtrsim \mu + .60\sigma$ |
| | AML | $y \approx (\mu - 1.2\sigma, \mu - .20\sigma)$ | $y \lesssim \mu - 1.1\sigma$, $y \gtrsim \mu - .20\sigma$ | $y \lesssim \mu - 1.2\sigma$, $y \gtrsim \mu + .75\sigma$ |
| *$y > \mu - 2\sigma$ | | | | |
| Estimation applied to separate channel–BCLL data. | | | | |
| **B.** | Cy5 | $y \approx (\mu - 1.3\sigma, \mu - .40\sigma)$ | $y \lesssim \mu - 1.3\sigma$, $y \gtrsim \mu - .40\sigma$ | $y \lesssim \mu - 2.0\sigma$, $y \gtrsim \mu + 1.9\sigma$ |
| | Cy3 | none | all | all |
| Expression-dependent missing rate. | | | | |
| **C.** | Cy5 | $y \approx (\mu - .50\sigma, \mu)$ | $y \lesssim \mu - .50\sigma$, $y \gtrsim \mu$ | $y \lesssim \mu - .60\sigma$, $y \gtrsim \mu + .10\sigma$ |
| | Cy3 | none | $y \lesssim \mu - .20\sigma$, $y \gtrsim \mu$ | all |

level. Furthermore, there are some notable patterns in this dependence between accuracy and actual gene expression level.

For example, KNN estimation is more accurate, compared to PLS regression estimation (e.g., PLS2), when the true expression is near the mean, $y \approx \mu$. This is indicated by the PLS error curves rising above the horizontal (KNN reference) line near $\mu$ (center vertical line). The range of expression where KNN is most accurate for the various data sets are more precisely summarized in the first column of Table 3. However, outside this range, PLS2 estimation is more accurate compared to KNN (Table 3, second column). This is apparent from the error curves falling below the KNN horizontal reference line. The ranges where OLSR is more accurate relative to KNN is also summarized in Table 3 (third column), although the relative gain in accuracy is substantially less than PLS2. Compared to both KNN and regression methods, the accuracy in MEAN imputation is unacceptably low across the observed range of expression levels for all five data sets. The general results described thus far are based on data within $\mu \pm 2.5\sigma$ to avoid extreme data points (although the same results hold for the $\mu \pm 3.0\sigma$ range as well). Thus, any potential artifact at extreme ends (e.g., $y$ beyond $\mu \pm 3.0\sigma$) does not play a role.

These findings hold similarly for other percentage of missing data (5%, 15%, and 20%). The estimation accuracy patterns for these cases are quite similar to the case of 10% missing data discussed above. Error curves for the other percentages of missing data are available at

http://stat.tamu.edu/∼dnguyen/supplemental.html.

We also investigated application of missing value estimation methods to the Cy5- and Cy3-channel data separately for cDNA microarrays. For illustration, we used the BCLL data (with 10% missing). A summary of the performance of the various estimation methods are given in Table 3B. The error curves for the Cy5- and Cy3-channel data are given in Supplemental Figures 5 and 6 respectively. The results here are similar to the earlier results, except for the Cy3-channel data. For the Cy3-channel data the accuracy of PLS2 is higher than KNN across the entire range of expression values. However, in the range $y \approx (\mu, \mu + .25\sigma)$ the gain in accuracy from PLS2 estimation over KNN is not as substantial as the gains from outside this expression range (see Supplemental Figure 5).

## 5. Discussion

### 5.1 Choosing the number of nearest neigbors $K$

The use of KNN imputation requires the selection of the number of nearest neighbors, the parameter $K$. The same $K$ was used for OLS and OLSR imputation. KNN imputation is repeated for a sequence of values for the parameter $K$

in the evaluation design, for each data set, to provide general guidelines for the selection of $K$ in practice. However, for a given data set in practice, missing data can be induced as described in the evaluation design to find $K$. Implementing the evaluation design to find $K$ for a given data set is preferable since the choice of $K$ is likely to depend on the particular data. For each of the five complete expression matrices (BCLL, DLBCL, BRCA, AML, and ALL) KNN imputation was carried out for $K = 1$, 2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 46, 54, 62, 70, 80, 90, 100, 140, 180, 250, 300, and 500. This was repeated for each data set with 5%, 10%, 15% and 20% missing data.

With more than $K = 30$ neighbors the estimation error, outside a moderate range of true expression, deteriorates rapidly and is unacceptably high, although the error is low near the mean expression level $\mu$. On the other extreme, when the number of neighbors is very small ($K < 6$) the accuracy is quite low near $\mu$. For example, for the BRCA data with 5% missing, a choice of $K$ between 10 and 22 is reasonable, although for this data $K = 14$ performed best overall. Similar results were observed for the other data sets and percentage of missing data. Detailed results of our study on the choice of $K$ is available at http://stat.tamu.edu/∼dnguyen/supplemental.html. We note that it was also found in Troyanskaya *et al.*, (2001) that, on average, KNN "is relatively insensitive to the exact value of $K$ within the range of 10-20 neighbors," based on a yeast cDNA array data. Our result supports this finding. For illustration of the methods, the results described above are based on $K = 14$ neighbors.

### 5.2 Choosing the distance metric in KNN and OLS

As described earlier, KNN imputation requires the selection of a "distance" function to measure closeness. A variety of distances were examined in Troyanskaya *et al.* (2001), including the Pearson correlation, and the results there indicate that the weighted Euclidean distance (3.1) did well. Our preliminary analysis compared the use of this distance and the Pearson correlation in KNN and results are similar to that in Troyanskaya *et al.* (2001; data not shown). Thus, for KNN we only used weighted Euclidean distance (3.1). However, for OLS imputation we used both the weighted Euclidean distance (OLS) and the correlation (OLSR).

### 5.3 Choosing $K_P$ in PLS

For PLS imputation, the number of PLS gene components, $K_P$, must be selected. For each data set and percentage of missing data we also examined the performance of PLS imputation using $K_P = 1$ to 8 gene components for prediction. The results suggest, not surprisingly, that in many cases there is not

a particular choice of $K_P$ which gives superior accuracy across the wide range of true values. However, differences in accuracy between various values of $K_P$ is small across the range of true expression values and a choice of $K_P$ beyond 5 appears unnecessary. Again, detailed results of our study of the choice of $K_P$ are available at http://stat.tamu.edu/∼dnguyen/supplemental.html.

Like classification with PLS where $K_P$ can be chosen to minimize the number of misclassifications (e.g., Nguyen and Rocke, 2002b), one can choose $K_P$ here to minimize, for example, the mean absolute error (MAE) $M^{-1}\sum_i |\text{true}_i - \text{estimate}_i|$ or some other measure of average error, $M^{-1}\sum_i e_i$. The values of $K_P$ minimizing the this total absolute error range from 1 to 5 for all cases. However, as mentioned earlier, the use of such overall measure in this context does not give indications of the accuracy of PLS imputation across the range of expression values. It is more informative to use this criterion in conjunction with the error curve to select $K_P$. For illustration of of the methods above we used $K_P = 4$.

Table 4: Variation of PLS estimation error for some examples with 15% missing data. Given are the average and variance of MAE (mean absolute error) from 14 repetitions of the study design.

| | $K_P = 3$ | | $K_P = 4$ | |
|---|---|---|---|---|
| | Mean | Variance | Mean | Variance |
| BRCA | 0.2880 | 0.0744 | 0.2882 | 0.0744 |
| DLBCL | 0.2919 | 0.0852 | 0.2939 | 0.0860 |
| ALL | 0.3667 | 0.1540 | 0.3685 | 0.1551 |

### 5.4 Variation of estimation error and sensitivity to initial values for PLS

To assess the variance in estimation error of PLS imputation we repeated the evaluation design. For example, Table 4 gives the average and variance of MAE (mean absolute error) from 14 repetitions of the evaluation design for PLS imputation. A similar assessment of variability for KNN is given in Troyanskaya *et al.* (2001) and the reader is referred there for details.

Computations involved with PLS dimension reduction (3.4) required a complete candidate expression matrix ($\mathbf{X}^A_{-j}$). However, this matrix contains missing entries so estimates from KNN were used as initial estimates to fill in $\mathbf{X}^A_{-j}$. Sensitivity of PLS imputation to the initial estimate was assessed. The results we have presented for PLS uses KNN estimates for the initial values and this appears to work well. To see how PLS would perform with poor initial estimates, PLS imputation for the BCLL data with 15% missing was run with initial estimates from

MEAN imputation. The PLS2 imputation error curve using poor initial estimates is quite similar to the PLS2 error curve using good initial estimates from KNN imputation (see http://stat.tamu.edu/∼dnguyen/supplemental.html). However, for $y \approx \mu$, PLS2 estimates with mean initial values are not as good. Thus, a KNN-PLS2 is the preferred strategy.

### 5.5 More complex missing data mechanisms

There are various missing data mechanisms, including missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). For microarray data, an example of MAR is when the missing data probability is a function of an observe covariate, $Z$, such as background noise. Missing not at random is the most complicated situation. In this situation the missingness depends on the expression intensity values ($Y$), that are *not observed*. It is well known that MNAR would cause non-identifiability problems. Issues become complex due to the high dimension in microarray data. Consequently, not assuming the missingness pattern depends on $Y$ to avoid non-identifiability induced by MNAR leads to the MCAR scenario. There are complex statistical issues associated with MNAR and the interested reader is referred to missing data imputation literature.

The evaluation procedure and methods proposed here and in Troyanskaya *et al.* (2001) provide a simple study model containing the essential issues and provide a framework upon which more complex studies can be built. For example, it may be of interest to see how the current study model performs when the MCAR assumption is violated. To study this, an experiment was conducted when the missingness rate was allowed to depend on the expression level in a uniform fashion; that is, the missingness rate does not vary from gene to gene beyond depending on the gene's expression level. Precisely, the missing rate in the low expression range ($y < \mu - 2\sigma$) was set to be twice that of the remaining expression range ($y > \mu - 2\sigma$). Table 3C gives the results for this missing not at random scenario (see also Supplemental Figure 7 and 8). For illustration, this was applied separately to the Cy5- and Cy3-channel of the BCLL data set. As can be seen the results are quite similar to the MCAR case (Table 3B and Supplemental Figures 7 and 8).

## 6. Conclusions

We have provided an evaluation of KNN (and MEAN) imputation by examining the accuracy across the full range of observed gene expression values. Our findings suggest that KNN imputation, although a very simple method, performed well on average for microarray data sets used. This is consistent with

previously reported results on yeast cDNA arrays (Troyanskaya *et al.*, 2001). However, the accuracy is high mostly near the center of the distribution of true expression values (e.g., within one standard deviation of the mean). The relative accuracy of KNN can be improved outside this moderate range of true expression values. Among the regression methods proposed, PLS2 provided the most gain in accuracy outside this moderate range.

Nonetheless, the methodological simplicity and modest computational cost are some appealing aspects of the KNN imputation. The extensive study of the neighbors, $K$, described earlier suggests some general guidelines to consider when choosing the number of neighbors for KNN imputation: (1) $K \lesssim 6$ provides poor accuracy near the center of the distribution of true values (2) good accuracy is achieved for $K$ between 10 and 22 and (3) although the moderate range of expression values is less sensitive to the choice of $K$, the accuracy deteriorates rapidly for KNN imputation with a large number of neighbors ($K \gtrsim 30$).

PLS imputation incorporates global information on all candidate genes as well as the target gene expression values in the training data set for predicting the missing values. Accuracy for PLS imputation is higher for some ranges beyond moderate expression. Gene expression beyond moderate expression may be of interest when searching for differentially expressed genes.

We have focused on the missing value estimation methods themselves and on the evaluation of estimation accuracy as a function of the expression level. Although beyond the scope of this work, it would be of interest to examine the performance of some downstream analyses in the context of data imputation. For example, to what extent will microarray-based cancer classification/prediction analysis differ based on using only available data, ignoring missing data (genes), and imputing missing data? These are issues that are worth investigating further.

## Acknowledgments

Our research was supported by National Cancer Institute grants CA90301, CA57030, and CA74552, and by the National Institute of Environmental Health Sciences (P30-ES09106).

## References

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Broldrick, J. C., Sabet, H. Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J. Jr., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O. and Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403**, 503-511.

Dudoit, S., Fridlyand, J. and Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* **97**, 77-87.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D. and Lander, E.S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286**, 531-537.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O., Wilfond, B., Borg, A. and Trent, J. (2001). "Gene-expression profiles in hereditary breast cancer," *The New England Journal of Medicine* **344**, 539-548.

Helland, I. S. (1988). On the structure of partial least squares. *Communications in Statistics–Simulation and Computation* **17**, 581-607.

Höskuldsson, A. (1988). PLS regression methods. *Journal of Chemometrics* **2**, 211-228.

Nguyen, D. V. and Rocke, D. M. (2002a). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics* **18**, 39-50.

Nguyen, D. V. and Rocke, D. M. (2002b). Multi-class cancer classification via partial least squares with gene expression profiles. *Bioinformatics* **18**, 1216-1226.

Nguyen, D. V. and Rocke, D. M. (2002c). Classification of acute leukemia based on DNA microarray gene expressions using partial least squares. In *Methods of Microarray Data Analysis*, eds. Lin, S. M. and Johnson, K. F. Kluwer, Dordrecht, 109-124.

Nguyen, D. V., Arpat, A. B., Wang, N. and Carroll, R. J. DNA microarray experiments: Biological and technological aspects. *Biometrics* **58**, 701-717.

PROC PLS, SAS Institute (1999).

Rubin, D. B. (1976). Inference and missing data. *Biometrika* **63**, 581-592.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P. O., Hastie, T., Tibshirani, R., Botstein, D. and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics* **17**, 520-525.

Wold, S. (1994). PLS for multivariate linear modeling. In *Chemometric Methods in Molecular Design*, (Edited by van de Vaterbeemd, H.) Verlag-Chemie, Weinheim, 195-218.

Wold, S., Ruhe, A., Wold, H. and Dunn, W. J. (1984). The collinearity problem in linear regression: the partial least squares (PLS) approach to generalized inverses. *SIAM Journal of Scientific and Statistical Computing* **5**, 735-743

Received May 7, 2002; accepted August 6, 2003.

Danh V. Nguyen
Division of Biostatistics and Preventive Medicine
One Shields Avenue, TB 168
University of California
Davis, CA, 95616 USA
ucdnguyen@ucdavis.edu

Naisyin Wang
Department of Statistics
Texas A&M University
TAMU 3143
College Station, TX, 77843-3143 USA
nwang@stat.tamu.edu

Raymond J. Carroll
Department of Statistics
Texas A&M University
TAMU 3143
College Station, TX, 77843-3143 USA
carroll@stat.tamu.edu