Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2016, Article ID 6134736, 19 pages
http://dx.doi.org/10.1155/2016/6134736

Research Article

An Improved Fuzzy Based Missing Value Estimation in DNA Microarray Validated by Gene Ranking

Sujay Saha,[1] Anupam Ghosh,[2] Dibyendu Bikash Seal,[3] and Kashi Nath Dey[4]

[1] *Department of Computer Science and Engineering, Heritage Institute of Technology, Kolkata 700107, India*
[2] *Department of Computer Science and Engineering, Netaji Subhash Engineering College, Kolkata 700152, India*
[3] *Department of Computer Science and Engineering, University of Engineering & Management, Kolkata 700156, India*
[4] *Department of Computer Science and Engineering, University of Calcutta, Kolkata 700098, India*

Correspondence should be addressed to Sujay Saha; sujay.saha@heritageit.edu

Received 22 March 2016; Accepted 16 June 2016

Academic Editor: Gözde Ulutagay

Most of the gene expression data analysis algorithms require the entire gene expression matrix without any missing values. Hence, it is necessary to devise methods which would impute missing data values accurately. There exist a number of imputation algorithms to estimate those missing values. This work starts with a microarray dataset containing multiple missing values. We first apply the modified version of the fuzzy theory based existing method LRFDVImpute to impute multiple missing values of time series gene expression data and then validate the result of imputation by genetic algorithm (GA) based gene ranking methodology along with some regular statistical validation techniques, like RMSE method. Gene ranking, as far as our knowledge, has not been used yet to validate the result of missing value estimation. Firstly, the proposed method has been tested on the very popular Spellman dataset and results show that error margins have been drastically reduced compared to some previous works, which indirectly validates the statistical significance of the proposed method. Then it has been applied on four other 2-class benchmark datasets, like Colorectal Cancer tumours dataset (GDS4382), Breast Cancer dataset (GSE349-350), Prostate Cancer dataset, and DLBCL-FL (Leukaemia) for both missing value estimation and ranking the genes, and the results show that the proposed method can reach 100% classification accuracy with very few dominant genes, which indirectly validates the biological significance of the proposed method.

1. Introduction

Microarray expression analysis is a widely used technique for profiling mRNA expression. The mRNA carries genetic information from DNA to the ribosome, where they specify the amino acid sequence of the protein products of gene expression. Microarray datasets often contain missing values which may occur due to various reasons including imperfections in data preparation steps (e.g., poor hybridization and chip contamination by dust and scratches) that create erroneous and low-quality values, which are usually discarded and referred to as missing. It is common for gene expression data to contain at least 5% missing values [1]. Most of the microarray data analysis algorithms, such as gene clustering, disease (experiment) classification, and gene network design, require the complete information, that is, the entire gene expression matrix without any missing values. Hence, different imputation techniques should be used which would accurately impute multiple missing data values. Numerous imputation algorithms have been proposed to estimate the missing values. At first, we have applied modified version of our existing imputation technique LRFDVImpute [2] that first finds a subset of similar genes using the fuzzy difference vector (FDV) algorithm used in [3] where gene expression profiles have been considered as continuous time series curves and then use linear regression on the subset to estimate the missing value. We have considered estimating only those genes with one, two, or three missing values since these genes constitute 5–10% of the entire dataset. Absolute error has been calculated from the difference between the

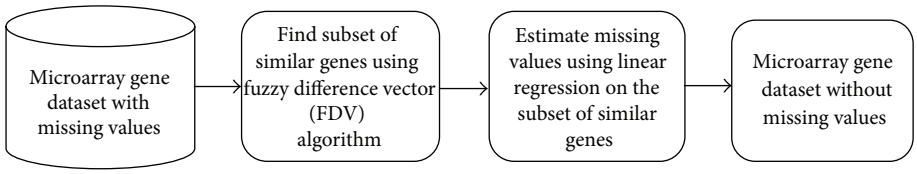

Figure 1: Workflow of the proposed missing value estimation technique.

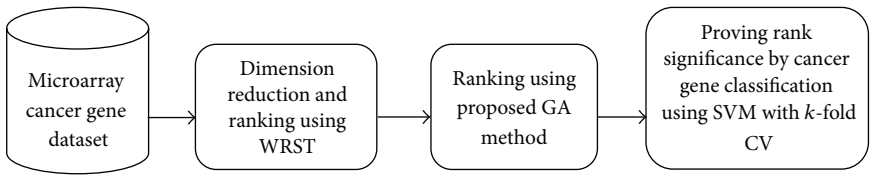

Figure 2: Workflow of the proposed gene ranking technique.

original value and the estimated value. Root Mean Square Error (RMSE) of those absolute errors is then determined.

The workflow for the first phase has been shown in Figure 1.

After that we rank those genes to find the top ranked genes [4]. We have used a hypothesis test, Wilcoxon rank sum test [5], to sort the features (genes) and rank them in order of their p values and select top n genes from them, thereby reducing the dimensionality, where n is the population size that has been used later for GA. The reduced set of genes has then been ranked by our GA method. The two ranks, one by Wilcoxon method and the other by our GA method, are then compared. The top m genes (value of m defined by the user) selected by our method are then used for classification using support vector machine (SVM) classifiers. The performance of classification justifies the efficiency of the ranking method used. Figure 2 shows the workflow for this phase.

Once this is done, we then forcibly make some cells missing in the top ranked genes and again estimate them using the same missing value estimation technique. Finally, we rank them once more to find the top ranked genes. Results show that most of the top ranked genes remain the same, which validates the proposed missing value estimation technique biologically as far as the estimation is concerned.

2. Present State of the Art

As discussed earlier, various statistical and analytical methods used for gene expression analysis are not robust to missing values and require the complete gene expression matrix for providing accurate results. Hence, it is necessary to devise accurate methods which would impute data values when they are missing. Many imputation methods have been proposed. The earliest method, named as row averaging or filling with zeroes, used to fill in the gaps for the missing values in gene dataset with zeroes or with the row average.

KNNImpute method proposed in [1] selects genes with expression profiles similar to the gene of interest to impute missing values. After experimenting with a number of metrics to calculate the gene similarity, such as Pearson correlation, Euclidian distance, and variance minimization, it was found that Euclidian distance was a sufficiently accurate norm.

The SVDImpute method, proposed in [1], uses Singular Value Decomposition of matrices to estimate the missing values of a DNA microarray. This method works by decomposing the gene data matrix into a set of mutually orthogonal expression patterns that can be linearly combined to approximate the expression of all genes in the dataset. These patterns, which in this case are identical to the principle components of the gene expression matrix, are further referred to as eigengenes [6, 7].

Another method named as LLSImpute [8] represents a target gene with missing values as a linear combination of similar genes. The similar genes are chosen by k-nearest neighbours or k coherent genes that have large absolute values of correlation coefficients followed by least square regression and estimation.

BPCAImpute method, proposed in [9], uses a Bayesian estimation algorithm to predict missing values. BPCA suggests using the number of samples minus 1 as the number of principal axes. Since BPCA uses an EM-like repetitive algorithm to estimate missing values, it needs intensive computations to impute missing values.

Another algorithm for time series gene expression analysis is presented in [10] that permits the principled estimation of unobserved time points, clustering, and dataset alignment. Each expression profile is modelled as a cubic spline (piecewise polynomial) that is estimated from the observed data and every time point influences the overall smooth expression curve. The alignment algorithm uses the

same spline representation of continuous time series gene expression profiles.

FDVImpute method, proposed in [11], incorporates some fuzziness to estimate the missing value of a DNA microarray. The first step selects nearest (most similar) genes of the target gene (whose some component is missing) using fuzzy difference vector algorithm. Then the missing cell is estimated by using least square fit on the selected genes in the second step.

FDVSplineImpute, presented in [3], takes into account the time series nature of gene expression data and permits the estimation of missing observations using B-splines of similar genes from fuzzy difference vectors.

Another method, LRFDVImpute, proposed in [2], estimates multiple missing observations by first finding the most similar genes of the target gene and then applying the linear regression on those similar genes. This approach works in two stages. At the first stage, it estimates the real missing cells of SPELLMAN_COMBINED dataset and at the later stage, it makes some cells miss forcefully of the same dataset and then using the estimated results from the first step, this approach estimates those missed cells using the same approach used earlier. Absolute error has been calculated from the difference between the original value and the estimated value. Root Mean Square Error (RMSE) of those absolute errors is then determined.

Extracting relevant information from microarray data is also difficult because of the inherent characteristics of the datasets, where there are the thousands of variables (genes) and very few numbers of samples. Finding out the set of significant genes or, in other words, the most differentially expressed genes, by studying data from tissues affected or unaffected by cancer cells, is an important task. This problem can be termed as gene selection. Several techniques have been used to rank genes and find out the most significant ones.

In [12], the algorithm used discriminant partial least squares (DPLS) and fuzzy clustering methods to interpret the gene expression patterns of acute leukemia and identify leukemia subtypes.

In [13], the proposed method used Mann-Whitney test and k-sample Kruskal-Wallis ANOVA test to rank genes. Dimension reduction was done using k-means clustering and PCA and classification performed using ANN trained during 8-fold cross-validation with recursive feature elimination (RFE) and leave-one-out testing.

In [14], the algorithm proposed a gene selection method based on Wilcoxon rank sum test and SVM. Wilcoxon rank sum test was used to select a subset of genes and then each selected gene is trained and tested using SVM classifier with linear kernel separately, and genes with high testing accuracy rates were chosen to form the final reduced gene subset. Classification was performed on two datasets: Breast Cancer [15] and ALL/AML Leukemia [16] using leave-one-out cross-validation (LOOCV).

A hybrid GA/SVM approach is proposed for gene selection in [17], where a fuzzy logic based preprocessing tool is used to reduce dimensionality, GA for finding out the most frequent genes, and a SVM classifier used for classification. Experiments were performed on two well-known cancer datasets, Leukemia [16] and Colon [18], and results were compared with six other methods.

A multiobjective genetic approach is proposed in [19] for simultaneous clustering and gene ranking where a method to simultaneously optimize the feature ranking and clustering has been used. NSGA-II (Nondominated Sorting Genetic Algorithm-II) [20] has been used as a multiobjective evolutionary algorithm to optimize the chromosomes.

In [21], the proposed algorithm uses feature selection method based on genetic algorithms (GAs) and classification methods focusing on constructive neural networks (CNNs), C-Mantec. Several comparison results on six public cancer databases are provided using other feature selection strategy (Stepwise Forward Selection method) and different classification techniques (LDA, SVM, and Naive Bayes).

A PSO based graph theoretic approach, proposed in [22], is used for identifying the nonredundant gene markers from microarray gene expression data. The microarray data is first converted into a weighted undirected complete feature graph where the nodes represent the genes having gene's relevance as node weights and the edges are weighted in order of correlation among the genes. The densest subgraph having minimum average edge weight (similarity) and maximum average node weight (relevance) is then identified from the original feature graph. Binary particle swarm optimization is then applied for minimizing the average edge weight (correlation) and maximizing the average node weight (gene relevance) through a single objective function.

A web based tool DWFS, proposed in [23], is used to select significant features for a variety of problems efficiently. The search strategy is implemented using Parallel Genetic Algorithm. DWFS also applies various filtering methods as a preprocessing step in the feature selection process. It also uses three classifiers, like KNN classifier, Naive Bayes Classifier, and the combination of these two. Experiments using datasets taken from different biomedical applications show the efficiency of DWFS and lead to a significant reduction of the number of features without sacrificing performance as compared to several widely used existing methods.

3. Proposed Method

3.1. Missing Value Estimation Using Linear Regression. This phase of the work modifies an existing method LRFDVImpute for estimating missing values present in the microarray dataset using linear regression. Earlier version of LRFDVImpute inserts the newly estimated gene into the training data after estimation of each target gene. In this way, the newly estimated gene is taken into consideration while estimating the next target gene. This process has the risk of increasing the error while estimating the subsequent genes since the error term is cumulatively multiplied. To overcome this problem, modified LRFDVImpute does not add the target gene to the training data after it has been estimated. This way, the training gene set size remains constant and with increasing membership values of θ, the size of training data reduces. The effects of modifications have been studied and results are shown in the experimental results section. In our problem,

the genes with missing values in the $(p \times q)$ (p is the number of genes and q is the number of samples) dataset are to be estimated. The method of finding a similar gene as used in [3] using fuzzy difference vector (FDV) algorithm is described below.

Target Row/Testing Data. The row whose missing value is being estimated: a target row may have multiple missing values but in a single run, a single value is estimated.

Similar Rows/Training Data. The rows that are similar to the target row: in this case only those rows are selected that have no missing values. Before applying the similarity measures all the columns from the complete matrix are removed that correspond to missing values in target row.

Let $g_1, g_2, \ldots, g_p$ be the set of genes in the dataset. Let ith g_i be the target gene, that is, the gene with m missing values. We remove the columns having missing values from the entire dataset. Let the resultant matrix contain $(q - m)$ columns. Each target gene i is compared with each of the similar rows in the dataset. For the ith gene g_i, the difference vector V_i of g_i is calculated as follows:

$$\text{DifferenceTable}_{i,j} = g_i(j) - g_i(j+1), \quad 1 \le j \le q - m - 1. \tag{1}$$

Once the difference vectors are calculated for each of the target rows and the similar rows, say DifferenceTable_1 (for target row) and DifferenceTable_2 (for similar row), we then calculate Membership(i) to obtain the number of matches between difference vectors DifferenceTable_1 and DifferenceTable_2 for each target gene g_i. A match in the jth component of the vectors DifferenceTable_1 and DifferenceTable_2 is determined by whether the signs of $\text{DifferenceTable}_1(i, j)$ and $\text{DifferenceTable}_2(i, j)$ are the same or not. Membership(i) defines the degree of match between the distribution of the target gene and the similar gene. We then define a membership grade for g_i as follows:

$$\text{memgrade}(i) = \frac{\text{Membership}(i)}{(q - m - 1)}. \tag{2}$$

The genes in the training data that have a membership value greater than a chosen membership grade θ are considered to be a part of the similar genes.

The steps for estimation can be summarized below:

(1) Load the dataset with missing values.

(2) Calculate the missing number of columns for each gene and start with the first row with the least number of missing values (for our dataset it is 1).

(3) Compute the corresponding membership grade for the target gene from the training data using the FDV algorithm as shown above.

(4) Estimate the missing value using linear regression.

(5) Obtain coefficients of the regression from the linear model object lmObj.

(6) Add a bias of 1 at the beginning of the target row to allow for the bias parameter.

(7) Perform a vector multiplication between the modified target row and the coefficients of regression and add the obtained vector's elements together to get the estimated value.

(8) Replace the missing value with the estimated value.

(9) Go to step (2) and repeat the above steps to fill the missing values unless the mentioned "least number of missing values" in step (2) is less than or equal to 3.

Although we mentioned here that we go on filling the missing value till a point, it is not true. In between we stop this filling in process to do assessment of our algorithm.

After we have filled in all the missing values corresponding to rows with single missing values we select a particular collection of row-column positions corresponding to rows that did not have missing values initially and deliberately treat the values at these positions as missing and use the exact same process to estimate the values.

The same collection of row-column positions are again used when the algorithm has filled up all the rows up to two missing vales and then when it has filled up missing values existing in rows with up to three missing values.

3.2. Gene Ranking Using Genetic Algorithm. In phase 2 of the proposed work, the result of the missing value estimation procedure carried out in phase 1 is biologically validated by ranking the genes using GA. Since a characteristic of gene expression microarray data is that the number of variables (genes) far exceeds the number of samples n, we must reduce its dimension. Executing GA on the original dataset is quite impractical and time consuming. As a preprocessing step, we have reduced the dimension using Wilcoxon rank sum test.

3.2.1. Dimension Reduction Using Wilcoxon Rank Sum Test (WRST). The inputs to the Wilcoxon rank sum test function are the two gene sets, the diseased set and the normal set, both of which have individually undergone the missing value estimation procedure (if there was any missing value). The two gene sets may have different number of samples. Let us consider that the diseased set is a $(p \times q_1)$ sized gene expression data, where p is the number of genes and q_1 is the number of samples, and the normal set has a size $(p \times q_2)$, where q_2 is the number of samples. The Wilcoxon rank sum function processes the two datasets in order to find out for which genes the null hypothesis is accepted or rejected. It returns two values, p value and h-value, as discussed earlier. The null hypothesis for our problem is that the genes are not differentially expressed; that is, either all the samples have come from diseased patients or they have come from normal patients. The alternative hypothesis can be that genes are differentially expressed. We record the p values and h-values for each gene.

In the next step, we consider only those genes for which the alternative hypothesis holds ($h = 1$) at the significance level alpha and sort the genes according to the p values thereby ranking the genes. We then select the topmost k

genes, where k is the population size that has been used for GA later. Thus, we have two reduced populations, one representing diseased and the other representing normal tissues. Let $(X_{ij})_{p\times q_1}$ be the diseased set, where p is the reduced set of genes and q_1 is the number of samples, respectively, and let $(Y_{ij})_{p\times q_2}$ be the normal set, where q_2 is the number of samples.

3.2.2. Chromosome Representation and Initial Population for GA. The reduced gene sets $(X_{ij})_{p\times q_1}$ and $(Y_{ij})_{p\times q_2}$ serve as the initial population for the genetic algorithm step. They contain pop_size number of genes which is preselected by the user. We use real value encoding to represent each chromosome; that is, X_{ij} and Y_{ij} are the measurements recorded for the ith gene and jth sample for each population, respectively.

3.2.3. Fitness Calculation. The fitness for each gene in the reduced gene sets is again calculated by a method similar to that used in [14] where gene expression profiles have been considered as continuous time series curves.

In our problem, we have two populations, one for the diseased tissues and the other for the normal tissues. The two populations contain the same number of genes p but may have different number of samples. In that case, we consider the minimum of the two and extract the same number of samples from each set.

Let $g_1, g_2, \ldots, g_p$ be the reduced set of genes in each population. If $q = \min(q_1, q_2)$, then for each population, the difference vector V_i of g_i is calculated using (1). Once the difference vectors are calculated for each of the two populations, say DifferenceTable_1 (for diseased) and DifferenceTable_2 (for normal), the number of matches between the difference vectors Membership(i) and the membership grade for g_i is computed using (2).

The fitness of gene g_i is the reciprocal of memgrade(i) and is calculated as

$$\text{fit}(i) = \frac{1}{\text{memgrade}(i)}. \tag{3}$$

This signifies that the more similar the distributions of gene g_i in the two populations are, the less differentially expressed the gene is, and vice versa. Thus, a fitter gene will have different distributions in the two populations. We then rank the genes in order of their fitness.

3.2.4. Elitism. We have used an elitist version of GA where the best chromosomes are carried forward to the next generation unchanged; that is, the crossover and mutation operators are not applied on the best chromosomes. This technique ensures faster convergence of the process by keeping track of the best solutions.

3.2.5. Selection. For selection, we have used a roulette wheel technique where genes are selected based on their relative fitness values. The better the chromosomes are, the more chances to be selected they have. Let count be the number of elite children. We construct a roulette wheel as follows [22]:

(i) Calculate the fitness value fit(i) for each chromosome g_i, count + 1 $\le i \le p$.

(ii) Find the total fitness of the population $F = \sum_{i=\text{count}+1}^{\text{pop\_size}} \text{fit}(g_i)$.

(iii) Calculate the probability of selection p_i for each chromosome g_i, count + 1 $\le i \le p$:

$$p_i = \frac{\text{fit}(g_i)}{F}. \tag{4}$$

(iv) Calculate a cumulative probability c_i for each chromosome g_i, count + 1 $\le i \le p$:

$$c_i = \sum_{j=1}^{i} p_j. \tag{5}$$

We now spin the wheel (pop_size − count) times and select a single chromosome as follows:

(i) Generate a random number (float) r between 0 and 1.

(ii) If $r < c_1$, we select the first chromosome g_1; otherwise, select the ith chromosome g_i ($2 \le i \le$ pop_size − count) such that $c_{i-1} < r \le c_i$.

Some chromosomes get selected more than once. According to *Schema Theorem* [24], the best chromosomes get more copies, the average stay even, and the worst die off.

3.2.6. Crossover. For crossover, we proceed as follows.

For each chromosome g_i in the population,

(i) generate a random number (float) r between 0 and 1,

(ii) if $r < p_{\text{cross}}$ (crossover probability), we select the given chromosome for crossover.

We have used single point crossover where the crossover site is also generated randomly in the range $[1 \cdots q-1]$, where q is the number of samples. Thus after crossover, a pair of parent chromosomes generates a pair of offspring chromosomes [25]. The new population obtained after crossover contains the new generation produced by crossover as well as the elite children that did not undergo crossover. This new population is used in the mutation process.

3.2.7. Mutation. A nonuniform mutation operator as proposed in literature [25] has been used here. The new operator is defined as follows:

(i) A random experiment is carried out which produces an outcome which is either 0 or 1.

(ii) Another random number pos is generated in the range $[1 \cdots q-1]$, where q is the number of samples, to select the mutation site.

(iii) Let $g_i^t = [g_i(1), g_i(2), \ldots, g_i(j), \ldots, g_i(q)]$, $1 \le j \le q$, be the chromosome, and let $g_i(j)$ be selected for mutation. Domain of g_i is [lb, ub]; the resultant vector $g_i^{t+1} = [g_i(1), g_i(2), \ldots, g_i(j)', \ldots, g_i(q)]$:

Table 1: Characteristic of Spellman dataset.

Dataset	Start	End	Sampling	Complete genes
alpha	0 m	119 m	Every 7 m	4489
cdc15	10 m	290 m	Every 20 m for 1 hr, 10 m for 3 hr, and 20 m for final hr	4381
cdc28	0 m	160 m	Every 10 m	1383
elu	0 m	390 m	Every 30 m	5766

Yeast *Saccharomyces cerevisiae* dataset of Spellman et al. [26].
Source: http://genome-www.stanford.edu/cellcycle/data/rawdata/combined.txt.
Organism: yeast.

$$g_i(j)' = \begin{cases} g_i(j) + \Delta(t, \text{ub} - g_i(j)), & \text{if outcome is 0} \\ g_i(j) - \Delta(t, g_i(j) - \text{lb}), & \text{if outcome is 1,} \end{cases} \quad (6)$$

where t is the generation number and the function $\Delta(t, y)$ returns a value in the range $[0 \cdots y]$ such that the probability of $\Delta(t, y)$ being close to 0 increases as t increases. This property causes this operator to search the space uniformly initially (when t is small) and very locally at later stages.

$\Delta(t, y)$ is calculated as

$$\Delta(t, y) = y\left(1 - r^{(1-t/T)^{\beta}}\right), \quad (7)$$

where r is a random number in the range $[0 \cdots 1]$, T is the maximum number of generations preselected by the user, and β is a system parameter determining the degree of uniformity. We have used $\beta = 2$ for our experiment.

The entire genetic transformation has been performed on one population with respect to the other. We made the diseased gene set to undergo genetic transformation while fitness evaluation has been made with respect to the normal gene set. The opposite transformation will produce similar results.

Once the genetic transformations are done, we obtain a final population set (here, genetically transformed diseased gene set) which have been ranked in order of their fitness. We compare the two ranks, one by the Wilcoxon method and the other by our GA method. A threshold of ±2 has been considered while comparing the two ranks. Results show that there is a good percentage of matches in the two ranks. Moreover, we find out the top ranked genes produced by both methods and the significant genes produced by the two methods are also similar. This also validates the result of the missing value estimation method carried out in phase 1.

3.3. Gene Classification Using SVM. In order to prove the significance of ranking by our GA method, we perform classification. The top ranked n genes, n' $\{5, 10, 15, 20, 25\}$, ranked by our GA method are used for the purpose. We use k-fold LOO cross-validations, where k is varied from one dataset to another depending on the number of samples. For cross-validation, we have divided our dataset into two sets, a training set and a testing set, in 80 : 20 ratio. The reason behind taking this ratio is that 80 : 20 is a commonly occurring ratio, which is often referred to as Pareto Principle. So, if there are n samples in the training set and $m - n$ samples in the test set, where m is the total number of samples, the training set is divided into k equal sized subsets. Of the k subsets, one subset is retained for validation and the remaining $k-1$ subsets are used as training data. Thus, k SVM classifiers with linear kernel are trained using the n training subsets. The classification accuracy rates are recorded and the classifier with the best accuracy rate is used to test the $m - n$ samples.

4. Experimental Results

4.1. Datasets Used. The missing value estimation part of the proposed modified LRFDVImpute technique has been evaluated on the publicly available yeast cell cycle time series dataset from Spellman et al. [26] described in Table 1.

After the experiments on Spellman dataset are done, the combined gene ranking and classification portion of the proposed method are evaluated on four publicly available datasets: Colorectal Cancer tumours dataset (GDS4382), Breast Cancer dataset (GSE349-350), Prostate Cancer dataset, and Leukaemia Cancer dataset (DLBCL-FL).

4.2. Platform Used. All algorithms have been implemented using MATLAB R2013a in Windows 8.1.

4.3. Results

4.3.1. Results of Missing Value Estimation Part. We perform the initial estimation using modified version of LRFDVImpute with a membership grade $\theta = 0.55$. After the initial estimation is over, we forcibly treat cells at specified locations as missing and estimate them using different membership values of θ and both earlier and modified versions. This has been carried out only once, after estimating rows with single missing values and the corresponding RMSE values have been recorded. We have performed our experiments only on alpha, cdc15, and elu data of Spellman dataset. The number

Table 2: Results for missing value estimation algorithm (Spellman, alpha).

RMSE\θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Original LRFDVImpute	0.012405344	0.012488181	0.012562782	0.012562782	0.012690904	0.012197374	0.012638865
Modified LRFDVImpute	0.012439936	0.012439366	0.012389872	0.012389872	0.012645466	0.011988263	0.013268721

Table 3: Results for missing value estimation algorithm (Spellman, cdc15).

RMSE\θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Original LRFDVImpute	0.016832094	0.016760968	0.016706119	0.016682418	0.016768837	0.016733642	0.049482242
Modified LRFDVImpute	0.016781257	0.016710318	0.016736613	0.016723349	0.016637753	0.017023671	0.057225615

Table 4: Results for missing value estimation algorithm (Spellman, elu).

RMSE\θ	0.4	0.45	0.5	0.55	0.6	0.65	0.7
Original LRFDVImpute	0	0	0	0	0	0	0
Modified LRFDVImpute	0	0	0	0	0	0	0

Table 5: Performance comparison with other existing methods.

Dataset	SVDImpute	LLSImpute	FDVLLSImpute	FDVSPLINEImpute	FDVLRImpute with $\theta = 0.55$	
					Original LRFDVImpute	Modified LRFDVImpute
alpha	0.03395	0.07853	0.096	0.063	0.012562782	0.012389872
cdc15	0.05055	0.1208	0.258	0.127	0.016682418	0.016723349
elu	0.01585	0.0033	0.044	.019	0	0

of missing values is too large for cdc28; that is why we ignore that segment. The results for the alpha, cdc15, and elu datasets using both methods are shown in Tables 2–4. Figures 3–5 show the corresponding plots of RMSE versus membership grade θ for each of the four datasets.

Table 5 compares the performance of both versions of LRFDVImpute method to that of some other existing methods, like SVDImpute, LLSImpute, FDVLLSImpute, FDVSPLINEImpute, and so forth, and the results show that modified version of LRFDVImpute outperforms the other existing methods as far as RMSE value is concerned.

4.3.2. Combined Results. We test the significance of our proposed missing value estimation technique using the gene ranking method. We have not found any state-of-the-art work on gene ranking so far where Spellman dataset is used. That is why we use four more publicly available real-life gene expression datasets, like Colorectal Cancer dataset (GDS4382), Breast Cancer dataset (GSE349-350), Prostate Cancer dataset, and Leukaemia Cancer dataset (DLBCL-FL) [4, 27–32], to perform steps such as missing values estimation and gene ranking and analyze the results. We start with the microarray dataset containing missing values and apply our proposed missing value estimation technique to estimate the genes with missing values (if any). We rank them using proposed gene ranking method and find the top ranked genes. We then forcibly insert missing values in the top ranked genes and again estimate them using the same missing value estimation technique. Finally, we rank them once more to find the top ranked genes. Results show that most of the top ranked genes remain the same, which implies that the proposed missing value estimation technique has been accurate in estimating the unknown values. We have normalized most of the datasets using z-score normalization method in order to bring the data values to a common scale.

Tables 6, 8, 10, and 13 show the estimated values for the four datasets, Tables 7, 9, 11, and 14 show the common gene indices before and after the estimation, and Tables 12 and 15 compare the performance of the proposed approach with two state-of-the-art methods [22, 23] for Prostate and Leukaemia dataset on the basis of accuracy, sensitivity, specificity, $F1$-score, and G-mean metrics. We have found that Prostate and Leukaemia are the common dataset on which both the existing methods have done their experiments. The results show that the proposed gene ranking approach performs far better compared to those existing approaches, where one is a PSO based graph theoretic approach [22] and the other is a web based tool DWFS, which uses KNN and NBC classifiers [23] as far as those metrics are concerned.

Table 6: Estimated values for Colorectal Cancer GDS4382.

Colorectal Cancer GDS4382							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
714	12	0.677252397	0.703080766	714	12	0.140108496	−0.163286572
1245	5	1.56686079	1.755118642	1245	5	1.212949296	1.148663192
1578	3	0.787510895	1.003134829	1578	3	0.256998387	0.22381735
1763	11	1.024714768	0.737414862	1763	11	0.20681001	0.056394235
2792	9	−0.861395162	−0.865043164	2792	9	−1.064597763	−1.005338727
4025	1	0.781326343	1.137532185	4025	1	−0.31562626	−0.297767551
4134	15	0.892338308	0.958542974	4134	15	0.329755342	0.247541595
5082	2	−0.006360431	0.32763543	5082	2	−0.480000425	−0.370717546
8426	13	1.210288879	0.790345743	8426	13	−0.082823022	0.157094029
9979	6	2.932068401	2.953449691	9979	6	2.387826603	2.246418246
10083	11	1.369142269	1.258410425	10083	11	1.915272483	1.809541605
10145	10	1.940467541	1.852338203	10145	10	2.750210569	2.834220655
10208	3	1.868505436	1.778084682	10208	3	2.224446282	2.122262802
10280	6	1.424951861	1.521769713	10280	6	0.98680627	1.139738659
10323	1	0.895845032	0.963530648	10323	1	0.378278185	0.376819137
10725	14	0.562534293	0.903674258	10725	14	1.263242627	1.285858157
10789	4	1.582210172	1.834961289	10789	4	0.507667085	0.540477481
10855	10	3.17769843	3.162360832	10855	10	2.562212194	2.546399932
11050	3	2.676634486	2.657786027	11050	3	1.872943693	1.894214081
11055	8	2.112261939	2.105431748	11055	8	1.680697142	1.713521495
11100	16	2.522783399	2.455429314	11100	16	3.208462284	3.064140274
11465	1	2.454481056	2.18211664	11465	1	1.232460868	1.356633277
11485	12	−0.701537989	−0.49772711	11485	12	0.609015246	0.316407836
11650	6	1.470662615	1.35686403	11650	6	1.940624638	2.047625027
11677	4	2.591635801	2.602068714	11677	4	2.903830552	2.934334312

Table 7: Top 25 gene indices before and after estimation for GDS4382.

Rank	Ranking Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	714	714
2	1245	1245
3	1578	1578
4	1763	1763
5	2792	2792
6	4025	4025
7	4134	4134
8	5082	5082
9	8426	8426
10	9979	9979
11	10083	10083
12	10145	10145
13	10208	10208
14	10280	10280
15	10323	10323
16	10725	10725
17	10789	10789
18	10855	10855
19	11050	11050
20	11055	11055
21	11100	11100
22	11465	11465
23	11485	11485
24	11650	11650
25	11677	11677
Number of common genes in top 25 positions = 25 *% of common genes = 100*		

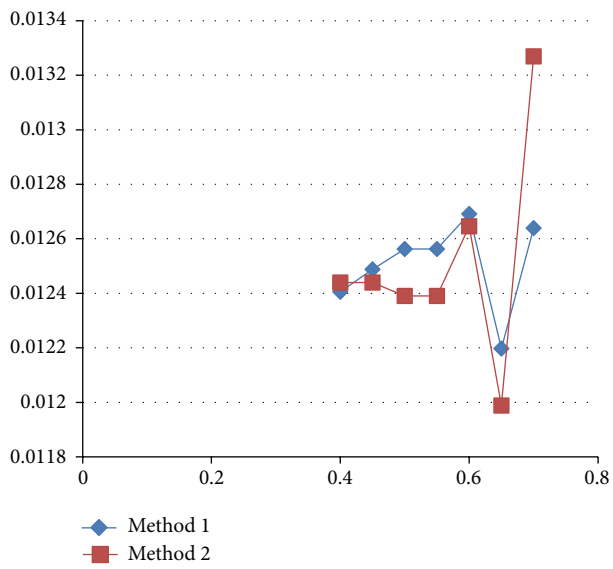

Figure 3: Plot of RMSE versus membership grade θ for alpha dataset.

Table 8: Estimated values for Breast Cancer dataset.

Breast Cancer							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
272	12	−0.354039943	−0.419709522	272	8	−0.407057103	−0.396601672
329	5	−0.176687651	−0.295980517	329	2	−0.021584122	0.092981006
491	3	0.222486126	0.30260015	491	10	0.140530363	0.067353238
869	11	0.79646566	0.852100043	869	5	0.345360946	0.279331888
1143	9	0.017956445	0.256958128	1143	4	0.054582833	−0.145618319
1937	1	0.22672464	0.33943153	1937	7	−0.128257731	0.05338273
2825	8	7.552441375	8.108907291	2825	3	6.080319682	5.969522121
3004	2	0.128475064	0.04404869	3004	6	−0.124353092	−0.113116385
4911	4	−0.550612392	−0.372472073	4911	9	−0.090276567	−0.152051028
5328	6	10.57228289	9.556257324	5328	5	8.316072021	7.887609502
6184	11	−0.493537621	−0.504575199	6184	8	0.035382884	−0.132574115
7941	10	0.233974208	0.14984931	7941	6	0.077199916	0.101298927
8452	14	−0.312716903	−0.32992237	8452	3	−0.071344506	−0.092371778
9076	6	0.708996621	0.154489296	9076	1	0.060554625	0.036078623
9267	13	−0.270092957	−0.33842598	9267	10	−0.034956721	0.30142636
9574	7	0.093118778	0.074772718	9574	7	−0.227882015	−0.159726559
9723	4	−0.018883445	−0.279787171	9723	5	0.407076237	0.815509208
9753	10	−0.23265185	−0.264909557	9753	8	−0.228475796	−0.265794544
9905	3	−0.31537376	−0.417539213	9905	3	−0.286431974	−0.199211435
10319	8	−0.049692673	−0.038561161	10319	9	−0.218073382	−0.242557019
10614	9	−0.511734814	−0.430792872	10614	2	−0.268390734	−0.1922339
11377	1	−0.055809992	0.083662727	11377	7	−0.268295765	−0.144527083
11737	12	−0.34275829	−0.269422135	11737	4	−0.387083296	−0.093221982
11976	6	0.030992978	−0.133189906	11976	9	−0.270598103	−0.272899078
12053	4	−0.374640827	−0.328380483	12053	10	−0.360077886	−0.244255658

Table 9: Top 25 gene indices before and after estimation for Breast Cancer dataset.

Rank	Ranking Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	3004	1143
2	7941	3004
3	10319	7941
4	869	10319
5	5328	11737
6	9723	491
7	491	9753
8	9574	869
9	9905	5328
10	11737	9723
11	2825	12053
12	4911	2825
13	8452	9574
14	272	9905
15	329	4911
16	6184	8452
17	9076	9076
18	9267	329
19	9753	6184
20	10614	9267
21	11377	11976
22	11976	2218
23	12053	2459
24	1143	2995
25	1937	4200
Number of common genes in top 25 positions = 21		
% of common genes = 84		

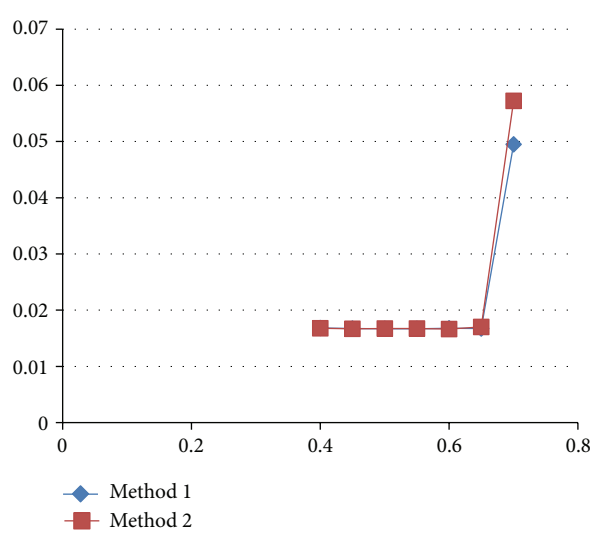

Figure 4: Plot of RMSE versus membership grade θ for cdc15 dataset.

Table 10: Estimated values for Prostate Cancer dataset.

Prostate Cancer							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
205	45	7.890085103	9.434976702	205	45	5.168380524	5.416011322
2839	5	−0.176203136	0.004447962	2839	5	−0.173585261	−0.183852307
3649	10	0.296457522	0.046654036	3649	10	0.329958724	0.617420849
3794	8	−0.241993865	−0.216817937	3794	8	−0.106322335	−0.109294537
4365	17	0.059874409	2.443242853	4365	17	−0.122967971	−0.099879376
5757	14	0.090277183	0.70723886	5757	14	−0.196702226	−0.067191045
5944	22	−0.180239863	−0.186977867	5944	22	−0.137411352	−0.217659599
6185	36	1.557345019	1.548681099	6185	36	−0.010802226	−0.181261903
6462	28	−0.16997265	−0.175762613	6462	28	0.202288499	0.191383577
7247	32	0.859643204	0.956551639	7247	32	0.932414697	2.21258573
7520	18	0.270631813	0.179429504	7520	18	0.064953651	0.155169008
7557	11	0.393712194	0.375941042	7557	11	1.450975133	1.084180805
7768	3	0.042090452	0.11563321	7768	3	0.59875936	0.576608917
8123	47	0.216968028	0.283330867	8123	47	0.128003038	0.165883246
8554	4	0.015161421	0.083268762	8554	4	0.40771258	−0.051156112
8768	26	−0.135452978	−0.04501778	8768	26	−0.138712043	−0.188517852
8850	17	2.708006948	1.916254815	8850	17	−0.364936796	−0.228740529
9034	29	−0.190217605	−0.178815842	9034	29	−0.194249889	−0.047431811
9050	34	0.351634344	0.174924563	9050	34	0.284389403	0.469260339
9172	42	2.277059356	2.347704101	9172	42	1.727648712	1.562259352
9850	50	−0.046243874	−0.290856086	9850	50	1.371476261	2.134924482
10138	14	0.599428228	0.628396963	10138	14	0.765017083	1.001711406
10494	22	−0.189181177	−0.122575727	10494	22	0.220022109	0.365466432
10537	48	0.139373755	0.406906442	10537	48	0.035059438	0.011805144
10956	7	0.337716243	0.155279221	10956	7	0.146282156	0.169249764

Table 11: Top 25 gene indices before and after estimation for Prostate Cancer dataset.

Rank	Ranking Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	6185	6185
2	10494	10494
3	9850	4365
4	4365	9850
5	10138	9034
6	9172	10138
7	9034	5944
8	5944	9172
9	3649	3649
10	8554	2839
11	2839	7557
12	7557	10956
13	205	9050
14	3794	7520
15	10956	3794
16	8850	205
17	7520	8850
18	9050	10537
19	10537	5757
20	5757	8554
21	8123	8768
22	6462	8123
23	8768	6462
24	7247	7247
25	7768	9093
Number of common genes in top 25 positions = 24		
% of common genes = 96		

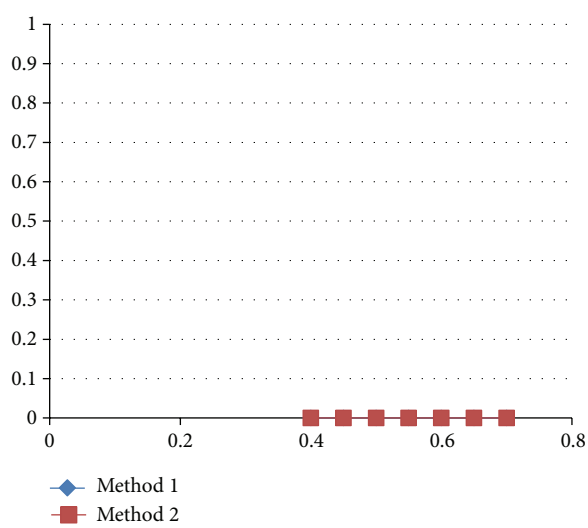

Figure 5: Plot of RMSE versus membership grade θ for elu dataset.

Table 12: Performance comparison for Prostate dataset.

Dataset name				Prostate			
Number of genes	Number of samples	Number of samples in Cancer dataset	Number of samples in normal dataset	Number of samples in training dataset	Number of samples in testing dataset	SVM kernel used	Number of folds for LOOCV
12600	102	52	50	82 (42 cancer, 40 normal)	20 (10 cancer, 10 normal)	Linear	41
Algorithm		% accuracy		Sensitivity/recall	Specificity	$F1$-score	G-mean
Proposed approach		100 (with top 5 genes)		1	1	1	1
PSO based graph theoretic approach		91		0.91	0.92	0.91	0.91
DWFS using KNN classifier		86		0.87	0.85	0.86	0.86
DWFS using NBC classifier		80		0.76	0.85	0.78	0.80

Table 13: Estimated values for DLBCL-FL.

Leukaemia (DLBCL-FL)							
Cancer				Normal			
Missing values inserted		Old_value	Estimated value with mem = 0.55	Missing values inserted		Old_value	Estimated value with mem = 0.55
At row	At column			At row	At column		
28	50	−0.188603483	−0.203477278	28	2	3.886811534	0.84848513
447	45	0.103798973	0.304241026	447	18	−0.146274929	0.009235547
546	8	1.055798216	1.246420135	546	8	0.140697423	0.320415944
640	52	0.052220359	−0.311252744	640	7	0.12552933	0.364757746
913	5	0.15759485	0.180411915	913	2	−0.135855848	−0.165390479
1129	4	1.051658222	0.598949762	1129	10	0.04903928	0.096097027
1142	31	−0.249901255	−0.311776993	1142	1	−0.249625598	−0.196558516
1293	34	1.91037315	2.330011528	1293	3	0.27829269	0.224522592
1553	48	−0.218067263	−0.107003318	1553	9	−0.00577268	0.602953445
1731	26	0.555312243	0.39207635	1731	7	0.448099319	0.524633075
2062	22	2.366047008	4.45272374	2062	13	0.201613611	0.849245039
2929	17	0.072886107	0.125837311	2929	11	−0.007410482	0.004016071
3965	56	0.137340738	−0.905551574	3965	16	−0.133906259	−0.083702381
3969	36	2.328255836	2.60971631	3969	6	0.520731724	0.584352541
4124	17	0.077170048	0.183068221	4124	5	−0.157303536	−0.036843328
4135	10	−0.35554626	−0.339641516	4135	10	−0.232750407	−0.161598404
4143	14	0.185986189	−0.024609518	4143	7	−0.032712551	0.024485269
4233	18	3.990305253	1.582890447	4233	8	0.770838584	0.706726475
4313	11	0.453186971	0.126025314	4313	15	−0.129336617	−0.079142717
4510	22	−0.167456263	−0.135667934	4510	17	−0.046588554	−0.100490086
5327	46	−0.415159893	−0.324486992	5327	12	−0.146934163	−0.22684469
6120	3	4.432535499	4.497409395	6120	3	0.919305283	0.723423487
6417	7	−0.253214481	−0.312493286	6417	10	−0.141004928	−0.007636336
6434	42	−0.220525385	−0.208472395	6434	14	−0.001659417	0.063668162
6756	55	2.0189449	1.307321937	6756	4	−0.009878439	0.570901233

Table 14: Top 25 gene indices before and after estimation for DLBCL-FL.

Rank	Ranking	
	Gene indices prior to missing value insertion	Gene indices after missing value insertion
1	447	447
2	913	4135
3	4135	913
4	546	640
5	2929	2929
6	640	1142
7	4510	546
8	3969	3969
9	5327	4510
10	6756	4233
11	4233	5327
12	4313	4313
13	6120	6756
14	1142	6120
15	1129	1553
16	1731	1129
17	4124	6417
18	3965	4124
19	1293	1731
20	6434	1293
21	28	6434
22	4143	28
23	2062	2062
24	1553	4094
25	6417	1984
Number of common genes in top 25 positions = 23		
% of common genes = 92		

5. Conclusion and Future Scope

The proposed modified version of LRFDVImpute technique has been tested on the dataset from Spellman et al. [26] and has shown impressive results. It outperforms some state-of-the-art methods. The plots of RMSE versus membership grade θ show that modified version is equivalent to or better than earlier version for the alpha and cdc15 datasets. However, for the cdc28 dataset, earlier version has shown better results. For the elu datasets, both have reached 0 error margin. For both versions, a membership grade between 0.55 and 0.65 produces minimum error and any value in this range can be considered as a threshold to be used for fresh experiments.

The validation of the missing value estimation shows that most of the top ranked genes remain the same, before and after imputation, which implies that the proposed modified LRFDVImpute technique has been accurate in estimating the unknown values.

As a future scope, we would like to analyze the effects of using quadratic regression for estimation of missing values and the use of data cleaning techniques before imputation which may remove outliers if any and may further reduce the error margin. For gene ranking, we wish to analyze the effects of different parameter settings for GA and observe the ranking and classification results using SVM with other kernels and also compare results with the ones mentioned in literature. We would also wish to modify our algorithms so as to make this ranking more efficient and find out the most significant genes that would correctly identify the subtypes of a particular type of cancer. For the Leukemia dataset [16], this could be identifying the B-cell and T-cell lineages for the acute lymphoblastic leukemia (ALL) samples.

Competing Interests

The authors declare that they have no competing interests.

Table 15: Performance comparison for DLBCL-FL.

Dataset name				DLBCL-FL			
Number of genes	Number of samples	Number of samples in Cancer dataset	Number of samples in normal dataset	Number of samples in training dataset	Number of samples in testing dataset	SVM kernel used	Number of folds for LOOCV
7070	77	58	19	62 (47 DLBCL, 15 FL)	15 (11 DLBCL, 4 FL)	Linear	31
Algorithm		% accuracy		Sensitivity/recall	Specificity	$F1$-score	G-mean
Proposed approach		100 (with top 5, 10, 15, and 20 genes)		1	1	1	1
PSO based graph theoretic approach		94		0.95	0.94	0.89	0.94
DWFS using KNN classifier		91		0.97	0.85	0.94	0.9
DWFS using NBC classifier		96		1	0.9	0.98	0.94

References

[1] O. Troyanskaya, M. Cantor, G. Sherlock et al., "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[2] S. Saha, P. K. Singh, and K. N. Dey, "Missing value estimation in DNA microarrays using linear regression and fuzzy approach," in *Proceedings of the 4th International Conference on Advances in Computer Science and Application (CSA '15)*, pp. 62–70, World Scientific, Thiruvananthapuram, India, October 2015.

[3] S. Saha, K. N. Dey, R. Dasgupta, A. Ghose, and K. Mullick, "Anirban ghose, and koustav mullick: missing value estimation in DNA microarrays using B-splines," *Journal of Medical and Bioengineering*, vol. 2, no. 2, pp. 88–92, 2013.

[4] L. C. Crossman, M. Mori, Y.-C. Hsieh et al., "In chronic myeloid leukemia white cells from cytogenetic responders and non-responders to imatinib have very similar gene expression signatures," *Haematologica*, vol. 90, no. 4, pp. 459–464, 2005.

[5] Graham Hole Research Skills, *The Wilcoxon Test*, Version 1.0, 2011.

[6] O. Alter, P. O. Brown, and D. Botstein, "Singular value decomposition for genome-wide expression data processing and modeling," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, no. 18, pp. 10101–10106, 2000.

[7] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[8] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[9] S. Oba, M.-A. Sato, I. Takemasa, M. Monden, K.-I. Matsubara, and S. Ishii, "A Bayesian missing value estimation method for gene expression profile data," *Bioinformatics*, vol. 19, no. 16, pp. 2088–2096, 2003.

[10] Z. Bar-Joseph, G. K. Gerber, D. K. Gifford, T. S. Jaakkola, and I. Simon, "Continuous representations of time-series gene expression data," *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 341–356, 2003.

[11] S. Chakraborty, S. Saha, and K. Dey, "Missing value estimation in DNA microarray—a fuzzy approach," *International Journal of Artificial Intelligence and Neural Networks (IJAINN)*, vol. 2, no. 1, 2012.

[12] C. Yooa, I. B. Leeb, and P. A. Vanrolleghema, "Interpreting patterns and analysis of acute leukemia gene expression data by multivariate fuzzy statistical analysis," *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1345–1356, 2005.

[13] L. E. Peterson and M. A. Coleman, "Comparison of gene identification based on artificial neural network pre-processing with k-means cluster and principal component analysis," in *Fuzzy Logic and Applications*, I. Bloch, A. Petrosino, and A. G. B. Tettamanzi, Eds., vol. 3849 of *Lecture Notes in Computer Science*, pp. 267–276, 2006.

[14] C. Liao, S. Li, and Z. Luo, "Gene selection using Wilcoxon rank sum test and support vector machine for cancer classification," in *Computational Intelligence and Security*, Y. Wang, Y.-M. Cheung, and H. Liu, Eds., vol. 4456 of *Lecture Notes in Computer Science*, pp. 57–66, 2007.

[15] M. West, C. Blanchette, H. Dressman et al., "Predicting the clinical status of human breast cancer by using gene expression profiles," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 20, pp. 11462–11467, 2001.

[16] T. R. Golub, D. K. Slonim, P. Tamayo et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–527, 1999.

[17] E. B. Huerta, B. Duval, and J.-K. Hao, "A hybrid GA/SVM approach for gene selection and classification of microarray data," in *Applications of Evolutionary Computing*, F. Rothlauf, J. Branke, S. Cagnoni et al., Eds., vol. 3907 of *Lecture Notes in Computer Science*, pp. 34–44, Springer, Berlin, Germany, 2006.

[18] U. Alon, N. Barka, D. A. Notterman et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 12, pp. 6745–6750, 1999.

[19] K. C. Mondal, A. Mukhopadhyay, U. Maulik, S. Bandhyapadhyay, and N. Pasquier, "MOSCFRA: a multi-objective genetic approach for simultaneous clustering and gene ranking," in *Computational Intelligence Methods for Bioinformatics and Biostatistics*, R. Rizzo and P. J. G. Lisboa, Eds., vol. 6685 of *Lecture Notes in Computer Science*, pp. 174–187, Springer, Berlin, Germany, 2011.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[21] R. M. Luque-Baena, D. Urda, J. L. Subirats, L. Franco, and J. M. Jerez, "Analysis of cancer microarray data using constructive neural networks and genetic algorithms," in *Proceedings of the 1st International Work-Conference on Bioinformatics and Biomedical Engineering-IWBBIO*, Granada, Spain, March 2013.

[22] M. Mandal and A. Mukhopadhyay, "A novel PSO-based graph-theoretic approach for identifying most relevant and non-redundant gene markers from gene expression data," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 30, no. 3, pp. 175–192, 2015.

[23] O. Soufan, D. Kleftogiannis, P. Kalnis, and V. B. Bajic, "DWFS: a wrapper feature selection tool based on a parallel genetic algorithm," *PLoS ONE*, vol. 10, no. 2, Article ID e0117988, 2015.

[24] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, UK, 2nd edition, 1970.

[25] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, New York, NY, USA, 3rd edition, 1996.

[26] P. T. Spellman, G. Sherlock, M. Q. Zhang et al., "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization," *Molecular Biology of the Cell*, vol. 9, no. 12, pp. 3273–3297, 1998.

[27] A. Khamas, T. Ishikawa, K. Shimokawa et al., "Screening for epigenetically masked genes in colorectal cancer using 5-aza-2′-deoxycytidine, microarray and gene expression profile," *Cancer Genomics and Proteomics*, vol. 9, no. 2, pp. 67–75, 2012.

[28] T. Sato, A. Kaneda, S. Tsuji et al., "PRC2 overexpression and PRC2-target gene repression relating to poorer prognosis in small cell lung cancer," *Scientific Reports*, vol. 3, article 1911, 2013.

[29] D. Singh, P. G. Febbo, K. Ross et al., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, no. 2, pp. 203–209, 2002.

[30] M. A. Shipp, K. N. Ross, P. Tamayo et al., "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68–74, 2002.

[31] M. H. Cheok, W. Yang, C.-H. Pui et al., "Treatment-specific changes in gene expression discriminate in vivo drug response

in human leukemia cells," *Nature Genetics*, vol. 34, no. 1, pp. 85–90, 2003.

[32] J. C. Chang, E. C. Wooten, A. Tsimelzon et al., "Gene expression profiling for the prediction of therapeutic response to docetaxel in patients with breast cancer," *The Lancet*, vol. 362, no. 9381, pp. 362–369, 2003.