BIOINFORMATICS ALGORITHMS: TECHNIQUES AND APPLICATIONS

BIOINFORMATICS ALGORITHMS: TECHNIQUES AND APPLICATIONS

Ion Mandoiu and Alex Zelikovsky

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Bioinformatics Algorithms: Techniques and Applications / Mandoiu and Zelikovsky
Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

CHAPTER 1

CLASSIFICATION ACCURACY BASED MICROARRAY MISSING VALUE IMPUTATION

YI SHI, ZHIPENG CAI, GUOHUI LIN[1]

Department of Computing Science, University of Alberta
Edmonton, Alberta T6G 2E8, Canada

Gene expression microarray has been widely employed in biological and medical studies. In general, these studies involve the data analyses that require complete gene expression values, which, however, are not always possible due to various experimental factors. In the past several years, more than a dozen of methods have been proposed to impute the microarray missing values, and most of them adopt the (normalized) root mean squared errors to measure the imputation quality. Considering the fact that the purpose of missing value imputation is for downstream data analyses, and among which one of the most important applications is the genetic profiling, we propose to use the microarray sample classification accuracy based on the imputed expression values to measure the missing value imputation quality. Our extensive study on five imputation methods, from the most known ROWimpute and KNNimpute, to the most complexed BPCAimpute and SKNNimpute, to the most recent ILLSimpute, shows that BPCAimpute and ILLSimpute can fill in the missing values to achieve the sample classification accuracy as high as that can be achieved on the original complete expression data.

[1]Corresponding author.

1.1 INTRODUCTION

Microarrays, typically the high-density oligonucleotide arrays such as Affymetrix GeneChip oligonucleotide (Affy) arrays, can monitor the expression levels of thousands to tens of thousands of genes simultaneously. Such a technology provides a unique tool for systems biology, and has become indispensable in numerous biological and medical studies. One of the most common and important applications of gene expression microarray is to compare the gene expression levels in tissues under different conditions, such as wild-type versus mutant, or healthy versus diseased, for genetic profiling. In general, a subset of a small number of biomarkers, which are discriminatory genes whose expression levels either increase or decrease under certain conditions, can be identified and together they can be used to build a classifier that predicts the microarray sample class membership, such as disease subtype and treatment effectiveness.

Genetic profiling, as well as many other applications, involves microarray data analysis which requires complete and accurate gene expression values. However, in practice, such a requirement is often not satisfied due to a number of defects in microarray experiments. These defects include systemic factors such as insufficient resolution and uneven distribution of fluids, and stochastic factors such as image corruption, dust and scratches on the slides and glass flaws. All these could create the artifacts on the microarray chips which result in a certain percentage of expression data corruption [17, 18]. Even with the high-density oligonucleotide arrays such as Affymetrix GeneChip oligonucleotide (Affy) arrays, as high as 20% percentage of expression spots on the arrays could be blemished which may cover hundreds of probes and affect the reading of a considerable percent of gene expression values [17]. Microarray data analyses, such as gene clustering, biomarker identification, sample classification, and genetic and regulatory network prediction, which seek to address biological or medical issues, only accept complete expression values. Therefore, before the data analysis, the gene expression levels have to be preprocessed in order to impute the missing values, as well as correct some portion of the blemished data. In the past several years, more than a dozen of methods have been proposed for microarray missing value imputation, including ZEROimpute, ROWimpute and COLimpute [1, 18], KNNimpute and SVDimpute [18], BPCAimpute [13], GMCimpute [14], SKNNimpute [11], LSimpute [4], CMVE [16], LinCmb [8], LinImp [15], LLSimpute [10], and ILLSimpute [5].

When applying ZEROimpute, those logarithmic missing gene expression values are replaced by 0's [1, 18]. By arranging the microarray samples in the way that a row represents a gene and a column represents a sample, a microarray dataset (which contains a number of samples, each of which contains a common set of genes) can be effectively represented as an expression matrix. In ROWimpute, a missing entry is filled with the average expression level of the corresponding gene across all samples; In COLimpute, a missing entry is filled with the average expression level of all the genes in the corresponding sample.

With the advance of the microarray technology and its increasing number of applications, missing value imputation attracts more attention and several more complexed imputation methods have been proposed, differing in pivotal ideas. Singular Value Decomposition (SVDimpute) and the weighted K-Nearest Neighbor (KNNimpute) missing imputation methods are proposed by Troyanskaya *et al* [18]. In SVDimpute, a set of mutually orthogonal expression patterns are obtained and linearly combined to approximate the expressions of all genes, through the singular value decomposition of the expression matrix. By selecting the K most significant eigengenes, a missing value in the target gene is estimated by first regressing the target gene against these K eigengenes and then using the coefficients of the regression to estimate the missing value from the linear combination of the K eigengenes.

In KNNimpute method, for a target gene, its K nearest neighbor genes (or rows) which do not contain missing values in the same columns as the target gene, are selected. Then the missing values in the target gene are estimated by a weighted linear combination of the K nearest neighbor genes, where the weights are calculated as the inverse of the distances between the target gene expression vector and the neighbor gene expression vectors.

Similar to KNNimpute, the Least Square imputation (LSimpute) method is proposed by Bø *et al* [4]. It utilizes the least square principle to determine the weights in the linear combination of the K nearest neighbors, from which the missing values in the target gene are estimated. Different from LSimpute where nearest neighboring genes are used, the local least square missing value imputation (ILLSimpute), proposed by H. Kim *et al* [10], estimates the missing values using the coherent genes under the Pearson correlation coefficients. Oba *et al* [13] proposed a microarray missing value imputation method based on Bayesian Principal Component Analysis (BPCAimpute). BPCAimpute essentially employs three elementary processes, principal component regression, Bayesian estimation, and an expectation-maximization-like repetitive algorithm. It estimates the latent parameters for a probabilistic model under the framework of Bayesian inference and estimates the missing values using the model. Ouyang *et al* [14] proposed GMCimpute method, which applies the idea of Gaussian Mixture Clustering and model averaging. CMVE, a Collateral Missing Value Estimation, is proposed by Sehgal *et al* [16], in which for a missing value entry, it first calculates several missing value estimates according to different scoring functions and then the overall estimate is distilled from these estimates.

There are several extensions or variants to the above imputation methods. For example, SKNNimpute, or Sequential K-Nearest Neighbor imputation, is proposed by K.-Y. Kim *et al* [11]. SKNNimpute sequentially imputes missing values from genes with the least number of missing entries to genes with the most number of missing entries. Within each iteration of SKNNimpute, the KNNimpute method is executed to impute the missing values in the target gene, where only those genes who have no missing value or whose missing values have already been imputed are the candidates of being neighbors. LinImp, which fits a gene expression value into a linear model concerning four factors, is proposed by Scheel *et al* [15]. LinCmb, which is a convex combination of several imputation methods, is proposed by Jörnsten *et al* [8]. Most recently, Cai *et al* [5] proposed an iterated version of LLSimpute, the ILLSimpute method, for missing value imputation.

Among the above mentioned more than a dozen imputation methods, some of them have been compared with each other. In fact, most of the complexed methods have been compared with ROWimpute and KNNimpute. These comparative studies all adopt a measurement called the *Root Mean Square Error* (RMSE), or its normalized variant NRMSE. Let $E = \{E_1, E_2, \ldots, E_t\}$ denote the missing entries in the microarray expression matrix. For each missing entry E_i, $i = 1, 2, \ldots, t$, let e_i^* and e_i denote the corresponding true expression value and the imputed expression value, respectively. The mean of the squared errors is calculated as

$$\mu^2 = \frac{1}{t}\sum_{i=1}^{t}(e_i - e_i^*)^2.$$

The mean of these t true expression values is

$$\bar{e} = \frac{1}{t}\sum_{i=1}^{t} e_i^*,$$

and the standard deviation is

$$\sigma = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(e_i^* - \bar{e})^2}.$$

The NRMSE of the involved imputation method on this expression matrix is defined as the ratio of μ over σ, i.e., NRMSE = $\frac{\mu}{\sigma}$.

Note that when the expression matrix is given, σ is given as a constant. Therefore, according to the definition of NRMSE, it is obvious that a smaller NRMSE value indicates a better imputation quality. The existing comparison studies show that, under the RMSE or the NRMSE measurement, some of the above imputation methods consistently performed better than the others [13, 4, 11, 14, 8, 10, 15, 16, 5]. Typically, in the most recent study in [5], it is shown that BPCAimpute and ILLSimpute are both efficient and effective, regardless of the microarray dataset type (non-time series, time series dataset with low noise level, noisy time series) or missing value rate.

The NRMSE measurement presumes that all the observed gene expression levels accurately measure the hybridization intensities of the genes or probes on the microarray chips. Unfortunately, however, this is not always the case. Gene expression microarray is considered as a useful technology to provide expression profiles or patterns correlated to the conditions, but the expression levels of individual genes might not be all accurate. As we mentioned earlier, even on the high-density oligonucleotide arrays such as Affymetrix GeneChip oligonucleotide (Affy) arrays, a significant percentage of chips could be blemished, and therefore in the gene expression values, a high percentage of them may be noisy or even should be treated as missing. Nevertheless, the boundary between noisy data or missing data is often difficult to determine, which red flags the use of only the RMSE or the NRMSE to measure the imputation quality. It has been suggested that, with known gene cluster information, one may use the percentage of mis-clustered genes as a measurement of imputation quality, in addition to NRMSE [14].

Note that in most of the existing missing value imputation methods, either implicitly or explicitly, the missing values in the target gene are estimated using the similarly expressed genes, the neighbors or the coherent genes. In this sense, it seems that using gene cluster information in final imputation quality measurement does not really tell much more than RMSE and NRMSE. Since one of the most important applications of gene expression microarray is for genetic profiling of the distinct experimental conditions, for example for disease subtype recognition and disease treatment classification, we propose to adopt one downsteam microarray data analysis, microarray sample classification, and to use the classification accuracy to measure the quality of imputed expression values. The main impact of using classification accuracy as a new measurement is that in general the imputed expression values themselves are not interesting, while whether or not the imputed expression matrix can be used in downstream applications is the major concern. To demonstrate that using classification accuracy is indeed a good measurement, we include two most known imputation methods ROWimpute and KNNimpute, two most complexed methods BPCAimpute and SKNNimpute, and the most recently proposed method ILLSimpute in our comparative study. The computational results on two real cancer microarray datasets with various simulated missing rates show that both BPCAimpute and ILLSimpute can impute the missing values such that the classification accuracy achieved on the imputed expression matrix is as high as that can be achieved on the original complete expression matrix, while the other methods do not seem to perform well. Some of these results are consistent with the previous experiments based solely on NRMSE measurement. One tentative conclusion we

may draw from this study is that, for the purpose of microarray sample classification, both BPCAimpute and ILLSimpute have already achieved perfect performance and probably there is nothing left to do in terms of missing value imputation.

The rest of this chapter is organized as follows: In the next section, those five representative missing value imputation methods included in this study, ROWimpute, KNNimpute, BPCAimpute, SKNNimpute, and ILLSimpute, will be briefly introduced. The task of microarray sample classification, and its associated gene selection, is also introduced, where we present four representative gene selection methods, F-test, T-test, CGS-F-test, and CGS-T-test. We also briefly describe two classifiers built on the selected genes, the K Nearest Neighbor (KNN) classifier and the Support Vector Machine (SVM) classifier, along with the definition of classification accuracy. The descriptions of the two real cancer microarray datasets and all the computational results are presented in Section 3. We discuss our results in Section 4. Specifically, we examine the impacts of the adopted gene selection methods. Section 5 summarizes our conclusions.

1.2 METHODS

We assume there are p genes in the microarray dataset under investigation, and there are in total n samples/chips/arrays. Let a_{ij} denote the expression level of the i-th gene in the j-th sample, which takes U if it is a missing entry. The expression matrix representing this microarray dataset is

$$A_{p\times n} = (a_{ij})_{p\times n}.$$

Let $E = \{E_1, E_2, E_3, \ldots, E_t\}$ be the set of all missing value entries in the expression matrix, where t records the number of missing entries. The *missing rate* of the dataset is calculated as $r = \frac{t}{p\times n}$. In real microarray datasets, r ranges from 0% to as high as 20%.

1.2.1 The Imputation Methods

There are more than a dozen of microarray missing value imputation methods proposed in the past several years, adopting different mathematical models. For example, ZEROimpute, ROWimpute and COLimpute are quite similar in the sense that they are simple and do not assume any correlations among the genes, neither the samples. The SVDimpute and KNNimpute are probably the first non-trivial ones, where SVDimpute looks for dependencies while KNNimpute seeks the help from neighbors. With various possible extensions, generalizations, or modifications, LSimpute, LLSimpute and LinImp are similar to KNNimpute in the essence; BPCAimpute, GMCimpute and CMVE are similar to SVDimpute. SKNNimpute applies sequential imputation, trying to use the data in decreasing reliability, and ILLSimpute implements iterated imputation intending to improve the quality stepwise. For this reason, we only include ROWimpute, KNNimpute, BPCAimpute, SKNNimpute, and ILLSimpute as representatives in this study. Note that most of these imputation methods need the notion of expression similarity between two genes, which is defined in the following.

Given a *target gene* that contains missing value entries to be estimated and a *candidate gene* (which should have known expression values corresponding to these missing value entries in the target gene), all of the missing value entries in the candidate gene are temporarily filled with the average expression value (row average). Then, by ignoring the same columns in both the target gene and the candidate gene, corresponding to the missing value entries in the target gene, we obtain two expression (sub-) vectors with no missing

entries. The Euclidean distance between these two vectors is computed and it is taken as the distance between the target gene and the candidate gene. For example, if the target gene is (U, 1.5, U, 2.0, -1.2, U, 2.8) and the candidate gene is (1.6, U, U, -0.4, 2.2, 3.8, U), where U denotes a missing value, then the row average for the candidate gene is $\frac{1}{4}(1.6 - 0.4 + 2.2 + 3.8) = 1.8$; and the two vectors we obtain are (1.5, 2.0, -1.2, 2.8) and (1.8, -0.4, 2.2, 1.8); and the distance between these two genes is $\sqrt{18.41} = 4.29$ [5]. In KNNimpute, the K closest candidate genes to the target gene are selected as the neighbors, or *coherent* genes, of the target gene, where K is pre-specified and it is set at 10 in most of its implementations [18, 11]. Suppose the target gene is i and its neighbors are $i_1, i_2, \ldots, i_K$. Let d_k denote the distance between gene i and gene i_k for $1 \leq k \leq K$. Then the missing value $a_{i,j}$ in the target gene i is estimated as

$$a_{i,j} = \sum_{k=1}^{K} \frac{1}{d_k} a_{i_k,j}.$$

Note that in the above version of KNNimpute, coherent genes are determined with respect to the target gene. Another version of KNNimpute is to determine coherent genes to the target gene with respect to one missing value entry. In this study, we examine the former version. In SKNNimpute, the missing value imputation is done sequentially and at every iteration, the gene containing the least number of missing value entries is chosen as the target gene, and KNNimpute is applied to estimate the missing values in this target gene where only those genes who have no missing values or whose missing values have already been imputed are considered as candidate genes. The K value in this internal KNNimpute is also set to 10 [11].

In LLSimpute [10], the coherent genes to a target genes are similarly determined but using the Pearson correlation coefficients rather than the Euclidean distance (in LSimpute), and its number is also pre-specified. Afterwards, the target gene is also represented as a linear combination of its coherent genes, where the linear combination is done through a local least square. Essentially, coefficients in this linear combination are set in the way that the sum of the square differences between the known expression values in the target gene and the linear combination of coherent genes is minimized. Though LLSimpute has a process to learn what the best number of coherent genes would be, this number remains the same for all target genes. Cai *et al* [5] realized that for distinct target genes, the distances between it and its coherent genes vary a lot, and consequently it is not wise to set a uniform number of coherent genes for all target genes. Instead, they proposed to learn a dataset dependent distance ratio threshold δ such that only candidate genes whose distances to the target genes within the threshold are considered as coherent genes. In addition, they proposed to iteratively re-impute the missing values using the imputation results from the last iteration, where LLSimpute is called, for a number of iterations or till the imputed values converge.

The missing value estimation method based on Bayesian Principle Component Analysis (BPCAimpute) consists of three primary progresses. They are (1) principle component regression, (2) Bayesian estimation, and (3) an expectation-maximization-like repetitive algorithm [13]. Given the gene expression matrix, the principle component regression seeks to represent every n-dimensional gene expression vector of gene i $a_i = \langle a_{i1}, a_{i2}, \ldots, a_{in} \rangle$ as a linear combination of K principal axis vectors a_{l_k}, $1 \leq k \leq K$:

$$a_i = \sum_{k=1}^{K} x_{l_k} a_{l_k} + \epsilon_i,$$

where K is a relatively small number $(K < n)$, $x_{l_k}(1 \leq k \leq K)$ are the coefficients, or the so called *factor scores*, and ϵ_i denotes the *residual error* associated with gene i. By using a pre-specified value of K, the principle component regression obtains x_{l_k} and a_{l_k} such that the sum of squared error $\|\epsilon\|^2$ over the whole dataset is minimized [13]. In Bayesian estimation process, the residual errors $\epsilon_i(1 \leq i \leq p)$ and the factor scores $x_{l_k}(1 \leq k \leq K)$ are assumed to obey normal distributions at first. Then, the Bayesian estimation is used to obtain the posterior distribution parameters according to the Bayes theorem. In the last process, an expectation-maximization-like repetitive algorithm is applied to estimate or re-estimate the missing values until the imputed results converge or the repetitive process attains the pre-specified iteration numbers.

1.2.2 The Gene Selection Methods

For microarray sample classification purpose, normally an expression matrix is provided with every sample labeled by its class. Such a dataset is used as the training dataset to learn the genetic profiles associated with each class, and subsequently whenever a new sample comes, its class membership can be predicted. One can use all the genes to compose the genetic profiles, but as there are usually thousands of genes involved in the study while only tens of samples in a class, a process called *gene selection* is conducted to selected a subset of discriminatory genes that are either over-expressed or under-expressed. Such a subset of genes are then fed to construct a classifier which can predict the class membership of a new sample.

There is a rich literature on general feature selection. Microarray gene selection only attracts attention since the technology becomes high-throughput. Nevertheless, gene selection has its unique characteristics, which make itself distinct from the general feature selection. Many gene selection methods have been proposed in the past decade, though they all center at how to measure the class discrimination strength for a gene. F-test method [2, 3] tries to identify those genes that have the greatest *inter-class variances* and the smallest *intra-class variances*. It scores a gene by the ratio of its inter-class variance over its intra-class variance — a greater score indicates a higher discrimination power the gene has. F-test method sorts all the genes in the non-increasing score order and returns a pre-specified number of top ranked genes. In T-test method [19], each gene has a score that is the classification accuracy of the classifier built on the single gene, and it returns also a pre-specified number of top scored genes. Within our group, several gene selection methods have been proposed, among which one of the key ideas is to select only those genes that do not have overlapping class discrimination strength. The intention is that using genes having similar class discrimination strength in building classifiers would be redundant. To this purpose, we proposed to firstly cluster the genes under some measurements of class discrimination strength, and then limit the number of genes per cluster to be selected. Combining this gene clustering idea with F-test and T-test, we have CGS-Ftest and CGS-Ttest gene selection methods. We use these four gene selection methods, F-test, T-test, CGS-F-test, and CGS-T-test, in this study.

1.2.3 The Classifiers

Two classifiers are adopted in this study. One is the K-Nearest Neighbor (KNN) classifier [6] and the other is a linear kernel Support Vector Machine (SVM) classifier [7]. The KNN-classifier predicts the class membership of a testing sample by using the expression values of (only) the selected genes. It identifies the K closest samples in the training dataset and

then uses the class memberships of these K similar samples through a majority vote. In our experiments, we set the default value of K to be 5, after testing K from 1 to 10. The SVM-classifier, which contains multiple SVMs, finds decision planes to best separate (soft margin) the labeled samples based on the expression values of the selected genes. It uses this set of decision planes to predict the class membership of a testing sample. One may refer to Guyon *et al* [7] for more details of how the decision planes are constructed based on the selected genes.

1.2.4 The Performance Measurements

At the end of experimental results, we will plot the NRMSE values for all imputation methods on the respective datasets. In this study, our main purpose is to demonstrate that using the microarray sample classification accuracy is another very effective measurement. Given a complete gene expression matrix with all samples being labeled with their classes, we adopt the *ℓ-fold cross validation* to avoid possible data overfitting. To this purpose, the complete dataset is randomly partitioned into ℓ equal parts, and $(\ell - 1)$ parts of them are used to form the *training dataset*, while the other part forms the *testing dataset* in which the class labels of the samples are removed. The predicted class memberships for the testing samples are then compared with the true ones to determine whether or not the prediction is correct. The process is repeated for each part. The percentage of the correctly predicted samples is the *classification accuracy* of the classifier. In this study, we report the experimental results on the 5-fold cross validation, where the partition process is repeated for 10 times. Consequently, the final classification accuracy is the average over 50 testing datasets. We remark that ℓ-fold cross validations for $\ell = 3, 7, 9, 11$ present similar results (data not shown).

1.2.5 The Complete Work Flow

To demonstrate that microarray sample classification accuracy is a very effective measurement for the imputation methods, we simulated missing values in the original complete gene expression matrix. On both the original and the imputed gene expression matrices, the sample classification was done by a classifier, whose classification accuracies were recorded and compared. In more details, given a complete microarray gene expression matrix containing p genes and n samples in L classes, we adopted 5-fold cross validation scheme to collect the sample classification accuracies for each of the four gene selection methods, F-test, T-test, CGS-Ftest, and CGS-Ttest, combined with the KNN-classifier and the SVM-classifier. The number of selected genes, x, ranges from 1 to 80. These accuracies are on the original dataset.

Next, for each of the missing rates $r = 1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%, 20\%$, we picked randomly $r \times p \times n$ entries from the original gene expression matrix and erased them to form a dataset containing missing values. The ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute, were called separately on the simulated dataset to estimate the missing values. After imputing the missing values in the simulated gene expression matrix, the subsequent procedure was the same as that for the original complete gene expression matrix in the above to collect the sample classification accuracies. For each missing rate, the missing value simulation was repeated for 10 times, and consequently the associated accuracies are the average over 500 entities.

To summarize, by regarding the original complete dataset as a dataset of 0% missing values, we have 9 missing rates, each associated with 10 simulated datasets (except 0%),

5 imputation methods, 4 gene selection methods, and 2 classifiers, under the 5-fold cross validation scheme, which is repeated for 10 times.

1.3 EXPERIMENTAL RESULTS

Given a complete microarray gene expression dataset (regarded as a dataset of 0% missing values), we simulated 10 datasets for each of the missing rates $r = 1\%$, 2%, 3%, 4%, 5%, 10%, 15%, 20%. On each simulated dataset, all five missing data imputation methods, ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute, were run separately to estimate the missing values. Afterwards, on either the original complete dataset or the imputed complete dataset, each gene selection method (F-test, T-test, CGS-Ftest, and CGS-Ttest) was called on randomly picked 80% samples to output x genes, for $x = 1, 2, \ldots, 80$. Each of the KNN-classifier and the SVM-classifier was then built on these x selected genes to predict the class memberships for the other 20% samples. The final classification accuracy was collected for further statistics.

We include two real cancer microarray gene expression datasets, SRBCT dataset [9] and GLIOMA dataset [12], in this study.

1.3.1 Dataset Descriptions

The SRBCT dataset [9] contains 83 samples in total, in four classes, *the Ewing family of tumors*, *Burkitt lymphoma*, *neuroblastoma*, and *rhabdomyosarcoma*. Every sample in this dataset contains $2,308$ gene expression values after data preprocessing. Among the 83 samples, 29, 11, 18, and 25 samples belong to the four classes, respectively.

The GLIOMA dataset[12] contains in total 50 samples in four classes, *cancer glioblastomas*, *non-cancer glioblastomas*, *cancer oligodendrogliomas*, and *non-cancer oligodendrogliomas*, which have $14, 14, 7$, and 15 samples, respectively. This dataset is known to have a lower quality for sample classification [12, 20]. In the preprocessing, for each gene, we calculated its expression standard deviation over all samples, and those genes with standard deviation lower than a threshold were filtered. Such a gene filtering is based on the intuition that if the expression standard deviation of a gene is too small, it may not have too much discrimination strength and thus is less likely to be selected by any gene selection method. After the preprocessing, we obtained a dataset with $3,550$ genes.

1.3.2 5-Fold Cross Validation Classification Accuracies

For each combination of a gene selection method and a classifier, its sample classification accuracy is the average over 50 testing datasets on the original gene expression dataset, and over 500 testing datasets on each of the missing rates $r = 1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%, 20\%$, under the 5-fold cross validation scheme. For ease of presentation, we concatenate the sequentially applied method names to denote the associated 5-fold cross validation classification accuracy. For example, ILLSimpute-CGS-Ftest-SVM denotes the accuracy that is achieved by applying the ILLSimpute method, followed by the CGS-Ftest to select a certain number of genes for building an SVM-classifier for testing sample membership prediction. Our further statistics include the sample classification accuracies with respect to a missing value imputation method, a gene selection method, the gene clustering based gene selection or the other, and a classifier, to be detailed in the following. For example, ILLSimpute-SVM denotes the average accuracy over all four gene selection methods, that is

achieved by applying the ILLSimpute method, followed by a gene selection method to select a certain number of genes for building an SVM-classifier for testing sample membership prediction.

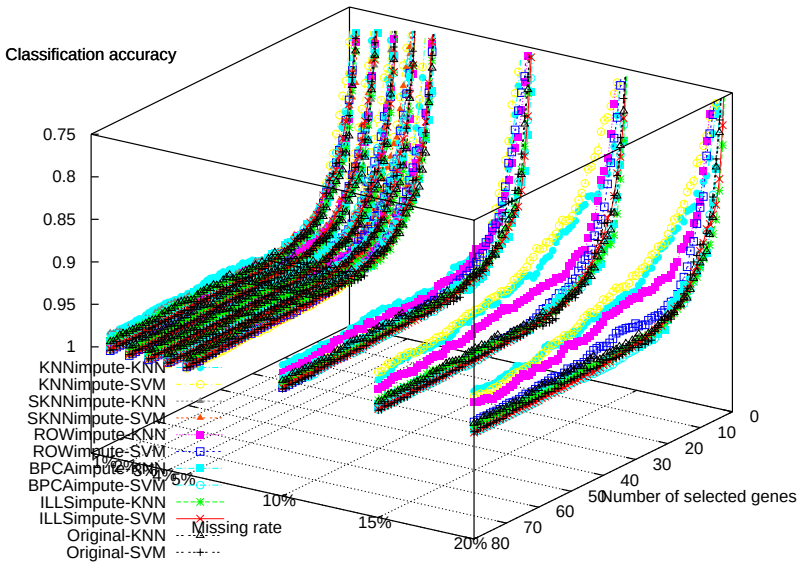

Figure 1.1. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the F-test method, on the original and simulated SRBCT dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original SRBCT dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

1.3.2.1 The SRBCT Dataset For each of the four gene selection methods, F-test, T-test, CGS-Ftest, and CGS-Ttest, we plotted separately the 5-fold cross validation classification accuracies for all combinations of a missing value imputation method and a classifier, on the original SRBCT dataset ($r = 0\%$, in which the missing value imputation methods were skipped) and simulated datasets with missing rates 1%, 2%, 3%, 4%, 5%, 10%, 15% and 20%, respectively. We chose to plot these classification accuracies in three dimensional where the x-axis is the number of selected genes, the y-axis is the missing rate, and the z-axis is the 5-fold cross validation classification accuracy. Figures 1.1., 1.2., 1.3., and 1.4. plot these classification accuracies for the F-test, T-test CGS-Ftest, and CGS-Ttest methods, respectively. Note that for the SKNNimpute method, if it cannot find K (in our experiments, $K = 10$) Nearest Neighbors which satisfy the candidate gene requirements, then it was not applied on the particular simulated dataset and more had to be simulated. Nevertheless, it has to be mentioned that once missing rate was higher than 5%, SKNNimpute failed quite often, and as a consequence, we did not have all the results for SKNNimpute on missing rates greater than 5%.

From Figures 1.1., 1.2. 1.3., and 1.4., we can see that when the missing rate is less than or equal to 5% (the five groups of plots to the left), all five imputation methods worked

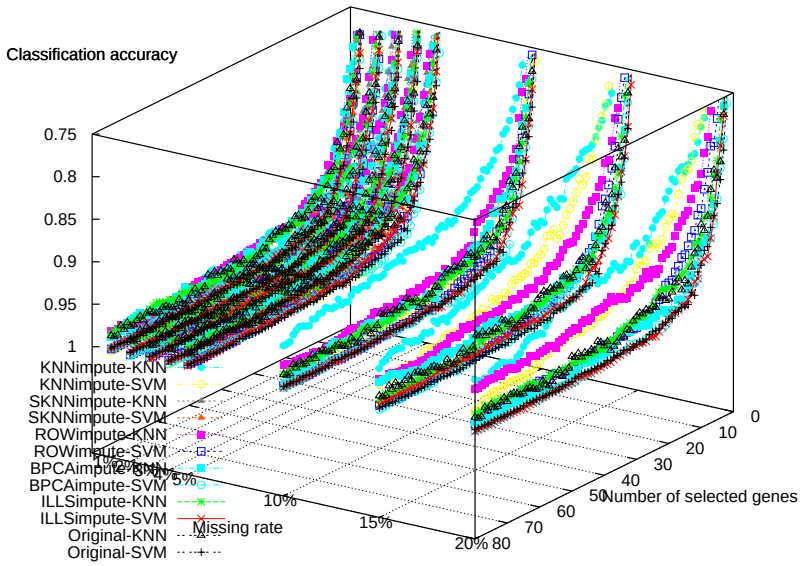

Figure 1.2. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the T-test method, on the original and simulated SRBCT dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original SRBCT dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

almost equally well, combined with either of the two classifiers, compared to the baseline classification accuracies on the original SRBCT dataset. However, the plots started to diverge when the missing rate increases to 10%, 15% and 20%. For example, besides the observation that the classification accuracies of the SVM-classifier were a little higher than that of the KNN-classifier (this is more clear with the T-test method, in Figure 1.2. and the right plot in Figure 1.5.), combined with the same imputation method. Overall, the general tendencies are that 1) ROWimpute performed slightly better than KNNimpute, 2) ILLSimpute and BPCAimpute performed the best among the five methods, and 3) the gaps between the performances became larger with increased missing rate r. For missing rate $r = 20\%$, the classification accuracies are separately plotted in Figure 1.5. and Figure 1.6., in each of which the left plot is for the F-test/CGS-Ftest method and the right plot is for the T-test/CGS-Ttest method (no SKNNimpute results were available). It is clearly seen that, the BPCAimpute and ILLSimpute methods performed consistently the best, the ROWimpute method performed slightly better than the KNNimpute method, and the imputed datasets by BPCAimpute and ILLSimpute had almost the same quality as the original SRBCT dataset, in terms of the final sample classification accuracy. Furthermore, the last observation holds true across all missing rates, a strong demonstration that BPCAimpute and ILLSimpute are the methods of choices for microarray missing value imputation.

All of the above plots show that in general the KNN-classifier performed a little worse than the SVM-classifier on the SRBCT dataset. However, we remark that it is not necessarily

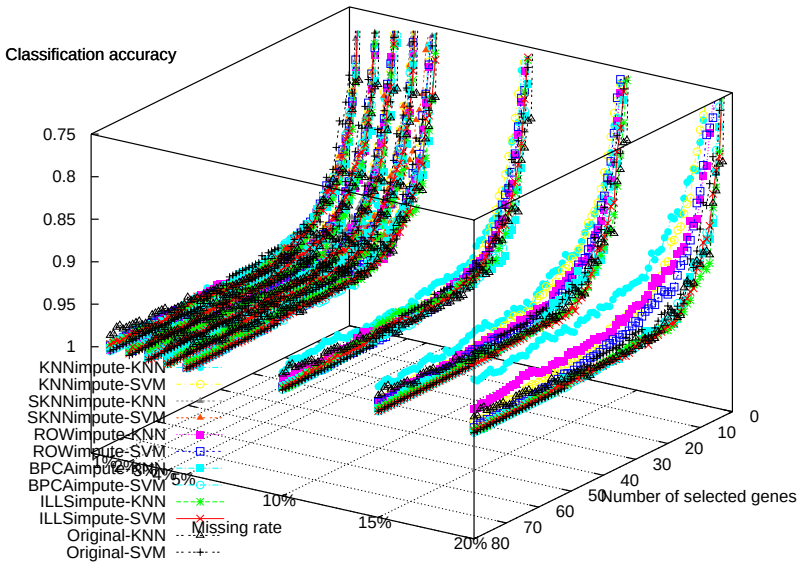

Figure 1.3. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the CGS-Ftest method, on the original and simulated SRBCT dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original SRBCT dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

the case that the KNN-classifier is always inferior (cf. [5]). By ignoring the detailed gene selection method and the classifier to calculate the classification accuracy of a missing value imputation method as the average over 8 values, corresponding to in total 8 combinations of a gene selection method and a classifier. We associated this classification accuracy with each of the five imputation methods. Figure 1.7. plots these classification accuracies on the SRBCT dataset, with missing rate $r = 0\%$ (the original dataset), 1%, 2%, 3%, 4%, 5%, 10%, 15% and 20%, respectively. Again, SKNNimpute only applied to missing rates less then or equal to 5%. From this 3D plot, one can see again that essentially there was not much performance difference between the five missing value imputation methods when the missing rate r was less than or equal to 5% (the five groups of plots to the left); But their performances started to diverge when $r \geq 10\%$, and again the general tendencies are that 1) ROWimpute perform slightly better than KNNimpute, 2) BPCAimpute and ILLSimpute performed the best, and 3) the gaps between the performances became larger with increased missing rate r. Similarly, for missing rate $r = 20\%$, the average classification accuracies are separately plotted in Figure 1.8. (no SKNNimpute results were available), where once again one can see that ROWimpute performed slightly better than KNNimpute and BPCAimpute and ILLSimpute performed the best. Furthermore, in terms of classification accuracy, the imputed expression matrices by BPCAimpute and ILLSimpute had the same quality as the original expression matrix.

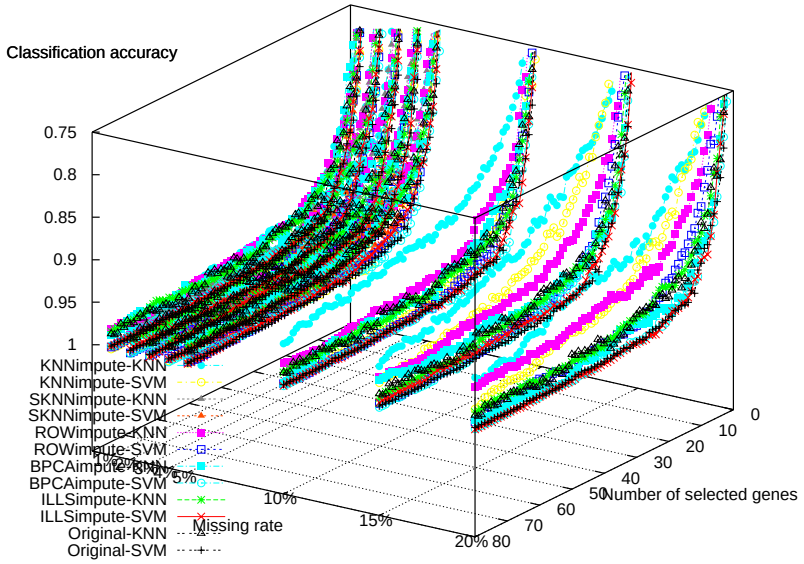

Figure 1.4. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the CGS-Ttest method, on the original and simulated SRBCT dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original SRBCT dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

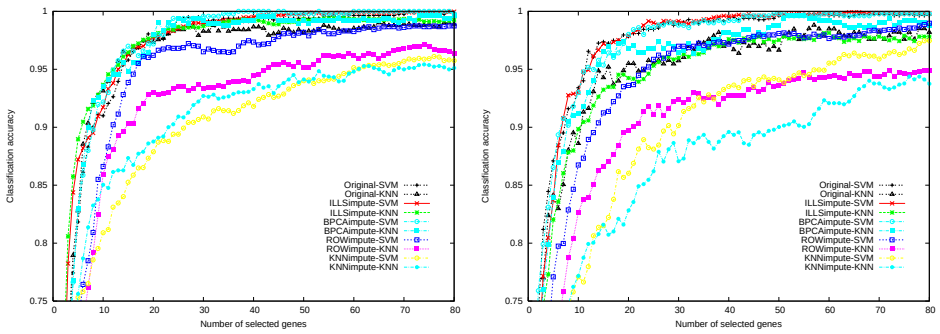

Figure 1.5. F-test (left) and T-test (right) performance on the SRBCT dataset simulated with missing rate $r = 20\%$.

1.3.2.2 The GLIOMA Dataset It has been recognized that the quality of the GLIOMA dataset is lower than that of the SRBCT dataset[12, 20]. Similarly, for each of the four gene selection methods, F-test, T-test, CGS-Ftest, and CGS-Ttest, we plotted separately the 5-fold cross validation classification accuracies for all combinations of a missing value imputation method and a classifier, on the original dataset ($r = 0\%$) and simulated datasets

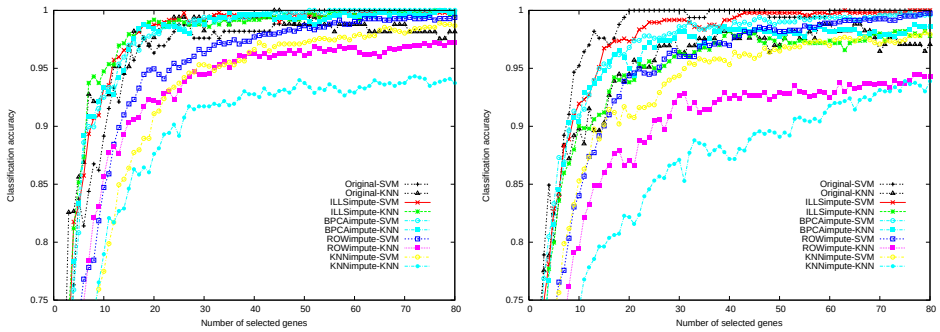

Figure 1.6. CGS-Ftest (left) and CGS-Ttest (right) performance on the SRBCT dataset simulated with missing rate $r = 20\%$.

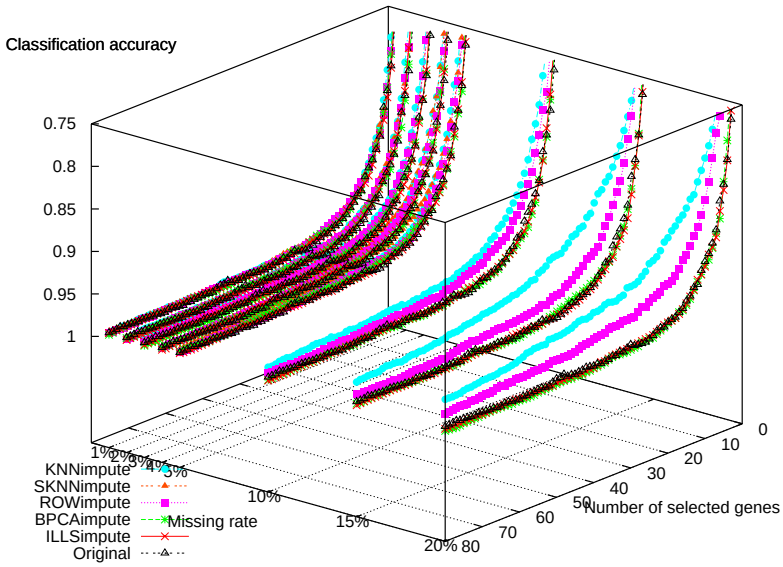

Figure 1.7. The 5-fold classification accuracies, averaged over 8 combinations of a gene selection method and a classifier, on the SRBCT dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the average classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original plots the average classification accuracies achieved on the original SRBCT dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute to missing rate less than or equal to 5%.

with missing rates 1%, 2%, 3%, 4%, 5%, 10%, 15% and 20%, respectively. Figures 1.9., 1.10., 1.11., and 1.12. plot these classification accuracies for the F-test, T-test CGS-Ftest, and CGS-Ttest methods, respectively. Again, we did not have complete results for SKNNimpute when the missing rate is greater than 5%. From these plots, we can see that the performances of all five imputation methods differed a lot on every missing rate, and more

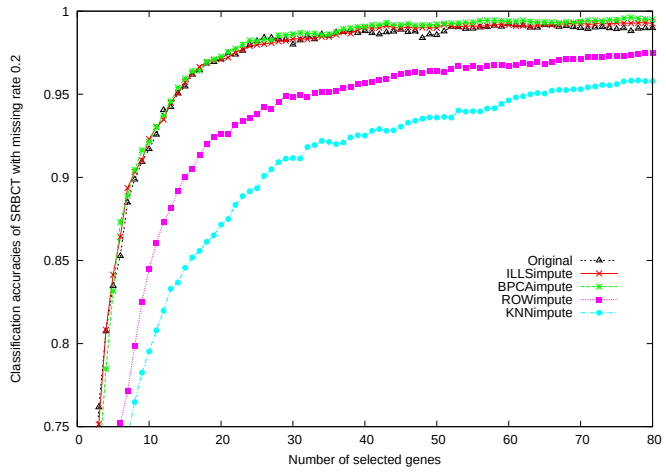

Figure 1.8. Average performances of ROWimpute, KNNimpute, BPCAimpute, and ILLSimpute methods, in terms of classification accuracies, on the SRBCT dataset with missing rate $r = 20\%$. The average classification accuracies on the original SRBCT dataset are also plotted for comparison purpose.

significantly with increasing missing rates. Nonetheless, overall, the general tendencies are still that 1) ROWimpute performed slightly better than KNNimpute, 2) ILLSimpute and BPCAimpute performed the best among the five methods and the imputed datasets by them have the same quality as the original one, in terms of the sample classification accuracy, and 3) the gaps between the performances became larger with increased missing rate r.

For missing rate $r = 20\%$, the classification accuracies are separately plotted in Figure 1.13. and Figure 1.14., in each of which the left plot is for the F-test/CGS-Ftest method and the right plot is for the T-test/CGS-Ttest method (no SKNNimpute results were available). It is clearly seen that, the BPCAimpute and ILLSimpute methods performed consistently the best, the ROWimpute method performed slightly better than the KNNimpute method, and the imputed datasets by BPCAimpute and ILLSimpute had almost the same quality as the original GLIOMA dataset, in terms of the final sample classification accuracy. Furthermore, the last observation holds true across all missing rates, a strong demonstration that BPCAimpute and ILLSimpute are the methods of choices for microarray missing value imputation.

1.4 DISCUSSION

1.4.1 Gene Selection Methods

Clearly, the detailed gene selection method adopted in the study will result in different final sample classification accuracy. The collected average classification accuracies were taken over all four gene selection methods, F-test, T-test, CGS-F-test, and CGS-T-test, and thus it is more convincing to conclude that BPCAimpute and ILLSimpute performed the best. We also compared the performances of the these four adopted gene selection methods by calculating their average classification accuracies over all the four missing value imputation methods ROWimpute, KNNimpute, BPCAimpute, and ILLSimpute (SKNNimpute was

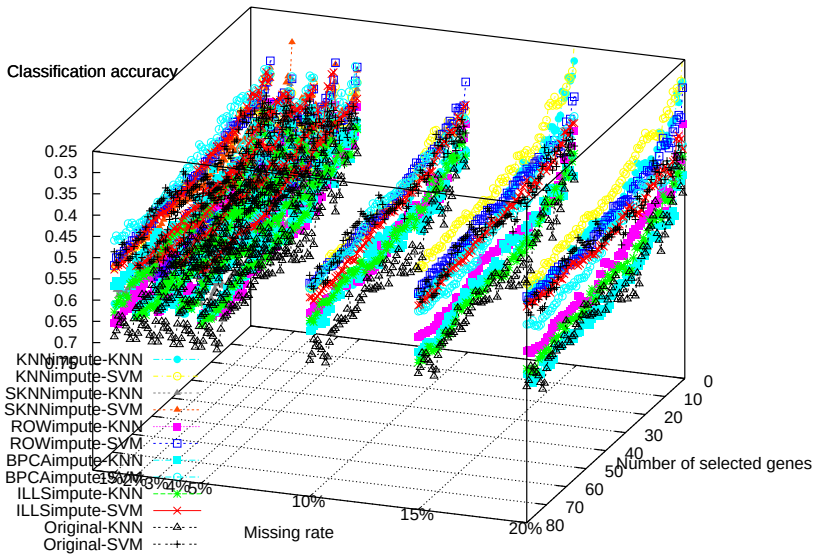

Figure 1.9. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the F-test method, on the original and simulated GLIOMA dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original GLIOMA dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

excluded because it did not apply to large missing rates). These classification accuracies and the classification accuracies obtained on the original SRBCT dataset are plotted in Figure 1.15. and Figure 1.16.

From these two plots, we can say that on the SRBCT dataset, F-test/CGS-Ftest performed slightly better than T-test/CGS-Ttest and CGS-Ftest/CGS-Ttest performed slightly better than F-test/T-test, respectively.

1.4.2 NRMSE Values

We have also collected the NRMSE values for the five imputation methods on the simulated SRBCT datasets with all missing rates, which are plotted in Figure 1.17. They again indicate that ROWimpute performed slightly better than KNNimpute (and SKNNimpute) and ILLSimpute and BPCAimpute performed the best among the five methods.

1.5 CONCLUSIONS

The performances of missing value imputation methods, BPCAimpute and ILLSimpute, have previously been shown to be better than most recent similar developments, using the NRMSE measurement [5]. The performance difference becomes significant when the missing rate is large. We realized that microarray gene expression data though is able to

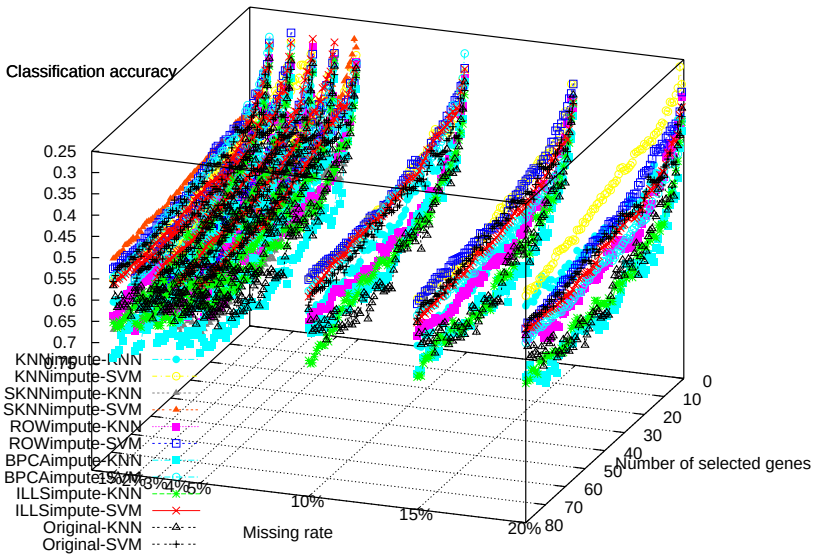

Figure 1.10. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the T-test method, on the original and simulated GLIOMA dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original GLIOMA dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

provide a global picture on the genetic profile, yet some portion of it is not reliable due to various experimental factors. Consequently, using solely the NRMSE measurement could sometimes be misleading. Considering the fact that missing value imputation is for the downstream data analysis, among which one of them is the sample classification, we proposed to adopt the classification accuracy as another measurement of imputation quality. Our simulation study on two real cancer microarray datasets, to include 5 imputation methods, 4 gene selection methods, and 2 classifiers, demonstrated that classification accuracy is a very effective measurement, and further confirmed that BPCAimpute and ILLSimpute are the best imputation methods. Furthermore, the imputed gene expression dataset can reach the same sample classification accuracy as that can be achieved on the original dataset.

Acknowledgments

This research is supported in part by CFI, iCore, and NSERC.

REFERENCES

1. A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson Jr, L. Lu,

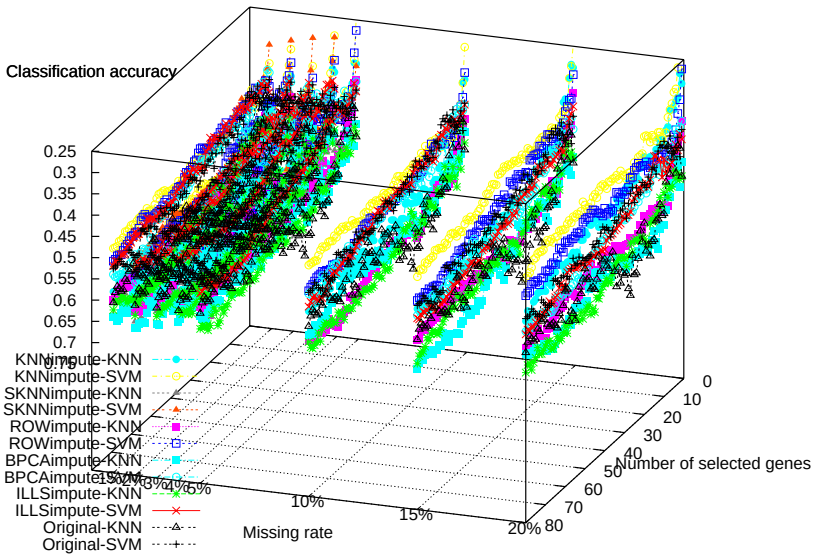

Figure 1.11. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the CGS-Ftest method, on the original and simulated GLIOMA dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original GLIOMA dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, L. M. Staudt, and *et al*. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

2. P. Baldi and A. D. Long. A Bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 17:509–519, 2001.
3. A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, S. Monti, P. Vasa, C. Ladd, J. Beheshti, R. Bueno, M. Gillette, M. Loda, G. Weber, E. J. Mark, E. S. Lander, W. Wong, B. E. Johnson, T. R. Golub, D. J. Sugarbaker, and M. Meyerso. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of National Academy of Sciences of the United States of America*, 98:13790–13795, 2001.
4. T. H. Bø, B. Dysvik, and I. Jonassen. LSimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research*, 32:e34, 2004.
5. Z. Cai, M. Heydari, and G.-H. Lin. Iterated local least squares microarray missing value imputation. *Journal of Bioinformatics and Computational Biology*, 4(5), 2006.
6. S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97:77–87, 2002.
7. I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.
8. R. Jörnsten, H.-Y. Wang, W. J. Welsh, and M. Ouyang. DNA microarray data imputation and significance analysis of differential expression. *Bioinformatics*, 21:4155–4161, 2005.

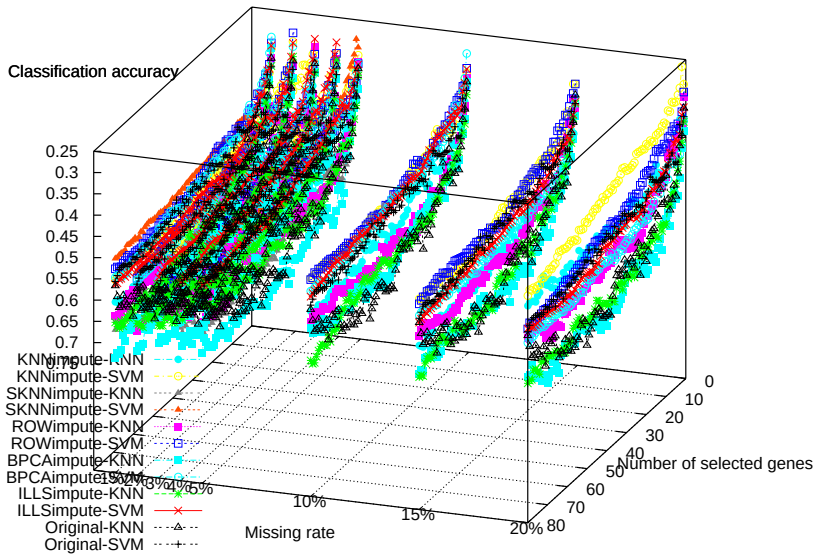

Figure 1.12. The 5-fold classification accuracies of the SVM-classifier and the KNN-classifier built on the genes selected by the CGS-Ttest method, on the original and simulated GLIOMA dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the 5-fold classification accuracy. The simulated datasets with missing values were imputed by each of the ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute. The Original-SVM/KNN plot the classification accuracies of the classifiers on the original GLIOMA dataset, i.e. $r = 0\%$. Note that we only applied SKNNimpute on missing rate less than or equal to 5%.

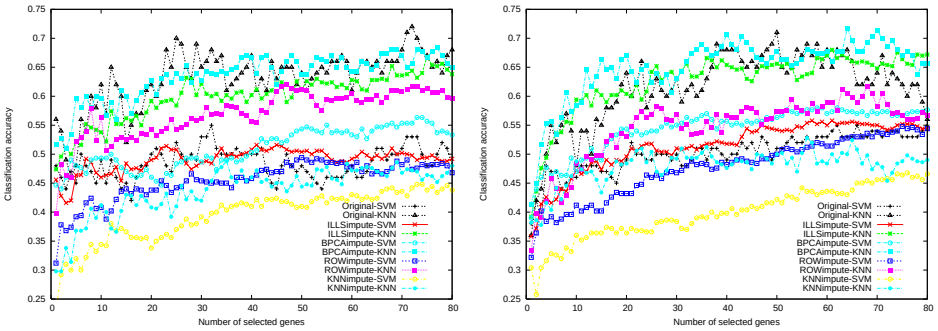

Figure 1.13. F-test (left) and T-test (right) performance on the GLIOMA dataset simulated with missing rate $r = 20\%$.

9. J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7:673–679, 2001.

10. H. Kim, G. H. Golub, and H. Park. Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics*, 21:187–198, 2005.

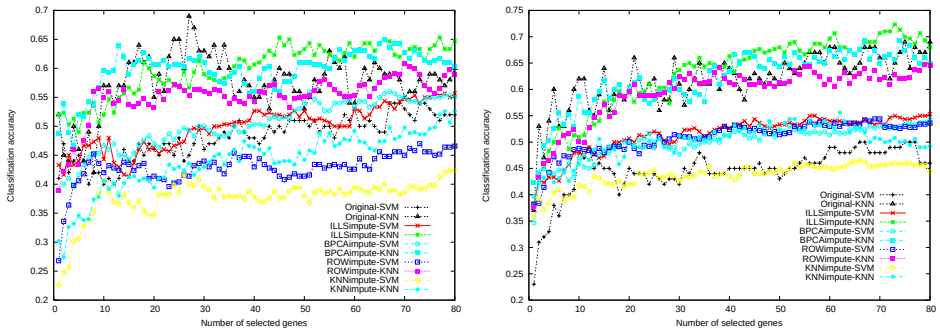

Figure 1.14. CGS-Ftest (left) and CGS-Ttest (right) performance on the GLIOMA dataset simulated with missing rate $r = 20\%$.

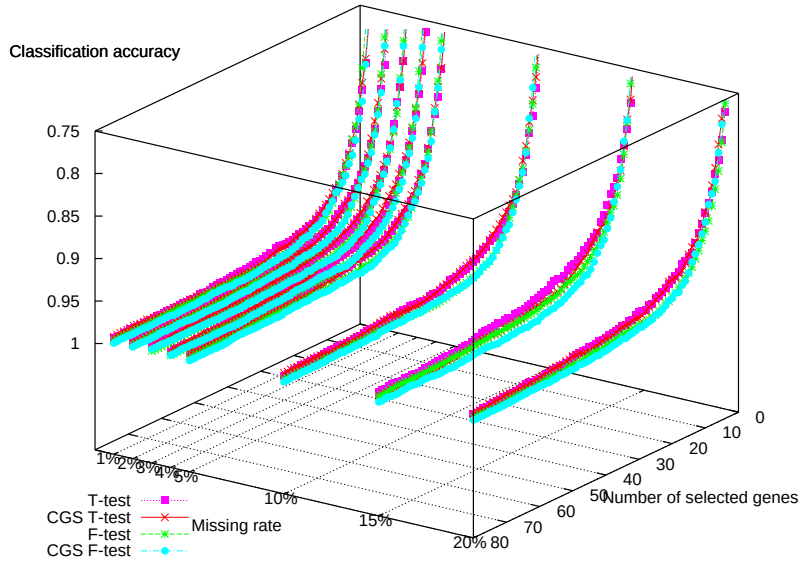

Figure 1.15. The 5-fold classification accuracies of four gene selection methods, F-test, T-test, CGS-Ftest, and CGS-Ttest, averaged over 8 combinations of a missing value imputation method and a classifier, on the SRBCT dataset. The x-axis labels the number of selected genes, the y-axis labels the missing rate, and the z-axis labels the average classification accuracy.

11. K.-Y. Kim, B.-J. Kim, and G.-S. Yi. Reuse of imputed data in microarray analysis increases imputation efficiency. *BMC Bioinformatics*, 5:160, 2004.

12. C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, J. G. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. E. McLaughlin, T. T. Batchelor, P. M. Black, A. V. Deimling, S. L. Pomeroy, T. R. Golub, and D. N. Louis. Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. *Cancer Research*, 63:1602–1607, 2003.

13. S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics*, 19:2088–2096, 2003.

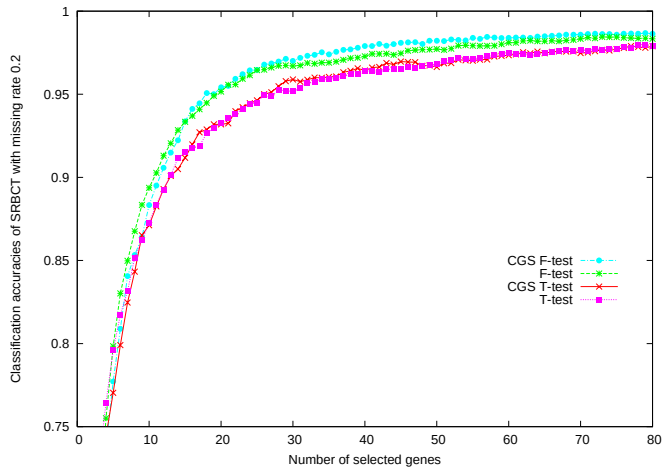

Figure 1.16. The 5-fold classification accuracies of four gene selection methods, F-test, T-test, CGS-Ftest, and CGS-Ttest, averaged over 8 combinations of a missing value imputation method and a classifier, on the simulated SRBCT dataset with missing rate $r = 20\%$.

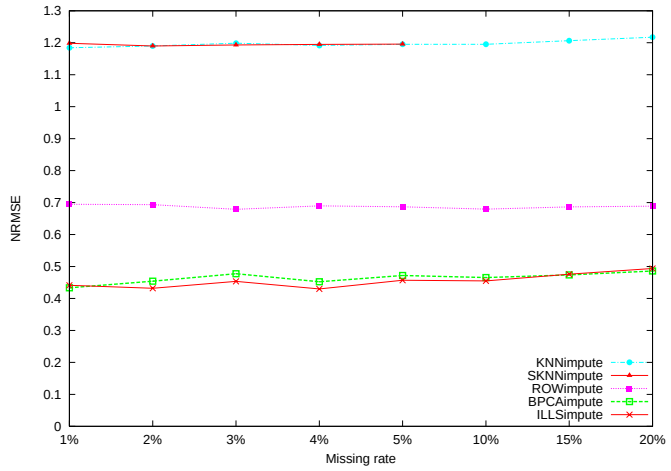

Figure 1.17. The NRMSE values of the five missing value imputation methods, ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, and ILLSimpute, with respect to the missing rate. SKNNimpute did not apply to the simulated SRBCT datasets with missing rates greater than 5%.

14. M. Ouyang, W. J. Welsh, and P. Georgopoulos. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, 20:917–923, 2004.

15. I. Scheel, M. Aldrin, I. K. Glad, R. Sørum, H. Lyng, and A. Frigessi. The influence of missing value imputation on detection of differentially expressed genes from microarray data. *Bioinformatics*, 21:4272–4279, 2005.

16. M. S. B. Sehgal, L. Gondal, and L. S. Dooley. Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data. *Bioinformatics*, 21:2417–2423, 2005.

17. M. Suárez-Fariñas, A Haider, and K. M. Wittkowski. "Harshlighting" small blemishes on microarrays. *BMC Bioinformatics*, 6, 2005.

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

19. M. Xiong, X. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Research*, 11:1878–1887, 2001.

20. K. Yang, Z. Cai, J. Li, and G.-H. Lin. A stable gene selection in microarray data analysis. *BMC Bioinformatics*, 7:228, 2006.