

Microarray Missing Values Imputation Methods: Critical Analysis Review

Mou'ath Hourani¹ and Ibrahiem M. M. El Emary²

¹ Faculty of Information Technology, Al Ahliyya Amman University,
Al Saro St, Amman 19328 Jordan
mouath.hourani@yahoo.com,

² Faculty of Engineering, Al Ahliyya Amman University,
Al Saro St, Amman 19328 Jordan
omary57@hotmail.com

Abstract. Gene expression data often contain missing expression values. For the purpose of conducting an effective clustering analysis and since many algorithms for gene expression data analysis require a complete matrix of gene array values, choosing the most effective missing value estimation method is necessary. In this paper, the most commonly used imputation methods from literature are critically reviewed and analyzed to explain the proper use, weakness and point the observations on each published method. From the conducted analysis, we conclude that the Local Least Square (LLS) and Support Vector Regression (SVR) algorithms have achieved the best performances. SVR can be considered as a complement algorithm for LLS especially when applied to noisy data. However, both algorithms suffer from some deficiencies presented in choosing the value of Number of Selected Genes (K) and the appropriate kernel function. To overcome these drawbacks, the need for new method that automatically chooses the parameters of the function and it also has an appropriate computational complexity is imperative.

Keywords: Completely at random (MCAR), Missing At Random (MAR), Sequential K-Nearest Neighbors (SKNN), Gene Ontology (GO), Singular Value Decomposition (SVD), Least Squares Imputation (LSI), Local Least Square Imputation (LLSI), Bayesian Principal Component Analysis (BPCA) and Fixed Rank Approximation Method (FRAA).

1. Introduction

The presence of missing values is a common problem for the analysis of microarray data. Typically, in an ordinary microarray, 1-10% of the data entries are missing [1], affecting up to 95% of the genes. There are different reasons for missing expression values [2]. The microarray may contain so-called "weak spots". Usually these spots are filtered out. After comparing the pixels of the spot with the pixels of the background, if the fraction of spot

pixels is greater than the median of the background pixels and is less than a given threshold, the gene expression that corresponds to this spot will be set as missing. Another reason for missing expression values is the occurrence of technical errors during the hybridization. Moreover, if fluorescent intensity of a given spot is below a certain threshold, the value of that spot will be defined as missing. A third reason for missing values is the presence of dust, scratches, and systematic errors on the slides [1]. Thus, the resulting data matrix will most likely contain missing values which may disturb the gene clustering obtained by the classical clustering methods, e.g., projection methods like PCA. As well known, many algorithms for gene expression analysis require a complete matrix of gene array values as input. For example, hierarchical clustering method is not robust to missing data, and may lose effectiveness even with a few missing values. To limit the effects of missing values in the clustering analysis, different strategies have been proposed: (i) the genes containing missing values are removed, (ii) the missing values are replaced by a constant (usually zero, or one), or (iii) the missing values are re-estimated on the basis of the whole gene expression data. Different estimation techniques have been applied to missing values in microarray data. The *K*-nearest neighbours approach (KNN), Local Least square (LLS) and Support Vector Regression (SVR) are among the most reliable and efficient methods.

Recently comparative studies of three data imputation methods; a singular value decomposition based method, weighted *K*-nearest neighbours, and row average were presented in Troyanskaya *et al.* (2001) [3]. Also, Bo *et al.* (2004) [4] compared methods that utilize correlations between both genes and arrays based on the least square principle and the method of *K*-nearest neighbours. Ouyang *et al.* (2004) [5] proposed an imputation method based on Gaussian mixture clustering and model averaging. There are several alternative ways of dealing with missing data, and this paper aim is to review and critically discuss the most common used methods for missing values estimation. For the purpose of the application, the imputation approaches will be discussed in the context of microarray data imputation. This paper begins with explaining the nature of missing data. Then it discusses general rules for dealing with missing values and finally in the subsequent sections it reviews and analyse the most common used microarray data imputation approaches as well as presenting some of major experimental results and discussion.

1.1. The Nature of Missing Data

An empirical examination of the patterns of missing data is required to determine whether it is distributed randomly across the cases and the variables. Two patterns are possible: missing completely at random or missing at random

Completely At Random (MCAR). Data are missing completely at random when the probability of obtaining a particular pattern of missing data is not dependant on the values that are missing and when the probability of obtaining the missing data pattern in the sample is not dependant on the observed data [6].

Missing At Random (MAR). Often data are not missing completely at random, but they may be classifiable as missing at random (MAR). MAR is a condition which exists when missing values are not randomly distributed across all observations but are randomly distributed within one or more subsamples (e.g., missing more among samples from diseased persons than non diseased, but random within each sample) [6]. In practical terms, it is quite difficult to determine if the data are MAR or MCAR. When a single variable contains missing data, it is not too difficult to determine if any of the other variables in the data set predicts whether there is missing data on a particular variable. In practice, however, data will be missing on a number of variables, and so determining if other variables are related may be considerably complex, and this is particularly the case with hierarchically structured data and individuals are missing entirely from specific groups. Some statistical programs have techniques specifically designed for missing data analysis (e.g., Missing Value analysis in SPSS statistical software package [27]), which generally include one or both diagnostic tests [6]. As a result of these tests, the missing data process is classified as either MAR or MCAR, which then determines the appropriate types of potential analysis processes. In microarray data analysis, MCAR is the most common available type and a wide range of potential techniques are suited for it [7].

1.2. Dealing with Missing Values

As mentioned before, different alternatives are available to deal with missing values, in addition to those mentioned before; next we categorize these methods into three general classes [6]:

Eliminate Data Objects or Attributes. A simple and effective strategy is to eliminate objects with missing values. However, even a partially specified data object contains some information, and if many objects have missing values, then a reliable analysis can be difficult or impossible to be obtained [6].

Estimate Missing Values. Sometimes missing data can be reliably estimated. For example, consider a time series that changes in a reasonably smooth fashion, but has a few, widely scattered missing values. In such cases, the missing values can be estimated by using the remaining values.

Ignore the Missing Values During Analysis. Many data mining approaches can be modified to ignore missing values. For example, suppose that objects are being clustered on the similarity between pairs of data objects needs to be calculated. If one or both objects of a pair have missing values for some attributes, then the similarity can be calculated by using only the attributes that do not have missing values. Likewise, many classification

schemes can be modified to work with missing values. The accuracy of this approach depends on the total number of the missing values in the microarray. Estimation of missing data is a well-studied problem in the statistical literature and imputation methods have traditionally been used in "several data analysis applications [3],[7]". Recently, such methods have been reinvented and extensively applied to the imputation of microarray data [3]. Details of missing values methods handling in microarrays will be fully discussed in the subsequent sections.

2. Missing Values Imputation Methods

Based on the review of the missing values imputation methods literature and its issues, few different missing values imputation algorithms are presented in the context of microarray data analysis. In order to obtain an in-depth understanding of the current research in missing values imputation methods, the following sections investigate each of these methods and explore the performance of each one and the differences between them.

2.1. Weighted K-Nearest Neighbors (KNNimpute)

KNNimpute is a standard missing value imputation method introduced by Troyanskaya et al. (2001) [3]. The KNN-based method takes advantage of the correlation structure in microarray data by selecting genes with expression profiles similar to the gene of interest to impute missing values. Accordingly, the imputation process is typically divided into two steps. In the first step, a set of genes nearest to the gene with a missing value is selected. To explain the way that this step works, consider gene g in experiment i so, let's say $V_{g,i}$ is missing value, thus, this method would find k other genes, with a known value for experiment i , and with the expression profile most similar to g considering *all* the experiments. The authors examined a number of metrics for gene similarity (Pearson correlation, Euclidean distance, variance minimization). In spite of its sensitivity to outliers which could be present in microarray data, Euclidean distance was found to be a sufficiently accurate norm. The reason behind this finding lies in using the log-transform to normalize the data, what in turn reduces the effect of outliers on gene similarity determination.

The second step involves the prediction of the missing value using the observed values of the selected genes. At this stage, a weighted average of values in experiment i from the k closest genes is then used as an estimate for the missing value in gene g . In the weighted average, the contribution of each gene is weighted by the similarity of its expression to that of gene g by using the following equation:

$$W_i = \frac{1/D_i}{\sum_{i=1}^k 1/D_i} \quad (1)$$

Where k is the number of selected genes and D_i is the distance between the i -th gene and the gene to be imputed.

Evaluation Methods and Results. KNNimpute was evaluated over different data sets and over different values of k . Two of the data sets were time-series data, and one contained a non-time series subset of experiments. One of the time-series data set had less apparent noise than the other. The missing values in the original expression profile were removed, yielding complete data sets. Then, between 1% and 20% of the data was deleted randomly to create the test sets, and each method was used to recover the introduced missing values. The accuracy of imputation method was evaluated by calculating the error between actual values and imputed values after missing values were estimated. The metric used to assess the accuracy of estimation was the root mean square error (RMSE). RMSE was calculated as follows,

$$RMS_{error} = \sqrt{\frac{1}{N} \sum_{i=1}^N (R_i - I_i)^2} \quad (2)$$

Where R_i is the real value, I_i is the imputed value, and N is the number of missing values. This gives small values for the method that best minimizes the squared errors between the estimated values and the real values. The imputation method achieving the smallest RMSE gives the most correct picture of the complete data matrix when estimated values are included. From the results obtained, the authors concluded that the KNNimpute is very accurate, with the estimated values showing only 6-26% average deviation from the true values, depending on the type of data and fraction of values missing. The algorithm is also robust to the increase in the proportion of values missing, with a maximum of 10% decrease in accuracy with 20% of the data missing. In addition, the method is relatively insensitive to the exact value of K within the range of 10-20 neighbours. Performance declines when a lower number of neighbours are used for estimation, primarily due to overemphasis of a few dominant expression patterns. However, when the same gene is present twice on the arrays, the method appropriately gives a very strong weight to that gene in the estimation. KNNimpute can accurately estimate data for matrices with as low as six columns [3]. However, it is not reasonable to use this method on matrices with less than four columns [3]. In principle KNN imputation works much better than the other traditional methods (i.e. row average, median average) but it requires to have enough complete patterns (patterns with no missing values) in the data set to be confident of finding the correct neighbors of the patterns with missing values. It also requires enough existing values in the patterns with missing values in

order to be able to determine their neighbors. The estimation ability of these advanced methods depends on important model parameters, such as the K -value in KNNimpute. At present, there is no known theoretical way, however, to determine these parameters appropriately.

2.2. Enhanced KNNimpute Method

Sequential K-Nearest Neighbor (SKNN) Methods. Sequential k -nearest neighbor (SKNN) method is a cluster-based method that uses the imputed missing values in a later imputation. SKNN method differs from traditional KNNimpute in that it imputes the missing values sequentially from the gene having the least missing values, and uses the imputed values for the subsequent imputations. After separating the data set into complete and incomplete sets, all missing values in a gene are filled with the weighted mean value of the corresponding column of the nearest neighbor genes in the complete set. Once all missing values of a gene are imputed, the imputed gene is moved into the complete set and used for the imputation of the rest of genes in the incomplete set [8]. The data sets used in this work were selected from a study of gene expression in yeast *Saccharomyces cerevisiae* cell-cycle regulation [9], calcineurin/crz1p signaling pathway [10] and certain environmental changes [11]. These data sets were classified into time series data set, mixed (time-series and nontime series) data set and non-time series data set respectively [8]. The efficiency of this algorithm is greatly improved in its accuracy and computational complexity over the traditional KNN-based method. The performance of SKNN was higher than KNNimpute method for the data with high missing rates and large number of experiments [8].

The GO-based KNNimpute Method. Gene ontology (GO) is a structured network of defined terms which describe gene product attributes [12]. The goal of the gene ontology is to produce a dynamic, controlled vocabulary that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing [12]. To this end, three independent ontologies accessible on the World-Wide Web (<http://www.geneontology.org>) are being constructed: biological process, molecular function and cellular component. The GO-based KNNimpute method uses the gene semantic similarity that originated from gene ontology annotations to improve the performance of KNNimpute method. The semantic dissimilarity is external information on the functional similarity of two genes that is used to select the relevant genes for missing value imputation. The relative contribution of each information source was automatically estimated from the data using adaptive weight value estimation. Using the gene ontology files downloaded from the GO web site, the ontology tree is created. The semantic dissimilarity between two genes $g1$ and $g2$ is calculated using the created ontology tree and is used as a correlation reference between those two genes.

The results obtained enhanced the performance of KNNimpute algorithm considerably, especially when the number of experimental conditions is small and the percent age of missing values is high. Consequently, gene ontology

method is a complementary method for KNNimpute algorithm better suited for small number of experiments [13]. However, more research is needed to check the validity of this method on different missing values imputation methods to either complement the algorithm or improve the performance of that algorithm.

2.3. Singular Value Decomposition (SVD)-Based Method

Singular value decomposition is used to obtain a set of mutually orthogonal expression patterns that can be linearly combined to approximate the expression of all genes in the data set [3]. SVDimpute is studied and implemented in the context of microarray data also by Troyanskaya et al. (2001) [3]. To explain the operation of this method, assume that there are m samples in matrix data Y . Let t be the number of missing entries in a row R , $1 \leq t \leq n$; assume the missing entries are in columns s_1, \dots, s_t . Let B be the complete rows of Y . In Singular Value Decomposition the $m \times n$ matrix, $m > n$, is expressed as the product of three matrices:

$$Y = U \Sigma V^T \quad (3)$$

where the $m \times m$ matrix U and the $n \times n$ matrix V are orthogonal matrices, and Σ is an $m \times n$ that contains all zeros excepts for the diagonal $\sum_{i,i}, i = 1, \dots, n$. Those diagonal elements are rank ordered ($\sum_{i,i} \geq \dots \geq \sum_{n,n} \geq 0$) square roots of the eigenvalues of YY^T [14]. Alter et al. (2000) [14] and Holter et al. (2000) [15] showed that several significant eigengenes are sufficient to describe most of the expression data, but the exact number (K) of significant eigengenes better for the estimation needs to be determined empirically by evaluating performance of SVDimpute algorithm while varying K .

Let R_1, \dots, R_k be the first K rows of V^T , and let R be a row of Y with the first t entries missing. The estimation procedure of SVDimpute performs a linear regression of the last $n - t$ column of R against the last $n - t$ columns of R_1, \dots, R_k . Let c_k be the regression coefficients. Then the missing entries of R are estimated by:

$$R^{(j)} = \sum_{k=1}^K R_k^{(j)} \cdot "j=1,2,\dots,t" \quad (4)$$

SVDimpute first performs SVD on B , then it uses the estimation procedure on each incomplete row of Y . Let \hat{Y} be the imputed matrix. SVDimpute repeatedly performs SVD on \hat{Y} by the estimation procedure, until the root mean squared error between two consecutive \hat{Y} 's falls below 0.01 [3].

SVDimpute was tested under the same data sets used by KNNimpute. SVDimpute estimation provided considerably higher accuracy than row average on all data sets, but SVDimpute yielded best results on time-series data with low noise levels. The increasing proportion of missing entries deteriorated the performance of SVDimpute algorithm sharply. Finally, SVDimpute was very sensitive to the exact parameters used (number of nearest neighbors K) with a sharp deterioration in performance for non-optimal fraction of missing values [3].

2.4. Least Squares Imputation - Based Methods

Least Squares Imputation (LSimpute)

LSimpute is a regression-based estimation method that exploits the correlation between genes. To estimate the missing value $V_{g,i}$ from gene expression matrix Y , the k -most correlated genes are first selected, considering all samples except i , and containing non-missing values for gene g . The LS regression method then estimates the missing value $V_{g,i}$. By having the flexibility to adjust the number of predictor genes k in the regression, LSimpute performs better when data have a strong local correlation structure [4]. From the discussion preceded, we observe that LSimpute works similarly to KNNimpute method, but instead of using equation (1) to impute the missing values it uses the Least Square regression method. To estimate the performance of the LSimpute algorithm, root mean square deviation (RMSD) was used:

$$\text{RMSD} = \sqrt{1/N \sum_{i=1}^N |R_i - I_i|^2} \quad (5)$$

Where R_i is the real value, I_i is the imputed value, and N is the number of missing values. In contrast to KNNimpute which uses Euclidean distance to measure the correlation, LSimpute method considers the negative correlation between genes in estimation model as well as positively correlated genes. To test the LSimpute algorithm, three data sets were selected [4]. Two cancer studies and one time series study [4]. One data set came from the NCI60 study [16]. The second data set came from a lymphoma study [17]. The third data set was from an infection time series study [18]. The LSimpute is demonstrated to perform better than KNNimpute on three example data sets with 5-25% of the data missing. Furthermore, The results obtained on data sets with 10% missing values reveals an RMSD between missing value estimates and the real values that is 15-20% smaller than that obtained using KNNimpute.

Local Least Squares Imputation (LLSimpute)

The LLSimpute algorithm uses the KNN process to select the most correlated genes and then predicts the missing value ($V_{g,i}$) using the least squares formulation for the neighbourhood gene and the non-missing entries w_1 of g_1 ; where w_1 represents the vector of non-missing entries for gene vector g_1 [19]. Specifically, the local least squares imputation first chooses K nearest neighbouring genes using the distance measure defined in the above section (K to be determined). These genes are regarded as coherent genes to the target gene. The missing values in these coherent genes are filled with their respective row averages. Then, based on these K neighbouring gene vectors, matrices A and B and a vector w are formed. If the missing value is to be estimated by the K -most correlated genes, each element of the matrix A and B , and a vector w are constructed as:

$$\begin{pmatrix} g_1 \\ g_{s1} \\ \dots \\ g_{sk} \end{pmatrix} = \begin{pmatrix} \alpha_1 & w_1 & w_2 & \dots & w_i & \alpha_2 \\ B_{11} & A_{11} & A_{12} & \dots & A_{1i} & B_{12} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ B_{k1} & A_{k1} & A_{k2} & \dots & A_{ki} & B_{k2} \end{pmatrix}$$

Where i is the number of experiments, α_1 and α_2 are the missing values and g_{s1}, \dots, g_{sk} are the k genes that are most similar to g . Then LLS proceeds to compute a K -dimensional coefficient vector x such that the square

$$|A^T x - w|^2 = (A^T x - w)^T (A^T x - w) \text{ is minimized, that is,}$$

$$\min |A^T x - w|^2 \tag{6}$$

Let x denote the vector such that the square is minimized, that is,

$$w \cong x_1 \alpha_1 + x_2 \alpha_2 + \dots + x_k \alpha_k \tag{7}$$

Where x_i are the coefficients of the linear combination, found from the least squares formulation (3.5). And, the missing values in g can be estimated by

$$\alpha_1 = B_{11} + B_{21}x_2 + \dots + B_{k1}x_k \tag{8}$$

$$\alpha_2 = B_{12} + B_{22}x_2 + \dots + B_{k2}x_k \tag{9}$$

Where α_1 and α_2 are the first and the second missing values in the target gene. Thus, for estimating the missing values of each gene, we need to build the matrices A and B and a vector w , and solve the system for all missing

values. In addition to the human colorectal cancer (CRC) data set [19], LLSimpute was tested under the same data sets used by KNNimpute. The performance of the missing value estimation was evaluated by the normalized root mean square (NRMSE):

$$NRMSE = \sqrt{\frac{1/N \sum_{i=1}^N (R_i - I_i)^2}{\text{variance}[R_i]}} \quad (10)$$

Where the R_i is the real value and I_i is the imputed value. The mean and the variance were calculated over missing entries in the whole matrix. The LLSimpute method takes advantage of the local similarity structures in addition to the optimization process by the least squares. In the results presented, LLSimpute outperformed BPCA (discussed in the next section) as well as KNNimpute when K is large; and it also outperformed the LSimpute method. The results showed that LLSimpute was also less sensitive to the noise level, being considered as an accurate reference method to compare with [19].

2.5. Bayesian Principal Component Analysis (BPCA)

Bayesian Principal Component Analysis (BPCA) is an estimation method that uses the probabilistic Bayesian theory to impute the missing values [2]. The entire data set of gene expression profiles is represented by a Y expression matrix. BPCA divides the data set into two sets (complete and non-complete). Then BPCA estimates missing values Y^{miss} in data matrix Y using those genes Y^{obs} having no missing values. The probabilistic PCA (PPCA) is calculated using Bayes' theorem and the Bayesian estimation calculates posterior distribution of model parameter θ and input matrix X containing gene expression samples using:

$$p(\theta, X | Y) \propto p(X, Y | \theta)p(\theta) \quad (21)$$

where $p(\theta)$ is called as the prior distribution which denotes a *priori* preference to θ and X . Missing values are estimated using a Bayesian estimation algorithm, which is executed for both θ and Y^{miss} and calculates distributions for θ and Y^{miss} , $q(\theta)$ and $q(Y^{miss})$ [2] using:

$$q(Y^{miss}) = p(Y^{miss} | Y^{obs}, \theta_{true}) \quad (32)$$

Where θ_{true} is the posterior of the missing value. Finally, the missing values in the gene expression matrix are imputed using:

$$Y^{imiss} = \int Y^{miss} q(Y^{miss}) dY^{miss} \quad (43)$$

BPCA takes advantage of the global correlation in the data sets, and thus, has the advantage of prediction speed incurring a computational complexity, which is one degree less than for both KNN and LSImpute [2]. For imputation purposes, however, improved estimation accuracy is always a greater priority than speed. Four test data sets taken from yeast cell-cycle [20] and human colorectal cancer clinical (CRC) [21] were prepared. Two methods were used to introduce the artificial missing entries: Rate-based way and Histogram-based way. In rate-based way, the entries were selected randomly in a specific percentage. Whereas, in Histogram-based way, the column-wise number of missing entries from original expression matrix were obtained. Then the corresponding entries of the artificial expression were removed. The performance of the missing value estimation was evaluated by NRMSE (Equation 10).

The method was evaluated by comparing it to KNNimpute and SVDimpute using various microarray data sets. The results obtained using this method showed marked improvement in estimation performance. From the experiments conducted, the K value can be determined automatically without a priori knowledge on the data set. Therefore, in BPCA the value of K can be estimated as $K = D - 1$ for every data set, where D is the number of samples. BPCA produced better results than KNNimpute or SVDimpute at the optimal K -value for each method. However, if the genes have dominant local similarity structures, the KNNimpute performs better than BPCA, as BPCA assumes the missing values in an expression matrix occur randomly and independently of other features in the matrix. Furthermore, normalization for the expression matrix before the missing value estimation process is not suggested when using BPCA, because their results showed that row-wise or column-wise normalization degraded the missing value estimation ability.

2.6. Fixed Rank Approximation Method

Fixed Rank Approximation Method (FRAA) is introduced by Friedland et al. (2005) [22]. FRAA uses an optimization algorithm in which the estimation of missing entries is done simultaneously, i.e., the estimation of one missing entry influences the estimation of the other missing entries [22]. If the gene expression matrix Y has missing data, the algorithm completes its entries to obtain a matrix Y' , such that the rank of Y' is equal to (or does not exceed) d , where d the number of significant singular values of Y [22]. Solving this problem requires an optimization algorithm for finding Y' using the techniques for inverse eigenvalue. FRAA is a global method, which finds the optimal values of the missing entries such that the obtained Y' minimizes the object function $f_l(X)$. Here $f_l(X)$ is the sum of the squares of all but the first l singular values of an $n \times m$ matrix X . The minimum of $f_l(X)$ is considered on the set S , which is the set of all possible choices of matrices $X = (x_{ij})$, such that $(x_{ij}) = g_{ij}$ if the entry g_{ij} is known. The completion matrix is computed iteratively, by a local minimization of $f_l(X)$ on S [22]. FRAA is a robust algorithm [19]. However,

it could not outperform KNNimpute even though it is more accurate than replacing missing values with 0's or with row means [22].

2.7. Gaussian Mixture Clustering Method

Gaussian mixture clustering is a partitioning clustering technique that estimates probability density functions (PDF) for each class, and then performs classification based on an expectation-maximization (EM) algorithm [5]. In statistical computing, an expectation-maximization (EM) algorithm is an algorithm for finding maximum likelihood estimates of parameters in probabilistic models. Ouyang et al.(2004) used Gaussian mixture clustering principle to introduce the GMCimpute missing imputation algorithm. In this method, the data are modeled by Gaussian mixtures, and missing entries are estimated by the expectation maximization algorithm [5]. GMC takes the approach of model averaging. The microarray data are clustered into K-component Gaussian mixtures by the classification expectation maximization algorithm. Then, the missing values are estimated by the expectation maximization algorithm as the arithmetic mean of the K estimates [5].

To explain the GMCimpute method, we start explaining the Gaussian mixture clustering step. In mixture clustering, the number of clusters K must be specified in advance. This can be satisfied by any statistical test (e.g., statistic B and Gap statistic [23]). Then, the mixtures are initialized by partitioning the data set into K subsets. The initial Gaussian mean μ_i , $i = 1, \dots, G$ (where G is the number of Gaussians) for mixture clusters can be determined by using K-means clustering with the Euclidean distance. Third, the covariance matrices, V_i are initialized as the distance to the nearest clusters. Fourth, initializing the weights $\pi = 1/G$ so that all Gaussians are equally likely. Each cluster K produced is mathematically represented as a weights sum of Gaussians:

$$p(X | \theta) = \sum_{i=1}^G \pi_i p(X | G_i) \quad (54)$$

Where G is the number of Gaussians, the π 's are the weights. In a Gaussian mixture, each cluster is modeled by a multivariate normal distribution. The parameters of component K comprise the mean vector μ_i and the covariance matrix V_k , and the probability density function is:

$$G_k = \frac{1}{(2\pi)^{n/2} |V_i|^{1/2}} \times e^{-1/2(X - \mu_i)^T V_i^{-1}(X - \mu_i)} \quad (65)$$

Where μ_i is the mean of the Gaussian and V_i is the covariance matrix of the Gaussian. The second step uses the iterative Classification Expectation Maximization algorithm (CEM) to maximize the likelihood of the mixture. There

are three steps in CEM. In the maximization step, μ_i , V_i and τ_{ip} are estimated from the partition:

$$\tau_{ip} = P(G_i | X) = \frac{\pi_i P(X | G_i, C_k)}{P(X)} = \frac{\pi_i P(X | G_i, C_k)}{\sum_{j=1}^G \pi_j P(X | \theta_j, C_k)} \quad (76)$$

GMCimpute method computes the τ_{ip} which is defined as probability of cluster- l given X , equal to the formula. The denominator is the probability of X to be in the class, and the numerator is the PDF of cluster- l multiplied by its weight. After we calculate τ_{ip} , use it to estimate new weights, means and covariance. And then, we use the new mean, weight, and covariance to estimate new τ . Iteratively update the weights, means and covariances:

$$\pi_i(t+1) = \frac{1}{N_C} \sum_{p=1}^{N_c} \tau_{ip}(t) \quad (87)$$

$$\mu_i(t+1) = \frac{1}{N_C \pi_i(t)} \sum_{p=1}^{N_c} \tau_{ip}(t) X_p \quad (98)$$

$$V_i(t+1) = \frac{1}{N_C \pi_i(t)} \sum_{p=1}^{N_c} \tau_{ip}(t) ((X_p - \mu_i(t))(X_p - \mu_i(t))^T) \quad (109)$$

Then, GMCimpute method recomputes τ_{ip} using the new weights, means and covariances. Stop training if

$$\Delta \tau_{ip} = \tau_{ip}(t+1) - \tau_{ip}(t) \leq \text{threshold} \quad (20)$$

Otherwise, continue the iterative updates. The GMCimpute estimated the missing entries row by row by applying equation. 20 until the convergence occur. Two data sets were chosen to test the algorithm: the first form the yeast cell cycle data and the yeast environment stress time series data [5]. RMSE metric (Equation 2) was used to evaluate the performance of the algorithm. GMCimpute, KNNimpute and SVDimpute were tested and compared on the two data sets. From the results obtained, GMCimpute was the best among the three methods for both data sets, and SVDimpute was better than KNNimpute on cell cycle data, and SVDimpute is better than SVDimpute on stress time series data.

2.8. Support Vector Regression

The SVR is a nonlinear algorithm which characterizes the properties of learning machines that enable the algorithm to generalize well to unseen or

missing data [24]. To explain the basic idea behind this algorithm, suppose we are given some training data $f(x_1, y_1), \dots, (x_i, y_i) \subset X \times \mathbf{R}$ where X denotes the space of the input patterns. The goal of SV regression is to find a function that has at most ε deviation from the actually obtained targets y_i for all the training data. In other words, we do not care about errors as long as they are less than ε , but will not accept any deviation larger than this. In missing value imputation, this fact assures that the difference between the predicted value and the actual value will be in the ε range [24]. The SVR algorithm consists of two steps: first, the SVR uses kernel function to transform the samples from the input space into a higher dimension space. Then SVR searches for the global optimal solution to the corresponding problem using the quadratic programming by finding the corresponding support vectors [24]. The performance of SVR algorithm is highly dependent on the type of the kernel function used. Furthermore, there is a computational complexity introduced in this algorithm when solving the missing value (finding the b value). To find the value of each missing point, optimization techniques are applied.

Six data sets were used to evaluate the performance of the support vector regression (SVR) and orthogonal coding scheme. The first two data sets focus on identification of the cell-cycle regulated genes in yeast *Saccharomyces cerevisiae*, and are all time series data sets [20]. The third data set is from Gaschs experiments [11] focusing on the response to the environment changes of genes in yeast. The fourth data set is original cDNA microarray data relevant to human colorectal cancer (CRC) [21]. The fifth data set is a gene expression data set relevant to the molecular pharmacology of cancer, which contains gene expression profiles in 60 human cancer cell lines in a drug discovery screen [25]. The last data set is the same data set used in Kim et al. (2005) [19] focusing on the cell-cycle-regulated genes. The performance of this algorithm was measured by using the NRMSE (Equation 10). The performance of the SVR has been compared with three imputing approaches: KNN, BPCA and LLS impute methods. When the SVR applied on the noisy data set, which was classified a challenging data set by Troyanskaya [3], and when the percentage of missing values in the data set was below 20%, the SVR achieved its best results. When the percentage of missing values reached 20%, the NRMSE of the SVR was a little higher than those of the BPCA and the LLS impute methods, and still much better than that of the KNN impute method. Furthermore, SVR was tested on the other data sets and the results were nearly similar to the other methods.

2.9. Collateral Missing Value Imputation

The collateral missing value estimation (CMVE) algorithm is introduced by Sehgal et al. (2005) [26]. The authors presented this algorithm based on the novel concept of multiple imputations that uses linear programming to optimize the missing value parameters. CMVE consists of three steps: first,

the algorithm locates the missing value position Y_{ij} . Then, it uses the CoV equation to compute the absolute covariance of expression vector v of gene l using the following equation:

$$CoV = \frac{1}{n-1} \sum_{i=1}^n (v_i - v')(w_i - w') \quad (21)$$

Where w is the predictor gene factor and v the expression vector of gene g which has the missing values. The algorithm then ranks the (rows) based on the CoV and select the most effective rows R_k . The R_k values obtained are used to estimate Φ_1 by using the following equation:

$$\Phi_1 = \alpha + \beta X + \xi \quad (22)$$

Where ξ is the error term that minimizes the variance in the LS model (parameters α and β). For a single regression, the estimate of α and β are, respectively,

$$\alpha = y' - \beta X \text{ and } \beta = \frac{\xi_{xy}}{\xi_{xx}}$$

Where $\xi_{xy} = \frac{1}{n-1} \sum_{j=1}^n (X_j - X')(Y_j - Y')$ is the empirical covariance between X and Y , Y_j is the gene with the missing value and X_j is the predictor gene in R_k .

where $\xi_{xx} = \frac{1}{n-1} \sum_{j=1}^n (X_j - X')^2$ is the empirical covariance of X with X'

and Y' being the respective means over X_1, \dots, X_n and Y_1, \dots, Y_n , so the LS estimate of Y given X is expressed as:

$$\hat{Y} = Y' - \frac{\xi_{xy}}{\xi_{xx}} (X - X')^2. \quad (23)$$

Finally, the same process is used to calculate the values of Φ_2 and Φ_3 and the algorithm then will seek the next missing value and repeat the entire process again. Four different types of microarray data were used including both time series and non-time series data. Two data sets were obtained from sporadic mutations of ovarian cancer data (non-time series) and the other two were from yeast sporulation data (time series) [26]. The performance of the algorithm was compared with three other methods: KNN, BPCA and LSImpute. The NRMSE (equation 10) metric was used to evaluate the estimation performance of each technique. In term of accuracy and robustness, this algorithm outperforms a wide range of randomly introduced missing values [26]. More experiments are required to compare this algorithm with other highly performed algorithms (e.g., LLS). Moreover, the CMVE algorithm should be tested on noisy data.

3. Data Sets and Experimental Results

In this section, we evaluate the performance of each imputation technique to predict the missing values using cDNA microarray. Experiments data sets as follows:-

Niehs is the first data set which is based on a study of human cell lines. This data is structured from three swaps, thus we have six arrays. In [18], the data from the Niehs experiments comparing treated and control human cell lines. There are 1907 genes in the Niehs data set and there are no missing values. Accordingly, there is a full intensity data matrix of dimension 1,907x12.

Gene expression data from the study of Schizophrenia disease is the second example. In this example, the data set is taken from Bowden et al (2005) [29], it has been generated in Newcastle university, Australia. It is composed of 14 nonpsychiatric control individuals and 14 patients diagnosed with schizophrenia, matched in age and gender. There is no recent history of substance abuse for all participants of this study as well as there is controversy about the effects that certain drugs have in Schizophrenia. The original data file contains 6000 genes, after removing genes with one or more missing values, the resulting gene expression profile contains 2,901 genes x 14 experiments. Details of this experiment are available on Bowden et al (2005) [29].

A Gene expression data from typical studies on primary tumors (CCDATA) [18] is the third example that we deal. In this case, the CCDATA data set is based on samples from cervical tumors before and after radiotherapy and is composed of 16 dye swaps and thus 32 experiments arrays. In the original cervical cancer data set, 22% of the data were missing which affecting 70% of the 14229 genes. Genes with one or more missing values are removed; this will leave 4,246 genes. The resulting intensity data matrix will be 4,246 genes x 64 experiments. The data of this example is available on the following website: <http://genome.stanford.edu/listeria/gut/>

The fourth data set is from an infection time series study [5]. Here, all the time course data are downloaded and we remove all genes with missing values, resulting in a 6,850 x 39 data matrix. The data is available on: <http://genomebiology.com/2002/4/1/R2>.

The last data set is gene expression data from a study of Parkinson Disease (PD) introduced in Brown et al. (2002) [28]. In the original file, 17% of the data were missing, affecting 30% of the 9,000 genes. The genes with one or more missing values were removed, leaving data from 5,636 genes. The resulting intensity data matrix is of dimension 5,636 x 80. Detailed information about this data is available in Brown et al. (2002) [28]

In our study, the data set that is used went through several processing steps. Firstly; they were log-transformed after being taken from the image (i.e. after normalization). Secondly, the rows and columns, which contained too many missing values, were excluded. Thirdly, before using the LSR method,

each of the columns was scaled to be between 0 and 1, which means the data sets are normalized. Mean-normalizing the data will further help in regression performance using Euclidean Distance. Finally, the data sets with these pre-processing steps were used to construct the complete matrix.

To evaluate the performance of the missing values estimation methods, we construct the complete matrices by removing all rows containing the missing values, and randomly create the artificial missing values from 10% to 25% of the matrix entries, the artificial missing entries were introduced in two different ways:

Row-based; randomly select a specific percentage of the entries in the complete matrix and remove them, between 10%-25% are removed in each row.

Column-based; randomly select a specific percentage of the entries in the complete matrix, and remove them. Between 10%-25% are removed in each experiment/sample. Column-based methods are only shown in this paper. We can evaluate the performance of the missing value estimation by using Normalized Root Mean Square Error (NRMSE).

3.1. Concluding Results

Table 1 presents a comparative study between the performances of various imputation methods. The results of applying four different methods on five data sets are shown. From the results shown in Table 1, we see that the results reveal that LSR5 method always performs better than the LSR3 method. For example, when the percentage of entries missing is 20%, the NRMSE of LSR5 reaches 0.10395, and the NRMSE of the LSR3 method is 0.12418 for Niehs data.

Fig.1 shows the performance of six different methods on five different data sets. The horizontal and vertical axes indicate the percentage of entries missing in the complete matrix and the NRMSE of each input scheme, respectively. With regards to performance comparison with other methods, the performance of the LSR impute method assessed over five different data sets, has been compared with four imputing approaches namely KNN, LLS, LS impute 3 and LS imputes impute methods. The K- value in the KNN impute method was preset as 15, according to the recommend range of 10 and 20 [3], and both LS impute 3, 5 and the LLS impute methods are non – parametric methods, so they do not require K value. Figs. 1 to 5 show the performance of each method on different data sets.

Table 1. Comparison of basic LSR3 and LSR5 methods against KNNimpute, Simpute3, 5 and LLSimpute with 10% - 25%.

Instance	Techniques					
	LSR 3	LSR 5	LS 3	LS 5	LLS	KNN
Niehs 10%	0.10741	0.093610	0.15942	0.09252	0.10234	
Niehs 15%	0.12418	0.10395	0.17812	0.10508	0.12112	
Niehs 20%	0.15481	0.13911	0.20714	0.13920	0.13408	
Schi 10%	0.98052	0.98389	1.03791	0.97455	0.95456	0.965757
Schi 15%	0.90436	0.89960	1.00851	0.90453	0.88912	0.91175
Schi 20%	0.92459	0.92031	1.09819	0.92283	0.975262	
CCData 10%	0.18637	0.17479	0.26849	0.17312	0.32889	0.80094
CCData 15%	0.19547	0.18220	0.27397	0.183401	0.34484	0.79821
CCData 20%	0.20293	0.18748	0.27530	0.18868	0.36769	
CCData 25%	0.21076	0.194508	0.28147	0.19788	0.38967	
TS 10%	0.26452	0.26345	0.29476	0.25961	0.34111	0.49093
TS 15%	0.26705	0.26569	0.29483	0.26098	0.34975	0.73361
TS 20%	0.26465	0.26266	0.29469	0.25816	0.35355	
TS 25%	0.27736	0.27552	0.31029	0.27102	0.37767	
PD 10%	0.68565	0.68559	0.70847	0.67132	0.78062	0.45583
PD 15%	0.68027	0.66629	0.70515	0.67958	0.77299	0.83692
PD 20%	0.68453	0.67165	0.71104	0.68359	0.78844	
PD 25%	0.68607	0.67312	0.71182	0.68563	0.79359	

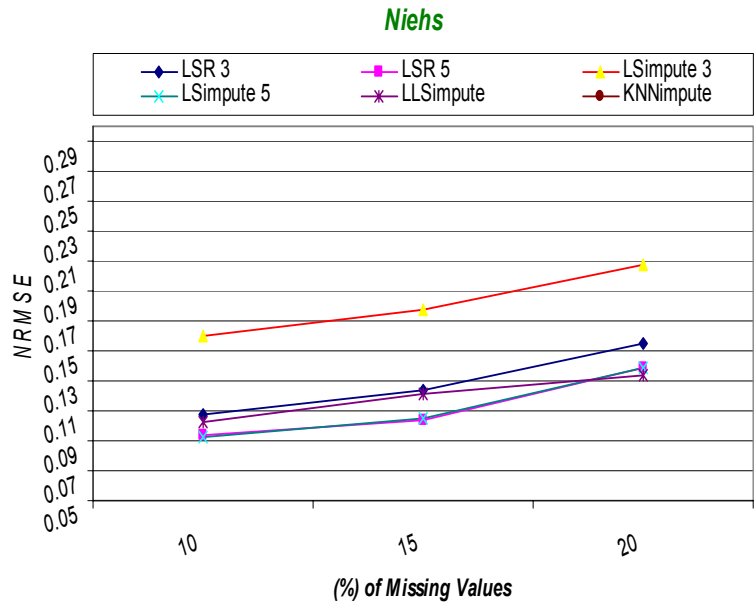


Fig. 1. Performance of the six methods on Niehs data. The percentage of entries missing in the complete matrix and the NRMSE of each missing value estimation method are shown in the horizontal and vertical axes, respectively.

Fig. 1 shows among all other methods, the LSR5 method gives comparable NRMSE values. From this Fig. 1, we see that when the percentage of missing values in the data set is 15%, the LSR achieves best results. When the percentage of the missing values reaches 20%, the NRMSE of the LSR is little larger than LLS impute method and LS impute 3. This shows that LSR method is comparable with if not better than the previous methods on this data set. Both time series data (TS) and the Schizophrenia are preprocessed by removing all genes that containing the missing values. Because our experiments are based on sample imputation, no samples were removed in this experiment, even the ones that contain considerable missing values rate.

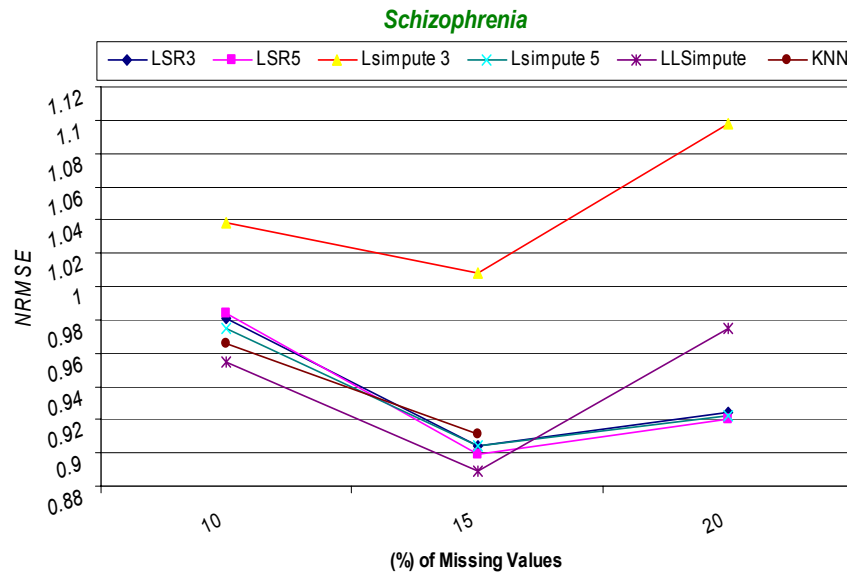


Fig. 2. Performance of the six methods on Schizophrenia data.

From Fig. 2 and 3, we see that LSR5 impute method starts to outperform the other methods when the missing rate is increased especially on the Schizophrenia data set. However, when we apply LSR5 on TS data, the NRMSE of LSR5 is a little larger compared to LS impute5. Generally, the LSR performs stable across the noisy data. Relevant to many kinds of human cancers, including colorectal, ovarian, breast, prostate, as well as Leukemia's and melanomas, which involve much more complex regulation mechanisms, CCData human cancer data requires more reliable algorithms for missing value estimation. In Fig. 4, we illustrate the performance of each method on this data set. In this case, the LSR5 method performs better than the other methods especially when the missing rate is increased. For example, all the other methods give an estimate performance with NRMSE between 0.19788 and 0.38967 for 25% missing, whereas our method gives 0.19451. Consequently, the LSR5 impute method performs robustly as the percentage of the missing values increase.

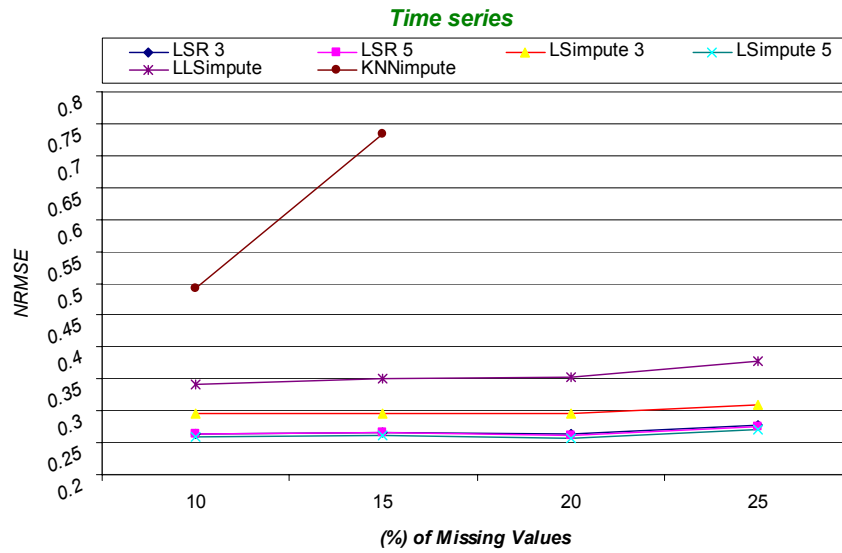


Fig. 3. Performance of the six methods on TS data.

The PD data is used to test how much an imputing method is able to take advantage of strongly correlated genes in estimating the missing values [28]. We can see from Table 1 and Fig. 5 that the LSR5 method outperforms than other methods. However, in terms of memory and running time performance, the LSR5 method can take better use of strongly correlated genes than the other four methods do in estimating the missing values.

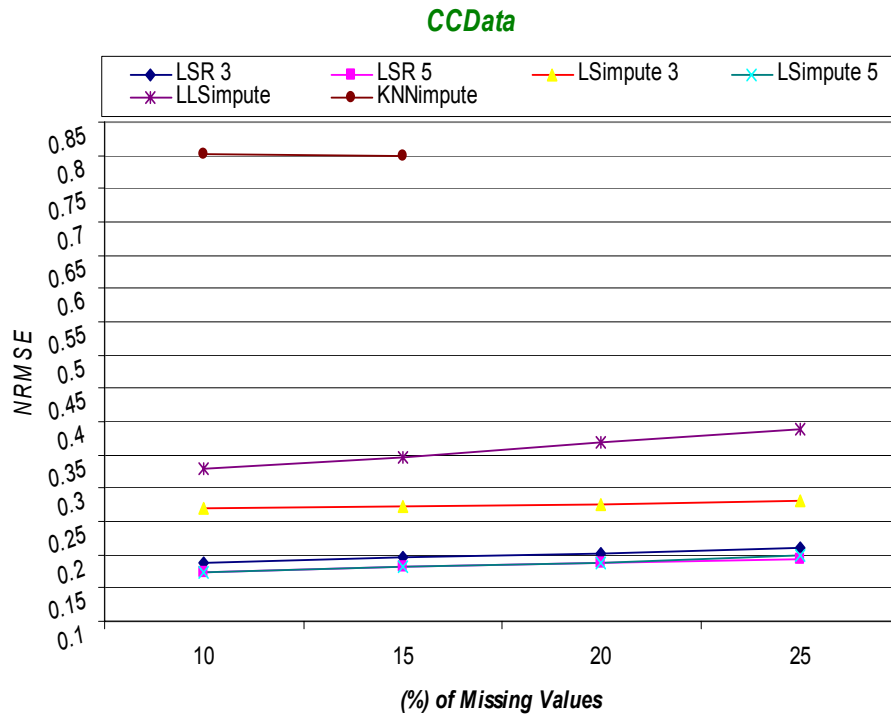


Fig. 4 Performance of the six methods on CCData data.

In this paper, we use four existing imputation methods to evaluate the performance of the LSR impute method. One of the major advantages of the LSR method is that it makes most use of the information from the original data sets. The stepwise regression raises the estimation performance notably, which contributes to the best performance of the LSR method among other methods. The redundant missing values in the samples with many missing values are just neglected in the case of KNN and the LLS while the LSimpute simply regards them equally when modeling the missing values. Another advantage comes from the LSR method itself. The LSR method is the method that is based on the structural minimization principle (SMP is a family of statistical models that seek to explain the relationship among the variables). In doing so, it examines the structure of irrelationships among multiple variables in which the global optimal solution is guaranteed [11],[13]. The KNN method linearly combines the similar genes by weighting the average values of them. The coefficients used in combinations are calculated by using Euclidean Distance, which is not an optimal measurement for gene or sample similarity. This lets the KNN method performs worst among all other methods. The LLS and LSimpute are methods based on linear similarity structure. They share the similar linear combination of K- nearest genes as

the KNN impute, and surpasses the KNN impute by optimizing the coefficients of the nonmissing part of the similar gene using the least square solution. The LLS and LS impute methods are based on local similarity structure of the data set, which draws back its performance when the total similarity is not very clear. In most cases, the LLS method performs worse than LSimpute 5 but better than LSimpute 3.

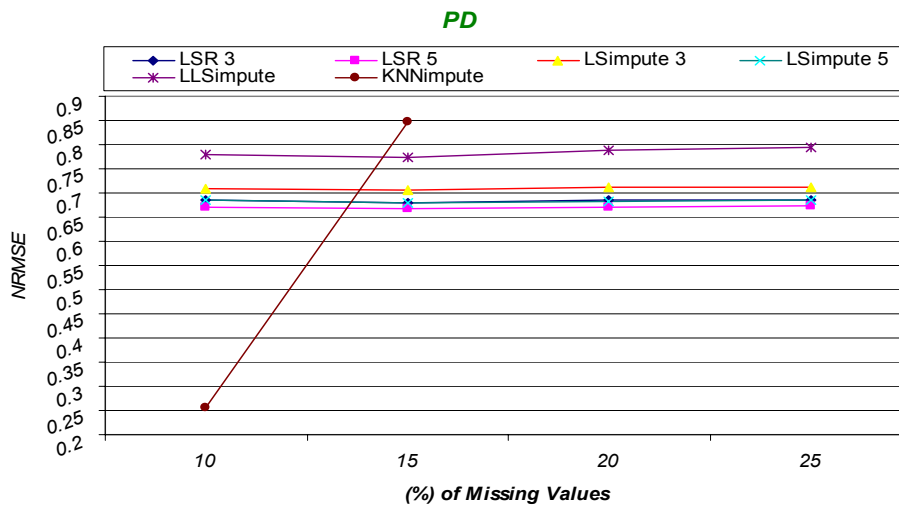


Fig. 5 Performance of the six methods on PD data.

Besides the PD highly correlated data, our method also works well on the data sets for those who are more difficult for regression-based methods, because of the complex regulation mechanisms involved as in the case of CCDATA (Fig.4). Furthermore, the length of the expression profiles in PD data is 80 experiments, which is larger than the experiments in other data sets (LSR isn't affected by the increase in the number of sample/experiments as does by most other methods). This will make it more complex for regression. On the other hand, Fig.3 shows that the LSR5 method achieves comparative results to the other previous methods. When the percentage of missing values becomes too large, the LSR impute method performs little worse than do the LSimpute5. This is partly due to the stepwise regression search strategy for the parameters sets (the number of samples that are chosen form ANOVA step). To maintain proper parameters sets (number of samples), the user should specify the range of the parameters been searched, so the parameters sets might not be the optimum. The parameter selection is also a problem that has to be solved in the linear stepwise regression. Even if the parameter set might not be optimum, the result is still comparative with other impute methods. Thus, the LSR impute method performs well in our presented paper.

Finally, using any imputing algorithm requires the creation of a complex matrix. Calculating a complete matrix can be carried out by using average, zeros or ones as in the case of KNN, LLS, and LSimpute. However, this will cause degradation in the performance of the final algorithm results. LSR algorithm uses a leading algorithm (LSimpute is used in this paper) to create the complete matrix which in turn increases the chances of getting more reliable results. However, if the number of samples in microarray is small, the performance of LSR declines. Consequently, we don't recommend using LSR method over 25% missing and if the number of experiments is less than 15.

4. Conclusion and Future Works

To conclude this paper, different missing values imputation algorithms were explained. Different metric measures were used to measure the performance of the algorithms. Each algorithm was tested under different data sets. However, to validate the performance of each algorithm, more test experiments are needed to be conducted. Furthermore, to be sure that the algorithms are reliable, the same data sets should be used to run the experiments. Finally, from the literature presented, we conclude that the LLS and SVR algorithms have achieved the best performances. SVR can be considered as a complement algorithm for LLS especially when applied to noisy data. However, both algorithms suffer from some deficiencies. LLS has the problem of assigning the parameter value K . Selecting different K -values results in different performances which in turn affects the final metric evaluation for this algorithm. Choosing the optimal K -value should be carried out each time the algorithm is used. SVR, on the other hand, has two disadvantages, first, the choosing of the appropriate kernel function and second, its computational complexity. As a future work to be done by others and to overcome these drawbacks, we need a new method that automatically chooses the parameters of the function and it also has an appropriate computational complexity is imperative.

5. References

1. Brevern, A., Hazout, S., Malpertuy, A.: Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. *BMC Bioinformatics*, Vol. 5. (2004)
2. Oba, S., Sato, M., Takemasa, I., Monden, M., Matsubara, K., Ishii, S.: A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, Vol. 19, 2088-2096. (2003)
3. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.: Missing value estimation methods for DNA microarrays. *Bioinformatics*, Vol. 17, 520-525. (2001)

4. Bo, T., Dysvik, B., Jonassen, I.: LSImpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research*, Vol. 32 (2004)
5. Ouyang, M., Welsh, W., Georgopoulos, P.: Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, Vol. 20, 917-923. (2005)
6. Hair, J., Black, W., Babin, B., Anderson, R., Tatham, R.: *Multivariate data analysis*. 6th edn. Pearson Education, Inc. (2006)
7. Nguyen, D., Wang, N., Carroll, R.: Evaluation of missing value estimation for microarray data. *Journal of Data Science*, Vol. 2, 347-370. (2004)
8. Kim, K.Y., Kim, B.J., Yi, G.S.: Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC Bioinformatics*, Vol. 5. (2004)
9. Brown, P., Botstein, D., Futcher, B.: Comprehensive identification of cell-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molecular biology of the Cell*, Vol. 9, 3273-3297. (1998)
10. Yoshimoto, H., Saltzman, K., Gasch, A., Li, H., Ogawa, N., Brown, P., Cyert, M.: Genome-wide analysis of gene expression regulated by the calcineurin/crzlp signaling pathway in *saccharomyces cerevisiae*. *The Journal of Biological Chemistry*, Vol. 277, 31079-31088. (2002)
11. Gasch, A., Spellman, P., Kao, C., Carmel-Harel, O., Eisen, M., Storz, G., Botstein, D., Brown, P.: Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, Vol.11, 4241-4257. (2000)
12. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J.: Gene ontology: tool for the unification of biology. *Nature Genetics*, Vol. 25, 25- 29. (2000)
13. Tuikkala, J., Elo, L., Nevalainen, O., Aittokallio, T.: Improving missing value estimation in microarray data with gene ontology. *Bioinformatics*, Vol. 6, 566-572 (2006)
14. Alter, O., Brown, P., Botstein, D.: Singular value decomposition for genome-wide expression data processing and modelling. *Proceeding in the National Academy of Sciences (PNAS)*, Vol. 97 (2000)
15. Holter, N., Mitra, M., Maritan, A., Cieplak, M., Banavar, J., Fedoroff, N.: Fundamental patterns underlying gene expression profiles: Simplicity from complexity. *Proceeding in the National Academy of Sciences (PNAS)*, Vol. 97 (2000)
16. Ross, D., Scherf, U., Eisen, M., Perou, C., Rees, C., Spellman, P., Iyer, V., Jeffrey, S., Rijn, M., Waltham, M.: Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, Vol. 24, 227-235 .(2000)
17. Alizadeh, A., Eisen, M., Davis, R., Lossos, I., Rosenwald, A., Boldrick, J., Sabet, H., Tran, T., Powell, J.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, Vol. 403 (2000)
18. Baldwin, D., Vanchianathan, V., Brown, P., Theriot, J.: gene expression program reflecting the innate immune response of intestinal epithelial cells to infection by *listeria monocytogenes*. *Genome Biology*, Vol. 4 (2002)
19. Kim, H., Golub, G., Park, H.: Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics*, Vol. 21, 187-198 (2005)
20. Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., Futcher, B.: Comprehensive identification of cell cycleregulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, Vol. 9, 3273-3297. (1998)
21. Takemasa, I., Higuchi, H., Yamamoto, H., Sekimoto, M., Tomita, N., Nakamori, S., Matoba, R., Monden, M., Matsubara, K.: Construction of preferential cDNA

Mou'ath Hourani and Ibrahiem M. M. El Emary

- microarray specialized for human colorectal carcinoma: Molecular sketch of colorectal cancer. *Biochemistry and Biophysics Research*, Vol. 285, 1244-1249. (2001)
22. Friedland, S., Niknejad, A., Chihara, L.: A simultaneous reconstruction of missing data in DNA microarrays. *Institute of Mathematics and its Applications*, Vol. 1948 (2005)
 23. Venables, W., Ripley, B.: *Modern Applied Statistics with S-PLUS*. 3rd edn. Springer Verlag (1999)
 24. Wang, X., Li, A., Jiang, Z., Feng, H.: Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, Vol. 7 (2006)
 25. Scherf, U., Ross, D., Waltham, M., Smith, L., Lee, J., Tanabe, L., Kohn, K., Reinhold, W., Myers, T., Andrews, D., Scudiero, D., Eisen, M., Pommier, E.S.Y., Botstein, D., Brown, P., Weinstein, J.: A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, Vol. 24, 236-244. (2000)
 26. Sehgal, M., Gondal, I., Dooley, L.: Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data. *Bioinformatics*, Vol. 21, 2418-2423 (2005)
 27. PASW Missing Values – Specifications: Build Better Models When You Estimate Missing Data (2009). [Online]. Available: <http://www.spss.com/software/statistics/missing-values/> (current October 2009)
 28. Brown, V., Ossaadtchi, A., Khan, A. Cherry, S. Leahy, R., Smith, D., High Throughput Imaging of Brain Gene Expression. *Genome Research*, Vol. 12, 244-254. (2002)
 29. Weidenhofer, J. Bowden, N., Expression in the Amygdala in Schizopgerenia: Up-regulation of Genes located in the cytomatrix Active Zone. *Molecular and Cellular Neuro Sciences*, Vol. 31, 243-250 (2006)

Mou'ath A. Hourani, PhD is an Assistant Professor in the Department of Information Technology, Al Ahliyya Amman University, Amman, Jordan. His research interest covers: large microarray datasets analysis using data mining techniques, Cluster analysis of protein, adapting E-learning in higher education environment, business intelligence and ERP systems and service-oriented architectures.

Ibrahim M. El Emary received the Dr. Eng. Degree in 1998 from the Electronic and Communication Department, Faculty of Engineering, Ain shams University, Egypt. From 1998 to 2002, he was an Assistant Professor of Computer science in different faculties and institutes in Egypt. Currently, he is a Visiting Associate Professor at Al Ahliyya Amman University. His research interests include: analytic simulation techniques, performance evaluation of communication networks, application of intelligent techniques in managing computer communication network, bioinformatics and business intelligence as well as performing a comparative studies between various policies and strategies of routing, congestion, sub netting of computer communication networks. He published more than 100 articles in international refereed journals and conferences.

Received: November 12, 2008; Accepted: October 05, 2009.