University of Alberta

Library Release Form

Name of Author: Yi Shi

Title of Thesis: Gene Expression Microarray Missing Value Imputation and Its Effects in Downstream Data Analyses

Degree: Master of Science

Year this Degree Granted: 2007

Yi Shi
2-21 Athabasca Hall, University of Alberta
Edmonton, Alberta
Canada, T6G 2E8

Date: _____

University of Alberta

GENE EXPRESSION MICROARRAY MISSING VALUE IMPUTATION AND ITS EFFECTS IN DOWNSTREAM DATA ANALYSES

by

Yi Shi

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Fall 2007

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Gene Expression Microarray Missing Value Imputation and Its Effects in Downstream Data Analyses** submitted by Yi Shi in partial fulfillment of the requirements for the degree of **Master of Science**.

_____

Dr. Guohui Lin (Supervisor)

_____

Dr. Russ Greiner (Examiner)

_____

Dr. Paul Stothard (External Examiner)

Date: _____

Abstract

DNA microarray is a high throughput gene profiling technology that has been employed in numerous biological and medical studies. These studies require complete and accurate gene expression values which are not always available in practice due to the so-called microarray missing value problem. In this dissertation, most of the existing microarray missing value imputation methods are reviewed and discussed. In these missing value imputation methods, the (normalized) root mean squared error is commonly adopted as a standard measurement of the imputation quality. However, considering that the imputed expression values are for downstream data analyses, we propose to use the microarray sample classification accuracy in addition to (normalized) root mean squared error, to measure the missing value imputation quality. Our extensive comparative study between seven missing value imputation methods circulate our conjecture that the sample classification accuracy is a more appropriate way for measuring the microarray missing value imputation quality.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Microarray Technology

DNA microarrays, also known as gene chips, DNA chips, or gene arrays, are a collection of microscopic DNA spots (probes) attached to a solid surface (e.g. glass, plastic or silicon chip). They are used for the purpose of gene expression profiling, to monitor the expression levels of thousands of genes simultaneously. Expression profiling is a microarray technology that detects the RNAs that may or may not be translated into active proteins. Currently, two types of microarrays are widely used in biological and medical experiments. One is the two-channel microarray, such as cDNA microarray, and the other is single-channel microarray, such as GE Healthcare, Affymetrix, or Agilent. The two-channel microarray is typically hybridized with cDNAs from two samples to be compared (e.g. disease and normal) that are labeled with two different fluorescent dyes (usually red and green). The samples can be mixed and hybridized to a microarray and the up-regulated and down-regulated genes are then scanned, visualized and quantified. In single-channel microarrays, the probes are designed to match parts of the mRNA sequences, and estimate the absolute gene expression values and therefore two separate microarrays are required to make the comparison of two different conditions.

Exploring the genes which are differentially expressed under different biological conditions by profiling their expression values is an important application of microarray experiments. Figure 1.1 illustrates how such an experiment is performed using a two-channel cDNA microarray. In general, there are six steps for such a microarray gene profiling experiment. (1) Cells are extracted from different samples (for example, disease sample and normal sample) and cultivated in different tubes for a period of time so that adequate amount of different kinds of cells could be collected. (2) Messenger RNAs (mRNA) are isolated in different conditions and extracted from them. (3) mRNAs from different conditions are reverse transcribed into their corresponding cDNAs of different fluorescent dyes (cDNAs are artificially synthesized DNAs from mRNA templates). (4) Different cDNAs are mixed up together and a proportion of them are hybridized to a single microarray chip. (5) The hybridized microarray chip is scanned by scanners of different color channels (red and green) and their fluorescence intensities are collected and quantified by specific sensor and software. (6) The

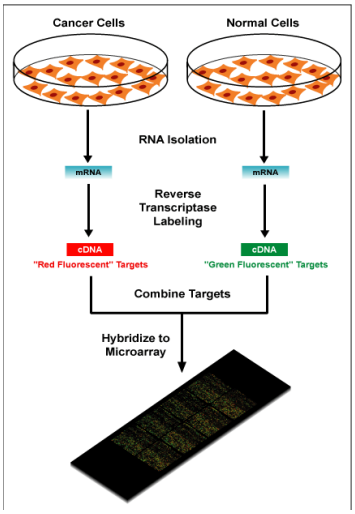


Figure 1.1: Microarray experiment procedure (figure from `wikipedia.org`).

logarithmic ratios of the intensities from different channels are computed and considered to be the gene expression levels.

1.2 Microarray Missing Value Problem

The DNA microarray technology provides an innovative tool for approaching the system-level understanding of biological systems, and has become indispensable in numerous studies covering a broad range of biological and medical disciplines. Most DNA microarray data analysis methods such as hierarchical gene clustering, biomarker identification, sample class prediction, and genetic and regulatory network prediction require the expression values to be complete and as accurately as possible. This requirement in practical experiments, however, is not always satisfied due to various experimental factors which originate from the imperfections at the level of chip production and treatment, and hybridization and scanning. Most of these imperfections occur at random. These include hybridization failures, artifacts on the microarray, insufficient resolution, dust on the slide, image noise, image corruption, or scratches on the slide. In addition to such expression values missing at random, systematic data missing might also occur. Note that during signal scanning, sensor and software are used to collect and compute the gene expression values, and software could flag signals which cannot be distinguished from the background or have a too irregular form because the signal itself is too low. In these cases, missingness depends on the signal intensity and therefore missing values are not at random [21].

Typically, 1-10% of the data on microarray can be missing, affecting up to 95% of the genes [21]. Even with the high-density oligonucleotide arrays such as Affymetrix GeneChip oligonucleotide arrays, as high as 20% percentage of expression spots on the arrays can be blemished and thus become

missing values. The missing value problem can affect the subsequent microarray data analyses to a serious extent. Although repetition of identical experiments [24] has been proposed and adopted to validate the microarray data analysis methods dealing with the missing value problem [25], to do so is not only costly and time consuming, but also cannot come to identical gene expression profiling results. Therefore, there is a strong motivation to design reliable and robust missing value imputation algorithms to estimate the missing values as accurate as possible. In the past several years, more than a dozen microarray missing value imputation methods have been proposed. As far as we know, the following missing value imputation methods are either commonly used or most representative or most recently proposed. They are (the nomenclature of the imputation methods in this paper is the acronym of the method + *impute*): Zero Imputation (ZEROimpute) [2, 26], Row Average Imputation (ROWimpute) [2, 26], Column Average Imputation (COLimpute) [2, 26], K-Nearest Neighbors Imputation (KNNimpute) [24], Singular Value Decomposition Imputation (SVDimpute) [24], Bayesian Principal Component Analysis Imputation (BPCAimpute) [19], Sequential K-Nearest Neighbors Imputation (SKNNimpute) [15], Gaussian Mixture Clustering Imputation (GMCimpute) [20], Least Squares Imputation (LSimpute) [5], Local Least Squares Imputation (LLSimpute) [14], Collateral Missing Value Imputation (CMVEimpute) [22], LinImp [21], Support Vector Regression Imputation (SVRimpute) [25], Iterated Local Least Squares Imputation (ILLSimpute) [7], Projection onto Convex Sets Imputation (POCSimpute) [11], and some combinations of them such as LinCmb [13]. In the Related Work chapter, these imputation methods will be reviewed in detail.

1.3 Motivation of Our Approach

Intuitively, the missing value imputation quality can be measured by how close the predicted values and the original expression values (the readout expression values) are. The closer they are, the better imputation quality the method achieves. Based on this principle, the *Root Mean Square Error* (RMSE) was proposed as a standard for measuring the imputation quality, and more recently, its *Normalized* version, *Normalized Root Mean Square Error* (NRMSE) was more commonly adopted [24, 19, 5, 15, 20, 13, 14, 21, 22, 25, 7, 11]. The NRMSE measurement presumes that all the observed gene expression values, which are not considered as missing values, should accurately measure the hybridization intensities of the genes or probes on the microarray chips. This presumption, however, is not necessarily the case. As we mentioned earlier, even on the high-density oligonucleotide arrays such as Affymetrix GeneChip oligonucleotide arrays, a considerable percentage of probes could have been blemished, so many should be treated as missing values. Moreover, the boundary between the accepted expression values and the treated-as-missing values is often vague, which means that among the accepted expression values, there still could be a considerable percentage of them that do not accurately measure the true gene hybridization intensities, although data noises in them may not be significant enough for them to be treated as missing values.

Based on these observations, in addition to NRMSE, some other measurements have been pro-

posed to measure the missing value imputation quality [20, 21]. These measurements are often ineffective or hard to apply to most public microarray datasets. Considering one of the most important applications of gene expression microarray is for discriminating different experimental conditions, for example, disease subtype recognition and disease treatment classification, we propose to adopt one downstream microarray data analysis, microarray sample classification, to be a measurement of the quality of missing value imputation, in addition to NRMSE. The main strength of this new measurement is that it resolves the issues caused by using NRMSE as described above, because the imputed expression values themselves are not interesting, while whether or not the imputed expression values can be trusted and used in downstream applications is the major concern.

The rest of this dissertation is organized as follows: In Chapter 2, the important microarray missing value imputation methods proposed in recent years are reviewed in detail, including ILLSimpute developed within our group, followed by some well-known missing value imputation measurements. The gene selection (biomarker identification) algorithms which are proposed for improving the sample classification accuracies are then introduced. In Chapter 3, the complete work flow of the new missing value imputation measurement, from missing value imputation methods, to gene selection, to classifier building, is introduced in detail. Chapter 4 presents the experimental results of the new measurement and Chapter 5 discusses and concludes this dissertation with some proposed future works of our study.

Chapter 2

Related Work

2.1 Missing Value Imputation Methods

2.1.1 ZEROimpute, ROWimpute and COLimpute

Before data analysis, microarray data are usually arranged in the following format. The microarray expression values of genes collected from different experiments form an expression matrix. In this expression matrix, each column contains the expression values of a single microarray experiment and each row contains the expression values of a single gene over all these experiments. In the two-color microarray, since expression values are usually pre-normalized so that they distribute in range $[-R, R]$, for some integer R [2, 24], ZEROimpute fills the missing values with zero. Although it is very simple and efficient, obviously, ZEROimpute could artificially create erroneous relationships between genes since the integrity and usefulness of the non-missing data on the expression matrix are not taken into account. As an improvement to ZEROimpute, mean imputation such as ROWimpute fills a missing value with the mean expression value of its corresponding row (excluding those spots that are missing values themselves). Similar to ROWimpute, COLimpute fills a missing value with the mean expression value of its corresponding column (excluding those spots which contain missing values). Both ROWimpute and COLimpute can be generalized for single-channel arrays such as Affymatrix arrays. Having the similar drawback as ZEROimpute, mean imputation methods do not make good use of information between genes, although the methods themselves are very efficient and simple to apply.

2.1.2 KNNimpute, SVDimpute, and SKNNimpute

With the advance of microarray technology and its increasing applications, the missing value problem began to attract more attention and much more complex missing value imputation methods have been proposed, differing from pivotal ideas. Troyanskaya *et al.* [24] proposed the Singular Value Decomposition (SVDimpute) and the weighted K-Nearest Neighbors (KNNimpute) missing value imputation methods. In SVDimpute, a set of mutually orthogonal expression patterns are obtained and their linear combination is used to approximate the expression of all genes, through the singular

value decomposition of the expression matrix. In more detail, after selecting the K most significant eigengenes, a missing value in the target gene is estimated as the following: First, the target gene is regressed against these K eigengenes. Then, using the regression, the missing value is estimated from a linear combination of the K eigengenes [24, 7]. In KNNimpute, for a target gene, its K nearest neighbor genes (rows) which do not contain missing values in the same column (sample) positions as the target gene, are selected, according to their Euclidean distances to the target gene. Then, the missing value in the target gene is estimated by a weighted linear combination of the K nearest neighbor genes, where a weight is calculated as the inverse of the Euclidean distance between the target gene and the neighbor gene. In the literature, there exist several variants of the KNNimpute algorithm. In some articles such as [16, 20], the neighbors are not allowed to have any missing values. Our experimental results show that this can cause problems in a dataset where a high percentage of missing values exist, because only a few, or even no gene, actually meet this requirement of having no missing value other than the position that the target gene has. The imputation could fail consequently. In some other articles such as [19], missing values in the neighbors are allowed, but they are not filled with certain initial estimates such as row averages before being used to estimate the missing values in the target gene. This will lead to false distances for neighbors which have a lot of missing values. All these variants of KNNimpute are less effective than the full version KNNimpute, which does not likely suffer from any of the weaknesses mentioned above. We choose to use the full version of KNNimpute in our comparison experiment. As an extension to the KNNimpute method, the Sequential K-Nearest Neighbors imputation (SKNNimpute) method imputes missing values from genes with the least number of missing values to genes with the most number of missing values sequentially. Within each iteration of SKNNimpute, KNNimpute is applied to estimate the missing values in the target gene but only the genes that have no missing value or whose missing values have already been imputed can be used as neighbor genes. Note that the KNNimpute process employed in the SKNNimpute is different from the full KNNimpute version, which initially fills the missing values in neighbor genes with row averages before the regression process, and this is because instead of doing so, SKNNimpute uses the imputed genes as candidate genes when estimating missing values for a target gene.

2.1.3 LSimpute, LLSimpute, and ILLSimpute

Bø *et al.* [5] proposed a least square principle based imputation method named LSimpute, which is similar to KNNimpute to some extent. Considering that not only genes but also samples in the microarray expression matrix could be correlated, LSimpute proposes and combines the LSimpute_gene and LSimpute_sample methods which are gene-based LSimpute and sample-based LSimpute, respectively, to estimate the missing values. In LSimpute_gene, for a target gene, the k genes that are the most correlated to it are selected according to the their absolute correlation values, and when selecting the correlated genes, only the columns (samples) where both genes (the target gene

and candidate gene) having non-missing values are included. Because multiple linear square regression on gene correlations is not feasible for more than a few genes, after the k correlated genes are selected, for each pair of the target gene and the candidate gene, the single regression for estimating the missing values in the target gene is computed based on the least square regression. Then, by using a weighted combination of these single regression estimates, the weighted estimate is computed, and the way to determine the weights is based on the absolute correlation values. The higher the absolute correlation value between a candidate gene and the target gene, the greater the weight assigned to their corresponding single regression estimate. Similar to LSimpute_gene, LSimpute_sample estimates the missing values in a target sample by using the most correlated samples. Because there are usually fewer samples than genes in a microarray expression matrix, after the k correlated samples are selected, a multiple regression estimate using these k samples, which is also based on least square principle, can to be applied to estimate the missing values. Then, the missing values are finally estimated by the weighted averages of the estimates from LSimpute_gene and LSimpute_sample, where the weights are determined by choosing one which minimizes the sum of squared errors between the artificially removed and later imputed missing values on the expression matrix and their corresponding original true values. Proposed by Kim *et al.* [14], local least square imputation, referred to as LLSimpute, is another missing value imputation method based on the least square principle. For a target gene, LLSimpute first determines the k genes most correlated to it by using the L_2-norm or Pearson correlation coefficient. Then, a linear combination of the k coherent genes is used to estimate a missing value in a target gene, and the linear combination is determined based on least square principle. Most recently, within our group, Cai *et al.* [7] proposed iterated local least square imputation, referred to as ILLSimpute, which extends LLSimpute by employing an iterated procedure in LLSimpute and by learning the parameter k for selecting the k coherent genes. Since tending to exploit the local correlation in dataset, LSimpute, LLSimpute, and ILLSimpute usually have better performance on datasets where strong local relations between genes or samples exist.

2.1.4 BPCAimpute

Instead of exploiting the local correlations in the microarray expression dataset for missing value estimation, Bayesian principle component analysis imputation, referred to as BPCAimpute, which was proposed by Oba *et al.* [19], took into consideration the global correlation in the expression dataset. Essentially, BPCAimpute employs three elementary processes, a principle component regression, a Bayesian estimation, and an expectation-maximization-like repetitive algorithm. BPCAimpute imputes the missing values using a probabilistic model under the framework of Bayesian inference, by estimating the latent parameters in the model. In BPCAimpute, missing values are estimated using a Bayesian estimation algorithm, which is used for both Θ, the model parameter, and Y^{miss}, the

missing values. Finally, the missing values $\hat{Y}$ in the expression matrix are estimated using:

$$\hat{Y} = \int Y^{miss} q(Y^{miss}) dY^{miss},$$

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

where $q(Y^{miss})$ is the posterior distribution for Y^{miss} and Θ_{true} is the posterior parameter of the missing value.

2.1.5 LinImp

Scheel *et al.* [21] proposed LinImp, which acts to individual channel missing values separately. By using a linear model, LinImp estimates a single channel missing value y_{ijkg}, which is the base 2 logarithm of the intensity in array i, channel (dye) j, variety k and gene g as:

$$y_{ijkg} = \mu + A_i + D_j + G_g + AD_{ij} + AG_{ig} + DG_{jg} + VG_{kg} + \varepsilon_{ijkg},$$

where ε_{ijkg} are the independent errors normally distributed with mean 0 and variance σ^2. The varieties are the experimental conditions under study. μ is the overall mean expression value, A_i is the effect factor caused by array i, D_j is the effect of dye j, G_g is the overall effect of gene g, AD_{ij} is the interaction between array i and dye j, AG_{ig} is the interaction between array i and gene g, DG_{ig} is the interaction between dye j and gene g, and VG_{kg} is the interaction between variety k and gene g [21]. The model can be re-written in matrix form:

$$y = X\beta + \varepsilon,$$

where y is a vector of gene expression values, X is a matrix of zeros and ones, and

$$\beta = (\mu, A^T, D^T, G^T, AD^T, AG^T, DG^T, VG^T)^T.$$

When imputing missing values, LinImp first imputes missing values using existing imputation methods, such as KNNimpute, to get the initial estimation vector Y^0. Using Y^0, based on the linear model, the parameter vector β is estimated, denoted as $\hat{\beta}^0$. Then, using $\hat{\beta}^0$, the new full data matrix Y^1 is obtained. LinImp repeats this iteration procedure until $\|Y^M - Y^{M-1}\| < \delta$, where M is the number of iterations and δ is a fixed small value. Disregarding the effectiveness of LinImp imputation, because it requires individual channel information, which is not always available in many public datasets, its applicability is limited. However, an extended version of LinImp which is applied the to logarithm ratio of two channels could be more applicable.

2.1.6 GMCimpute

Ouyang *et al.* [20] proposed a Gaussian mixture clustering based missing value imputation algorithm, referred to as GMCimpute. In GMCimpute, data are modeled by Gaussian mixtures and missing values are estimated by the Expectation-Maximization (EM) algorithm. Given an empirically

determined value S, for the expression matrix A with missing values, Gaussian mixture clustering estimations $A_1, A_2, A_3, \ldots, A_S$ with cluster numbers $K = 1, 2, 3, \ldots, S$ are computed and the final missing value estimation is calculated as the average of them. Given K and A, a K_estimate procedure is used to estimate A_K by first extracting complete rows (or genes) from A to construct a matrix B which does not have missing values. Gaussian mixture clustering is then applied on matrix B and the Gaussian mixture clustering parameters are computed accordingly. Then, given matrix A and these computed parameters, the EM algorithm, denoted as EM_estimate, is used to compute the first version of the estimated expression matrix A'. After the first version of matrix A' is available, a new set of Gaussian mixture clustering parameters are generated based on it and the EM_estimate procedure is again applied on A' using the new generated parameters to estimate the new version of A'. These two steps of generating the new parameters and the EM_estimate procedure are iteratively executed until the whole process converges. Disregarding the effectiveness, there are two limitations in GMCimpute. First, when the missing ratio in the expression matrix is high enough, which actually happens in practice, the size of complete gene matrix B could be very small or even be zero, which would affect the subsequent procedure of GMCimpute or even make GMCimpute inapplicable. Although other imputation methods could be used to initially fill the missing values, biases could be introduced into the GMCimpute [20]. Second, in GMCimpute, the EM_estimate and K_estimate are not guaranteed to converge [20], which would fail the whole imputation.

2.1.7 CMVEimpute

The collateral missing value imputation (CMVEimpute), proposed by Sehgal *et al.* [22], considers both positive and negative relations between genes when estimating missing values. In CMVEimpute, three estimates (Φ_1, Φ_2, and Φ_3) are generated and the final estimate is distilled from them. When first selecting the K most correlated genes for the target gene, the covariance function, rather than the Euclidean distance used in KNNimpute, is employed to measure the similarities between a candidate gene and the target gene. Then a least square regression method is applied to estimate Φ_1. When estimating Φ_2 and Φ_3, CMVEimpute uses the non-negative least square (NNLS) algorithm, which is superior for estimating positive correlated values. The final missing value estimate χ is formed using

$$\chi = \rho \cdot \Phi_1 + \delta \cdot \Phi_2 + \lambda \Phi_3,$$

where $\rho = \delta = \lambda = 0.33$ are used to avoid bias toward one particular estimate [22]. In practice, however, since each of these three estimates could be highly data-dependent, adjusting the three weight parameters could improve or repress the imputation quality.

2.1.8 SVRimpute

Wang *et al.* [25] proposed a support vector regression based missing value imputation, namely SVRimpute. For a missing position j in the target gene, all rows (or genes) in the expression matrix

with non-missing values in the j-th position are used to form a training set, which is then mapped into a higher dimensional space to construct a model for regression, and all rows with missing values in the j-th position are used to compose a testing set [25]. By identifying support vectors in the higher dimensional space, SVRimpute builds the regression and predicts the missing values for the testing set. Note that the support vector regression can only predict one missing value in one row, while in practice, multiple missing values could exist in a single row. To solve this problem, SVRimpute uses a coding scheme to pre-process the data. For a row containing more than one missing value, SVRimpute first fills in each missing value with zero or the row average or the column average except the one which is to be predicted. Disregarding the innovative ideas used in SVRimpute, according to the experiment results [25], SVRimpute does not consistently outperform other existing missing value imputation methods such as BPCAimpute and LLSimpute. Moreover, the coding scheme for multiple missing values problem in the pre-process is questionable since ZEROimpute, ROWimpute, and COLimpute have been proven less effective than other missing value imputation methods. A more convincing coding scheme may improve the performance of SVRimpute and an iterative process which continually updates the missing values could be considered to be adopted in SVRimpute.

2.1.9 POCSimpute

Most recently, Gan *et al.* [11] proposed a missing value imputation method using a set theoretic framework based on projection onto convex sets (POCS) for the prior knowledge of the expression data. POCSimpute captures three pieces of information during the imputation: the gene-wise correlation, the array-wise correlation, and the phenomenon of synchronization loss. First, when capturing gene-wise correlation, similar to LSimpute, POCSimpute first selects the K most correlated genes in matrix for a target gene. Then it estimates the missing values in the target gene using each of these K genes based on the single regression model. By using a weight function, the weighted average of these single regressions is computed and is considered as the gene-wise missing estimates. Based on the estimation, a convex set is obtained through a projection procedure. Second, when finding the array-wise correlation, POCSimpute uses the principle component analysis approach to capture the global array-wise variation. Then, another convex set is obtained based on the estimation. Third, when capturing the phenomenon of synchronization loss, the missing values in different time periods are projected onto a convex set. Finally, according to these three convex sets, an iterative process estimates a missing value from an initial estimation and approaches the final solution iteration by iteration, using an equal-weight function which combines the three projectors until the process converges.

According to the NRMSE values, POCSimpute shows better imputation quality than previous imputation methods on their testing datasets. Note that the third POCS in POCSimpute is only applicable on the time series datasets.

2.2 Imputation Measurements

In most of the missing value imputation methods introduced above, the *Root Mean Square Error* (RMSE) is employed as a standard criterion for measuring the imputation quality, and more recently, its *normalized* version, *Normalized Root Mean Square Error* (NRMSE) has been more commonly used [24, 19, 5, 15, 20, 13, 14, 21, 22, 25, 7, 11]. NRMSE analysis works as follows [7]: Let $E = \{E_1, E_2, E_3, \ldots, E_t\}$ denote the missing entries in the microarray expression matrix. For each missing value entry E_i ($i = 1, 2, 3, \ldots, t$), let e_i^* and e_i correspond to original observed expression value and the imputed expression value, respectively. The mean of the squared errors is calculated as

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

The mean of these t original expression values is calculated as

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

and

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

stands for the standard deviation of the original expression values. The NRMSE is then calculated as the ratio of μ over σ, i.e., NRMSE = $\frac{\mu}{\sigma}$. According to the definition of NRMSE, the smaller an NRMSE value is, the better the corresponding imputation quality is.

In addition to NRMSE, Ouyang *et al.* [20] suggested that, with known gene cluster information, the percentage of mis-clustered genes could be used as a measurement of imputation quality. However, in most of the proposed missing value imputation methods, either implicitly or explicitly, the missing values in the target gene are estimated using the genes with similar expression patterns, the neighbors or the coherent genes. Therefore, using gene cluster information in final imputation quality measurement may not tell much more than RMSE or NRMSE. Moreover, to the best of our knowledge, better clustering algorithms are still needed to divide genes into their actual clustering patterns, and the quality of clustering would affect the measurement to some extent.

Scheel *et al.* [21] investigated the influence of imputation on the detection of differentially expressed genes from cDNA microarray data and proposed to use the lost number of differentially expressed genes as a new measurement of microarray missing value imputation quality. However, this measurement uses the expression data which has expression values of each channel (red and green), not their log ratios, and these data are rarely published. Therefore, this measurement cannot be widely applied on most published microarray expression datasets.

2.3 Gene Selection and Classification

For the purpose of microarray sample classification, an expression matrix is provided with a label vector where each label indicates the class membership of its corresponding sample. In diagnosis, for example, such sample labelling can be added done as long as the kind of disease (or its subtype) that the sample (patient) has is apparent. Such a dataset that can be used to learn the expression profiles associated with each class is considered as the training dataset, and subsequently, whenever a new sample comes, its class membership can be predicted using a classifier built based on the training dataset. All the genes in the training dataset can be used to compose the expression profiles. However, before using the training dataset, a process called *gene selection* (or *biomarker identification*) is usually indispensably employed to select a subset of genes for building a better classifier because of two reasons: First, not all the genes contribute to the improvement of sample classification accuracy. Instead, only a few of them are the most discriminatory genes that either over-express or under-express under different conditions (or classes), and these genes are the target genes that should be selected in the gene selection process. Second, by selecting a subset of genes, the computational workload can be reduced so the genetic profiles can be built more efficiently.

Many gene selection methods have been proposed in the past several years [3, 17, 26, 9, 6]. Among these methods, we adopt four of them in this work, F-test, Cho, CGS-Ftest, and CGS-Cho, which have been successful in practice.

The first gene selection method used in our work is the F-test gene selection method [3, 4], which tries to capture the genes that have the largest score among all the genes according to a scoring function, which is basically a ratio of the inter-class variance over intra-class variance:

$$\text{score } = \frac{\sum_{i=1}^{c}(\bar{M}_i - \bar{M})}{\sum_{i=1}^{c}\sum_{j=1}^{n_j}(v_{ij} - \bar{M}_i)},$$

where c is the number of classes in the dataset, $\bar{M}_i$ represents the mean expression value of the gene in class i, $\bar{M}$ is the mean expression of all the expression values of the gene over all the samples, n_j is the number of samples in class j, and v_{ij} is the expression value of the gene in sample j in class i. According to this scoring function, the F-test method sorts all the genes in the order of decreasing score order and returns a specified number of top ranked genes.

The second gene selection method used in our work is the Cho gene selection method proposed by Cho *et al* [9]. For sample i, Cho defines a weighted factor w_i, which is $1/n_k$ if sample i belongs to class k. Let $W = \sum_{i=1}^{n} w_i$, where n is the total sample number. The weighted mean expression value M_j for gene j is defined as:

$$M_j = \sum_{i=1}^{n} \frac{w_i}{W} x_{ij}.$$

The weighted standard deviation is defined as:

$$SD_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij-M_j})^2}{(n-1/n)W}}.$$

where x_{ij} is the expression value of sample$i$, gene j.

The score of gene j is calculated as:

$$\text{score}_j = \frac{M_j \times SD_j}{\bar{SD}},$$

where

$$\bar{SD} = \sqrt{\frac{1}{c}\sum_{p=1}^{c}(E_p - \bar{E})}.$$

Here, c is the number of classes in the dataset, E_p is the centroid expression value of the genes in class p, and $\bar{E}$ is the mean expression value of these centroid expression values. That is, $\bar{SD}$ is actually the standard deviation of the centroid expression values of each class. Similar to F-test, the Cho method also sorts all the genes in the non-increasing score order and returns a specified number of top ranked genes.

Z. Cai *et al.* [8] recently proposed to apply a Clustered Gene Selection (CGS) method. CGS assumes that all the genes in the microarray expression matrix belong to a certain number of clusters, and the class discrimination strength of genes within the same cluster are similar to each other. Therefore, when many genes belonging to the same cluster are selected by some gene selection methods such as F-test and Cho, using all of them could be redundant and keep other significant genes from being selected due to the pre-specified total number of genes to be selected. CGS first uses a clustering process for all the genes and then combines with other gene selection methods to select the genes with the most discrimination strength from each of these clusters. In our work, we combine the CGS approach with the F-test and Cho gene selection methods to create CGS-Ftest and CGS-Cho, respectively.

Chapter 3

Methods

3.1 The Classifiers

In our study, two classifiers, the K-Nearest Neighbors (KNN)-classifier [10] and a linear kernel Support Vector Machine (SVM)-classifier [12] are adopted for sample class membership prediction. The KNN-classifier predicts the membership of a testing sample based on the expression values of a subset of genes that are selected using certain gene selection methods. The classifier first identifies the K closest samples in the training dataset and then uses the class labels of these K similar samples to predict the label of the testing sample through a majority vote. In our experiments, after testing K from 1 to 10, we set the default value of K to be 5, since in practice it leads to a high and stable classification accuracy. For the SVM-classifier, when given a set of selected genes from a gene selection method, the SVM-classifier, which contains multiple SVMs, finds decision hyperplanes to best separate (soft margin) the labeled samples based on the expression values of these selected genes. Subsequently, it uses this set of decision hyperplanes to predict the class label of a testing sample. For more details of how the decision hyperplanes are constructed, the interested readers may refer to Guyon *et al.* [12].

3.2 Cross Validation

Given a complete gene expression matrix with all samples being labeled with their class memberships, we employ the *l-fold cross validation* to avoid the possible data overfitting problem. For doing the l-fold cross validation, the complete dataset is randomly partitioned into l equal parts. Each part of the l equal parts is used as the *testing dataset* at one time by removing its sample labels, while the rest $(l-1)$ parts are used as the *training dataset*. This process is repeated for each of the l parts. Based on the classifier built on the training dataset, the sample labels of the testing dataset are predicted and compared with the original true sample labels. The percentage of the correctly predicted samples is the *classification accuracy* of the classifier. In the experiments, after testing $l = 3, 5, 7, 9, 11$, we report the results on the 5-fold cross validation, but similar results (data not shown) present when $l = 3, 7, 9, 11$. The random partition pro-

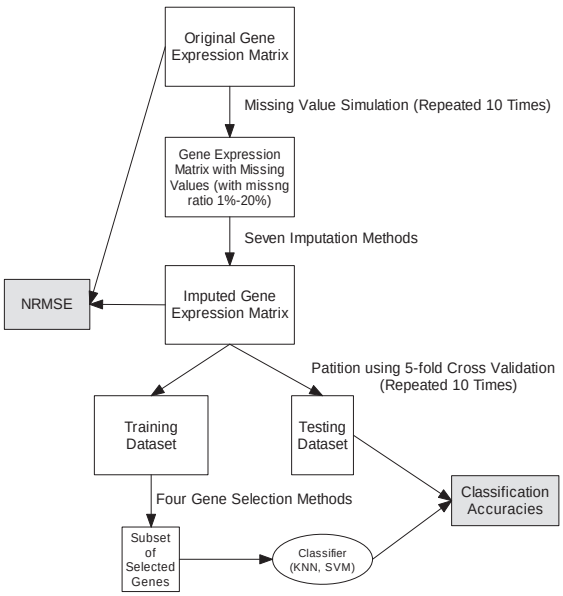


Figure 3.1: The complete work flow of computing the sample classification accuracies.

cess is repeated for 10 times. Consequently, the final classification accuracy is the average over $500(= 10simulations \times 10crossvalidation \times 5folds)$ testing datasets.

3.3 The Complete Work flow

In addition to presenting the NRMSE values for all the seven imputation methods on the respective datasets, we use the microarray sample classification accuracy to demonstrate that the sample classification accuracy is another very effective measurement for the missing value imputation quality. To start with, we first randomly simulate the missing values in the original complete gene expression matrix with certain missing ratios (or missing rates) r ($r = 1\% - 20\%$). In more details, if a complete expression matrix contains p genes, n samples and c classes, we randomly pick $p \times n \times r$ entries from it and erase them to form a dataset containing missing values. Note that the simulation of the missing values on the original expression matrix is based on the uniform distribution. Although the missingness on the original microarray chip may occur not completely at random (e.g., unbalanced distribution of the hybridization solution on the a chip can lead a group of probes within one area of the chip darker than the others which may cause a high percent of probe values missing in this area), due to the development of probe designing technology (Locating redundant probes in different areas of the chip to avoid simultaneous same probe loss. And when combining signals from the many probes for a gene into a single estimate of the abundance of that gene, normalization technologies are used.) and more redundant probes introduced in newly developed microarray chips (e.g., for

most Affymetrix GeneChip, 22 probes are routinely used for each expression measurement and 40 for each genotype call [1]), in our simulation, we assume that the missingness occurs at random as previous studies assumed [14, 5, 22, 11, 20, 24]. Moreover, what we are trying to focus on is proposal of using gene selection-driven sample classification accuracy to measure the missing value imputation quality, so as long as imputation are performed on dataset with same missingniss, the results for downstream analyses are comparable and in the same level. Then, on the simulated missing value expression matrix, a missing value imputation method is applied to estimate the missing values. Among the missing value imputation methods which are either commonly used, or most well known, or most recently published, in our study, we use seven missing value imputation methods, i.e. COLimpute, ROWimpute, KNNimpute, SKNNimpute, BPCAimpute, LLSimpute, and ILLSimpute, to estimate the missing values to obtain an imputed full expression matrix. On an imputed full expression matrix, the 5-fold cross validation method is applied to partition the whole dataset into two subsets, the training dataset and the testing dataset. On the training dataset, a gene selection (biomarker identification) method is applied to select a subset of genes with the selected gene number ranging from 1 to 80. The gene selection methods we employed in our study are F-test, Cho, CGS-Ftest and CGS-Cho. Based on the expression values of the subset of selected genes, two classifiers, i.e. the KNN classifier and the SVM classifier, are built to predict the class membership of each sample in the testing dataset. The classification accuracies are then collected. Note that for each missing ratio, the missing value simulation is repeated for 10 times. Therefore, along with the 5-fold cross validation, the associated classification accuracies are the average over 500 entities.

To summarize, by regarding the original complete dataset as a dataset of 0% missing values, we have 21 missing ratios, each associated with 10 simulated datasets (except 0%), seven missing value imputation methods (except 0%), four gene selection methods, and 2 classifiers, under the 5-fold cross validation scheme, which is repeated for 10 times. Figure 3.1 illustrates the complete work flow of computing the sample classification accuracies.

Chapter 4

Experimental Results

Given a complete microarray gene expression dataset (which also can be regarded as a dataset with missing ratio 0%), based on the uniform distribution, we randomly simulated 10 datasets for each of the missing ratios $r = 1\% - 20\%$. On each simulated dataset, all the seven missing value imputation methods, namely COLimpute, ROWimpute, KNNimpute, SKNNimpute, LLSimpute, ILLSimpute, and BPCAimpute, were run separately to estimate the missing values. Then, on either the original complete dataset or the imputed complete dataset, each of the four gene selection methods, namely, F-test, Cho, CGS-Ftest, and CGS-Cho was applied on the randomly picked $4/5$ samples to select x genes, for $x = 1, 2, ..., 80$. The KNN-classifier and the SVM-classifier were then built based on these x selected genes to predict the class memberships of the other $1/5$ samples. The final classification accuracies were collected for further statistics.

4.1 Dataset Description

We adopt two real cancer microarray gene expression datasets, the Gliomas dataset [18] and the Carcinomas dataset [23], in our study.

The Gliomas dataset [18] contains 50 samples, in four classes, the *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 low quality for sample classification [18, 27]. Considering that among all the genes, there could exist a certain percent of noisy genes, housekeeping genes for instance, which do not actually have too much discrimination strength and are less likely to be selected by any gene selection method, on the whole dataset, we calculate for each gene its expression standard deviation over all samples, and those genes with standard deviation lower than a threshold are filtered out. By doing this, we can also improve the efficiency of the gene selection and the sample classification. After this filtering preprocessing, we obtain a dataset with $4,434$ genes out of the original 12000 genes.

The Carcinomas dataset [23] is a relatively larger dataset compared with the Gliomas dataset. It contains 174 samples, which are in 11 classes, the *Ovary*, *Bladder/ureter*, *Breast*, *Colorectal*, *Gas-*

troesophagus, *Kidney*, *Liver*, *Prostate*, *Pancreas*, *Lung Adeno*, and *Lung Squamous*, with class sizes 27, 8, 26, 23, 12, 11, 7, 26, 6, 14, and 14, respectively. After the same filtering preprocessing as in Gliomas dataset, the Carcinomas dataset contains 1585 genes comparing to the gene number 12533 on the original dataset (Since in this work, we focus on whether the idea of using gene selection based sample classification to investigate the quality of missing imputations work, rather than examining how the dataset quality affect the classification accuracy, the dataset size is not too much concerned here. Also note that large dataset may extremely consume runtime with insignificant affection in the final classification accuracy). We found in our experiment that rather than depressing the sample classification quality, the filtering preprocessing can affect the sample classification quality in a positive way, i.e. it can improve the sample classification quality to some extend as long as proper filtering threshold is chosen.

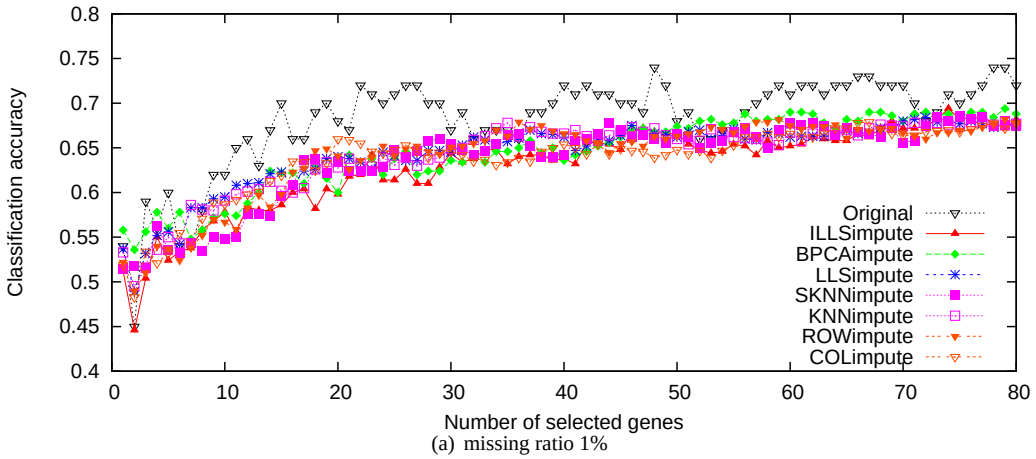
4.2 Classification Accuracies of the KNN-Classifier

Since we discovered in our experiment that the sample classification accuracies computed based on the SVM-classifier are not as high as the accuracies computed based on the KNN-classifier, we only present in this paper the results of the latter classifier, since the purpose of this study is to indicate that the sample classification accuracy is another effective measurement for the missing value imputation quality in addition to the NRMSE measurement, rather than searching for the best classifier. For each gene selection method, under the 5-fold cross validation scheme, its corresponding sample classification accuracy is the average over 500 testing datasets on each of the missing ratios $r = 1\% - 20\%$. To simplify our presentation, we concatenate the sequentially applied method names, i.e. the names of the missing value imputation method, the gene selection method, and the classifier, to denote the associated 5-fold cross validation classification accuracy. For example, ILLSimpute-CGS-Ftest-KNN denotes the accuracy that is achieved by applying the ILLSimpute missing value imputation method, followed by the CGS-Ftest to select a subset of genes for building a KNN-classifier for the class membership prediction of the testing data. Our further statistical analyses include the sample classification accuracies with respect to a missing value imputation method. For example, ILLSimpute-KNN denotes the average accuracy over all of the four gene selection methods, that is, ILLSimpute-Ftest-KNN, ILLSimpute-Cho-KNN, ILLSimpute-CGS-Ftest-KNN, and ILLSimpute-CGS-Cho-KNN.

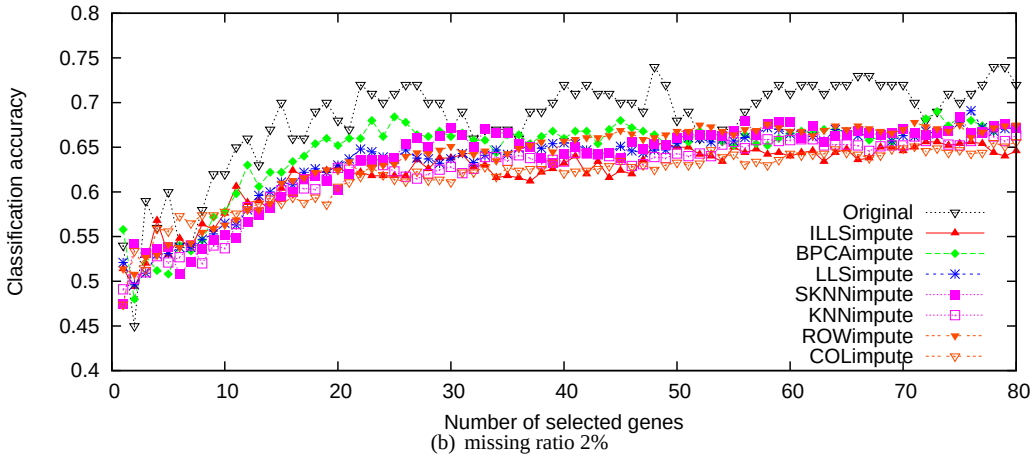
4.2.1 The *Gliomas* Dataset

For each of the four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, we plot a set of two dimensional figures of the sample classification accuracies computed on the original Gliomas dataset (missing ratio $r = 0\%$, in which the missing value imputation methods are skipped) and the simulated datasets with missing ratio $1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%,$ and 20%, respectively. In such a two dimensional plot, the x-axis is the number of selected genes and the y-axis is the 5-fold

cross validation classification accuracy. Figures 4.1–4.4 plot the classification accuracies for the F-test gene selection method on the original and the simulated expression datasets with missing ratio $1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%,$ and 20%, respectively. Figures 4.5 and 4.8 plot the classification accuracies for the Cho gene selection method on original and simulated expression datasets with missing ratio $1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%,$ and 20%, respectively. Figures 4.9 and 4.12 plot the classification accuracies for the CGS-Ftest gene selection method on original and simulated expression datasets with missing ratio $1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%,$ and 20%, respectively. Figures 4.13 and 4.16 plot the classification accuracies for the CGS-Cho gene selection method on original and simulated expression datasets with missing ratio $1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%,$ and 20%, respectively.

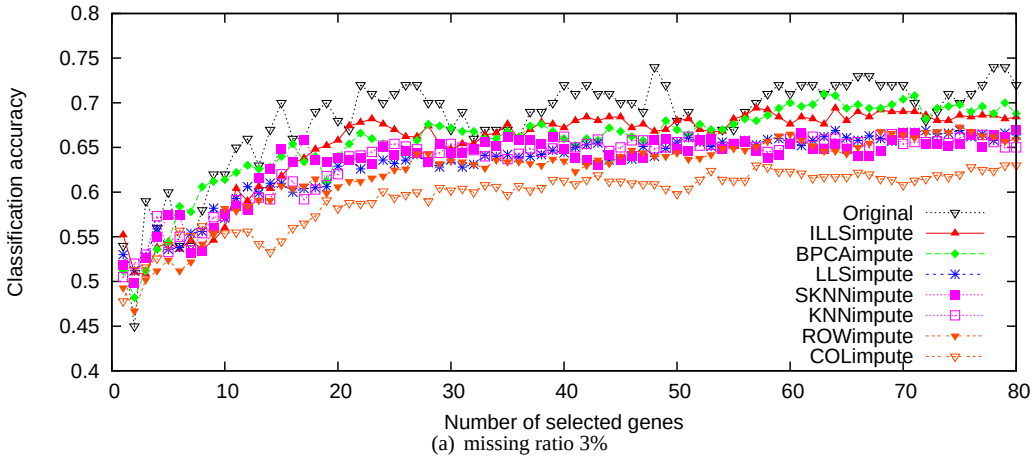


(a) missing ratio 1%

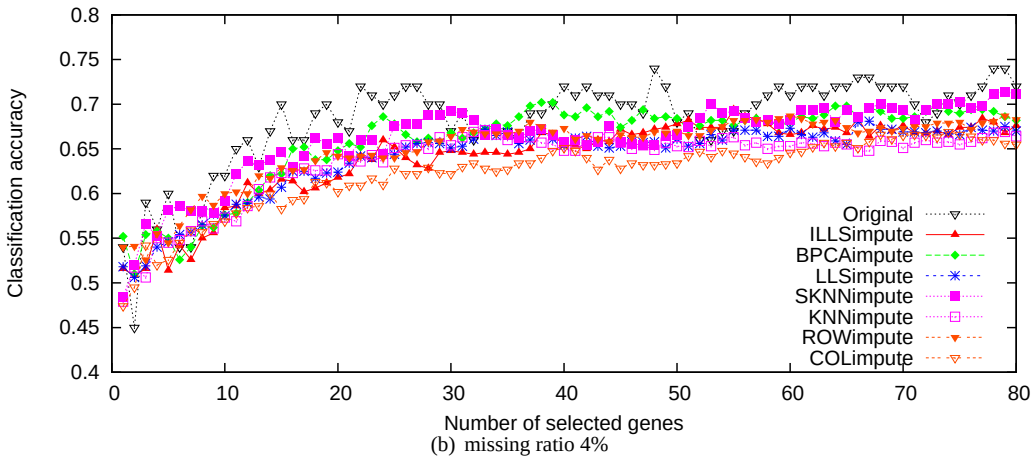


(b) missing ratio 2%

Figure 4.1: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

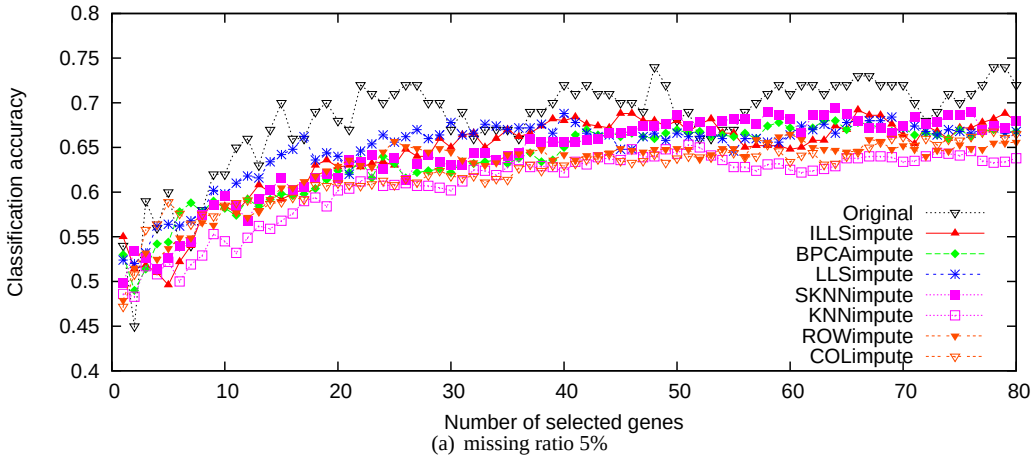


(a) missing ratio 3%

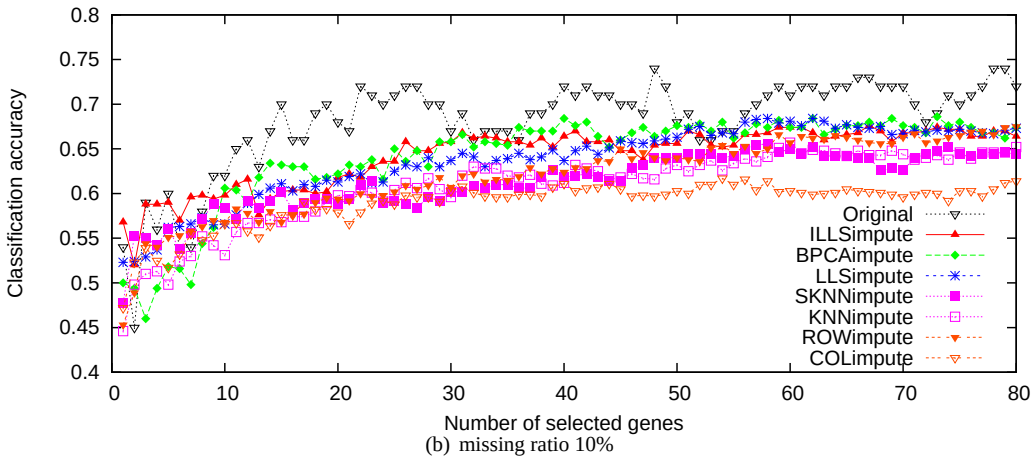


(b) missing ratio 4%

Figure 4.2: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

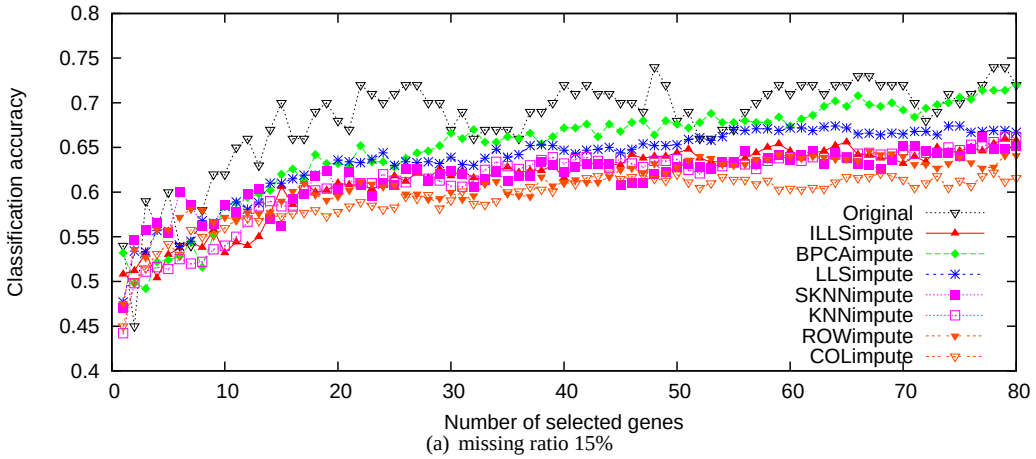


(a) missing ratio 5%

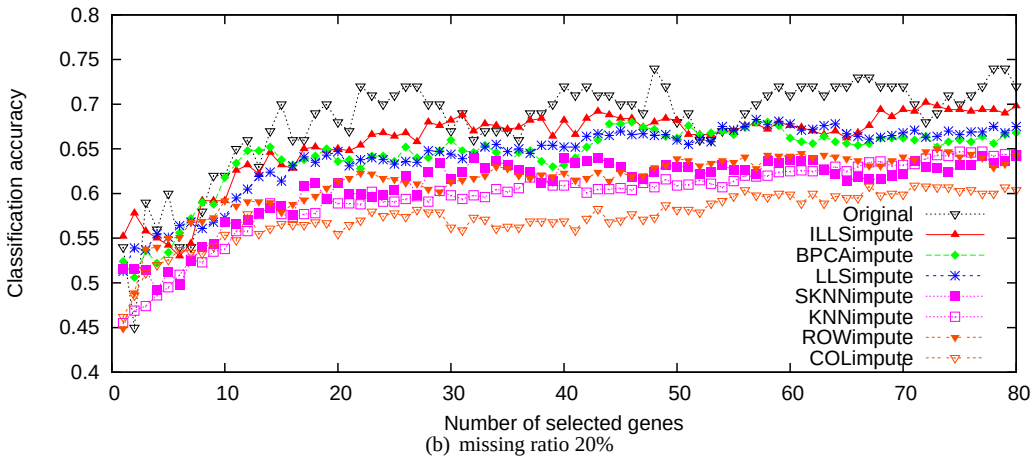


(b) missing ratio 10%

Figure 4.3: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

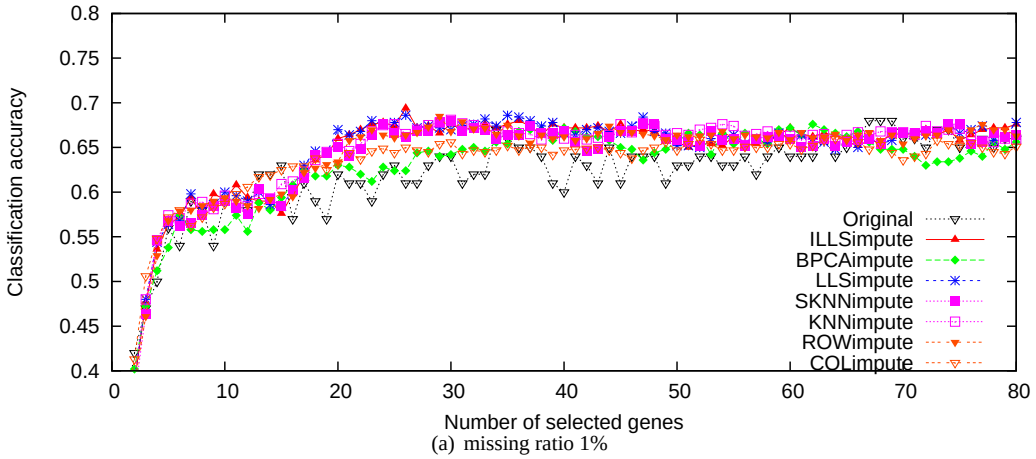


(a) missing ratio 15%

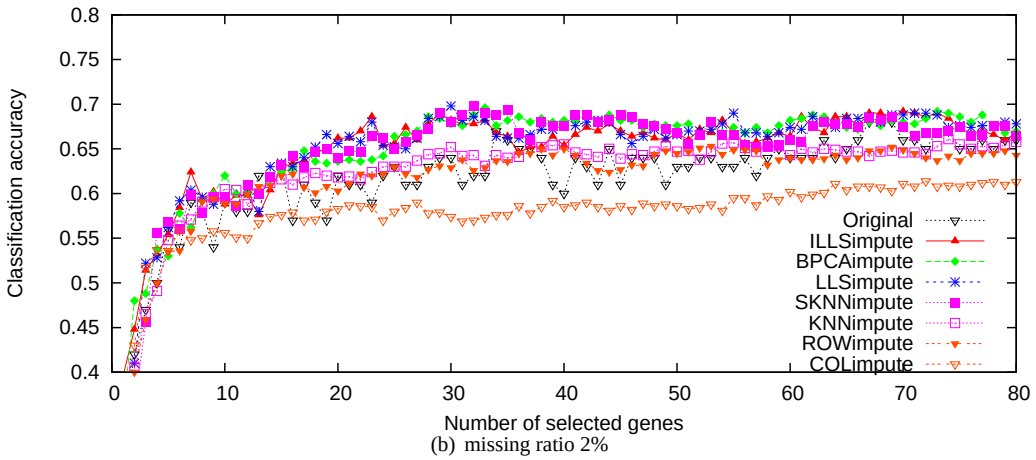


(b) missing ratio 20%

Figure 4.4: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

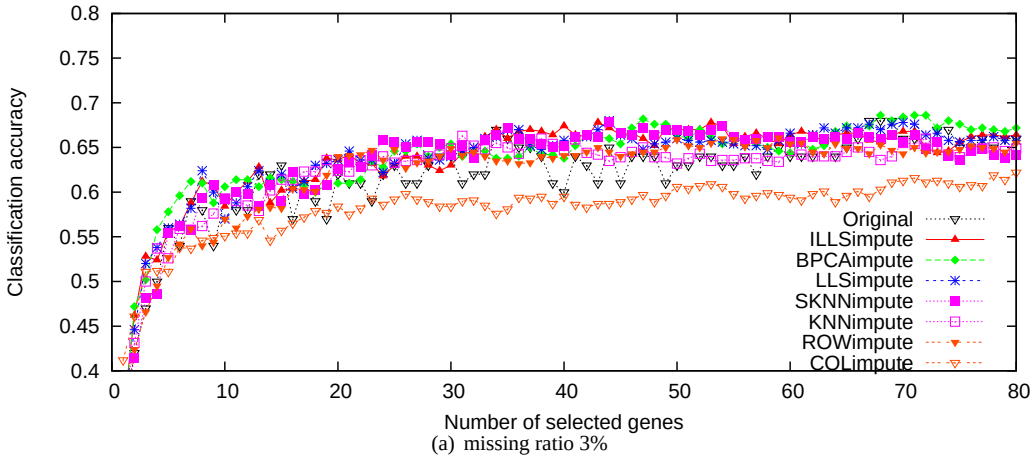


(a) missing ratio 1%

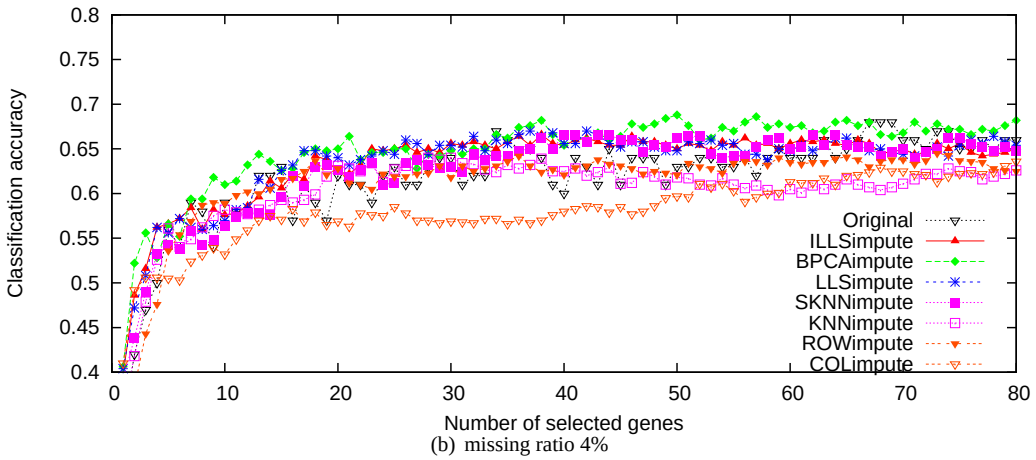


(b) missing ratio 2%

Figure 4.5: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

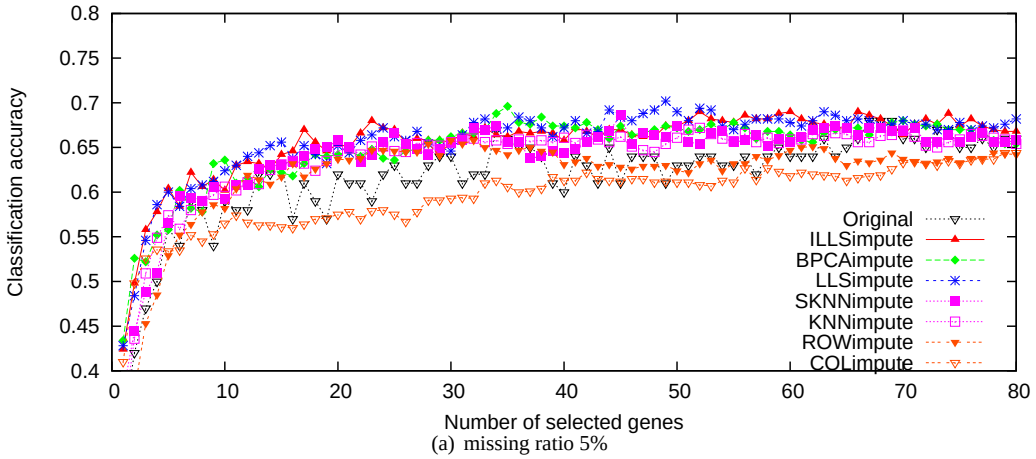


(a) missing ratio 3%

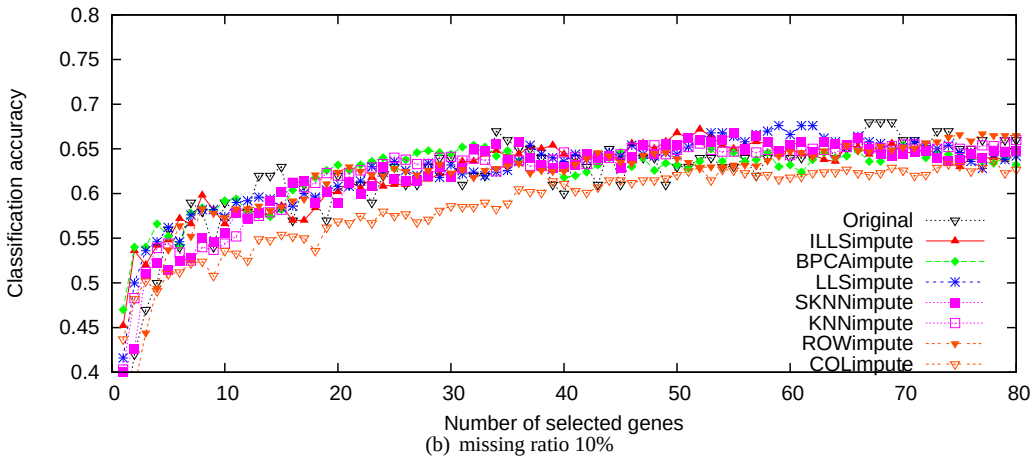


(b) missing ratio 4%

Figure 4.6: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

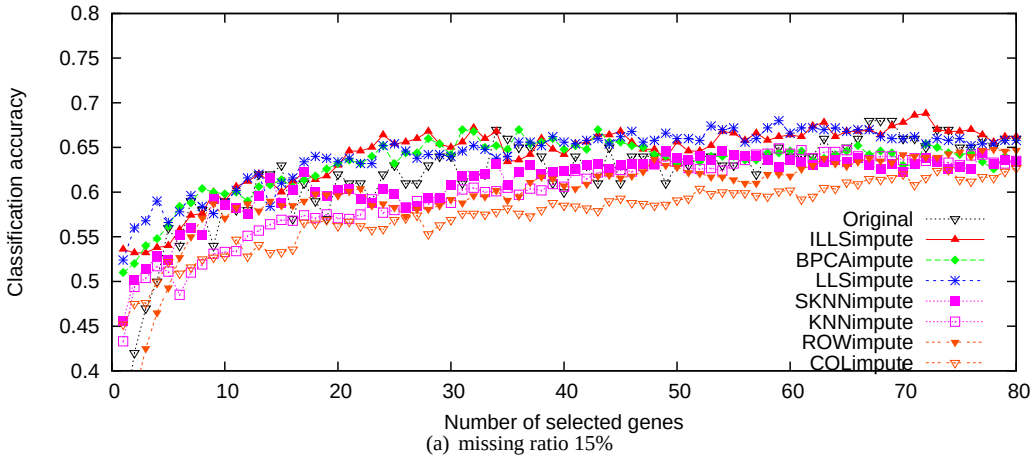


(a) missing ratio 5%

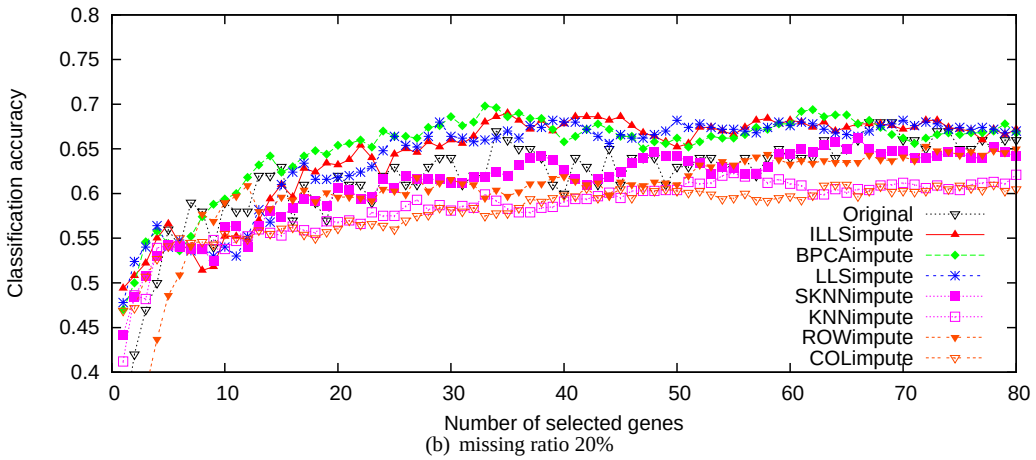


(b) missing ratio 10%

Figure 4.7: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

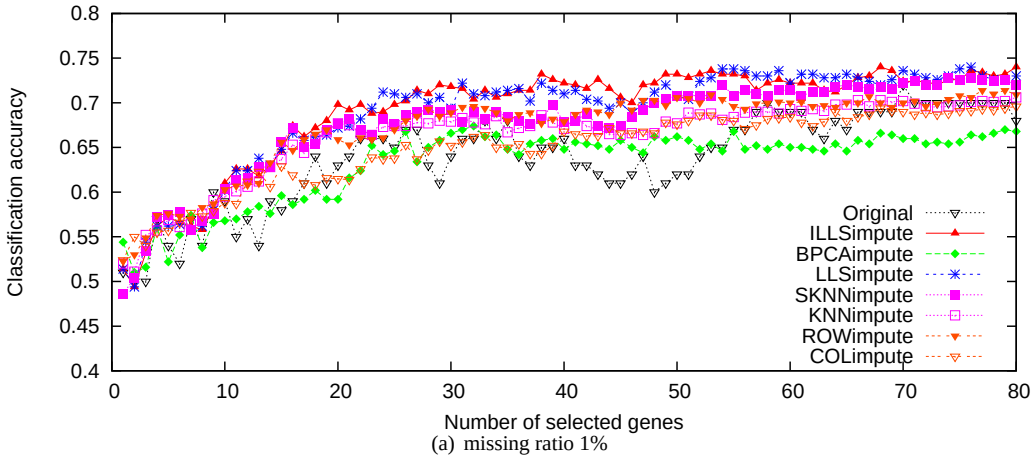


(a) missing ratio 15%

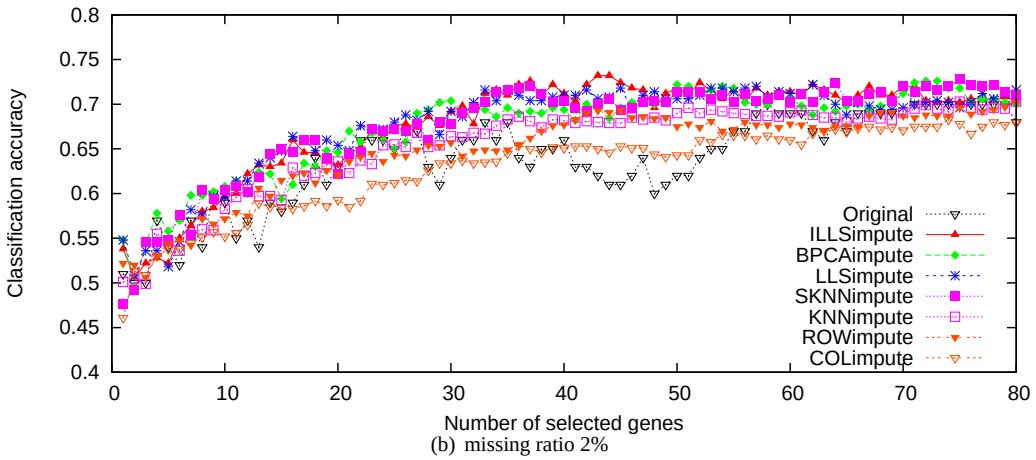


(b) missing ratio 20%

Figure 4.8: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

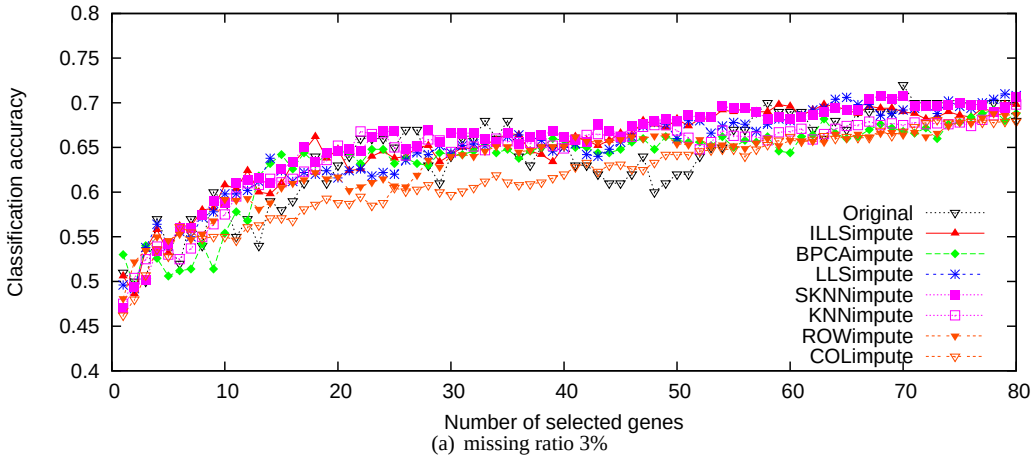


(a) missing ratio 1%

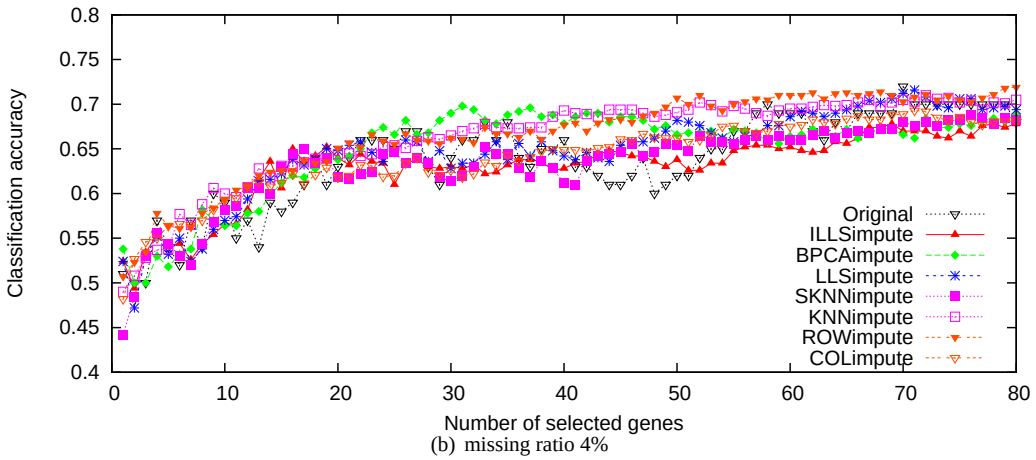


(b) missing ratio 2%

Figure 4.9: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

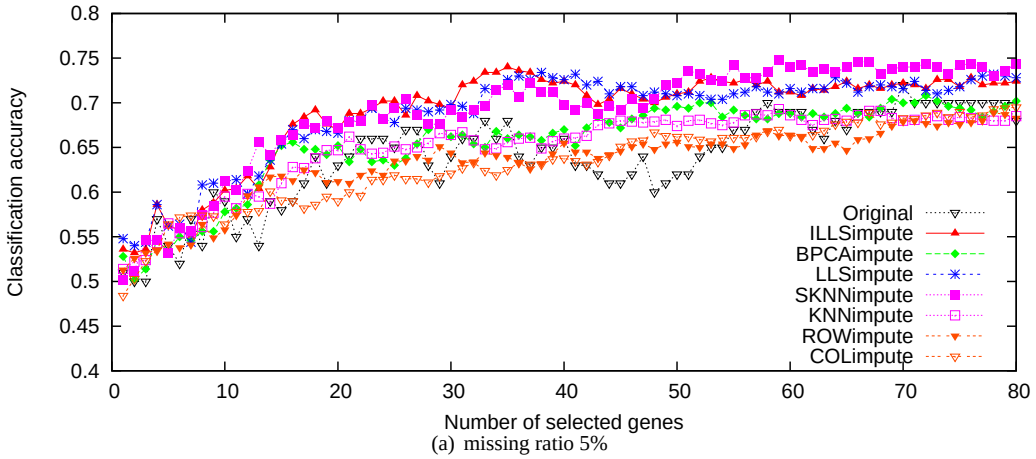


(a) missing ratio 3%

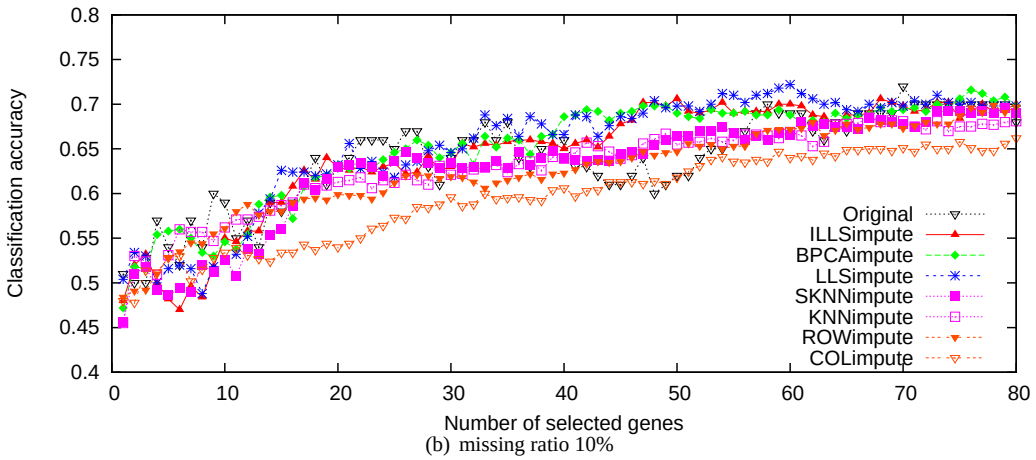


(b) missing ratio 4%

Figure 4.10: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

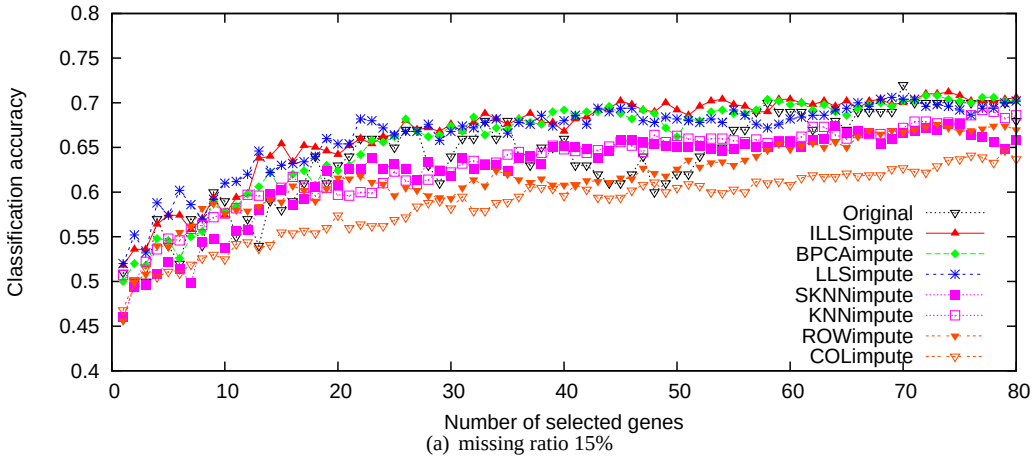


(a) missing ratio 5%

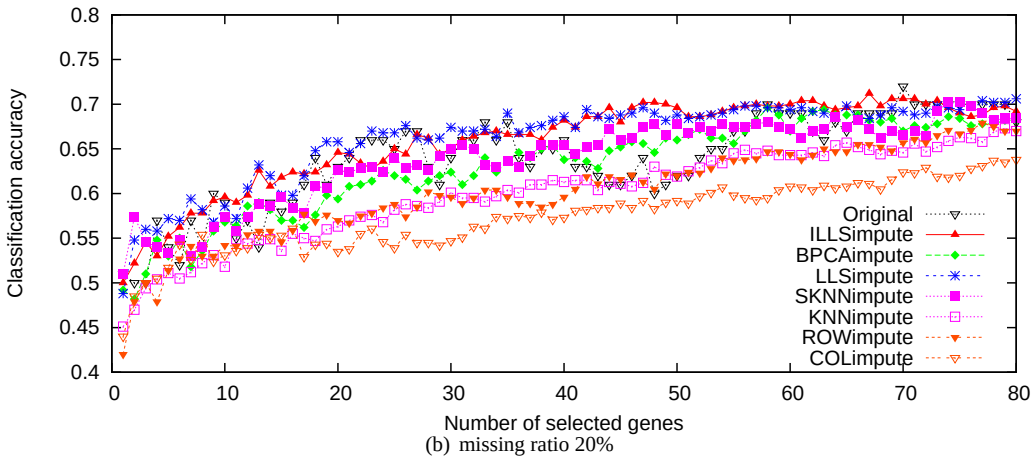


(b) missing ratio 10%

Figure 4.11: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

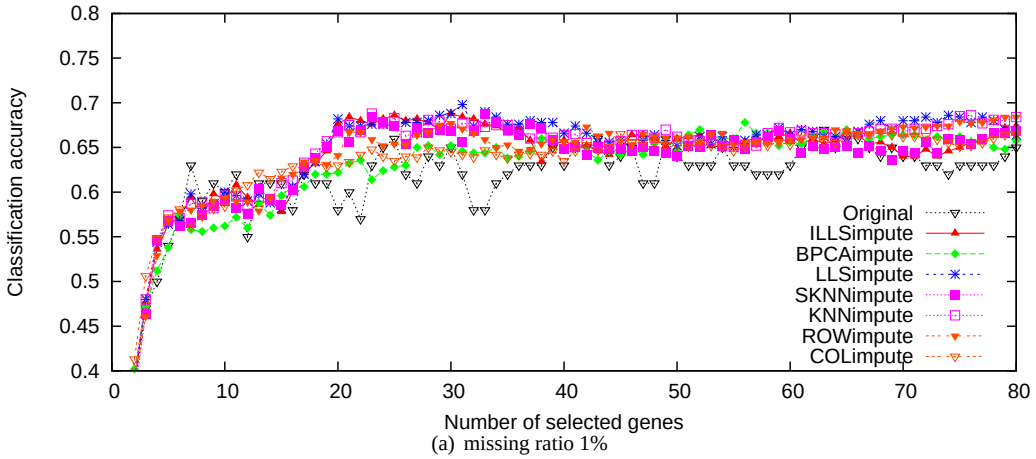


(a) missing ratio 15%

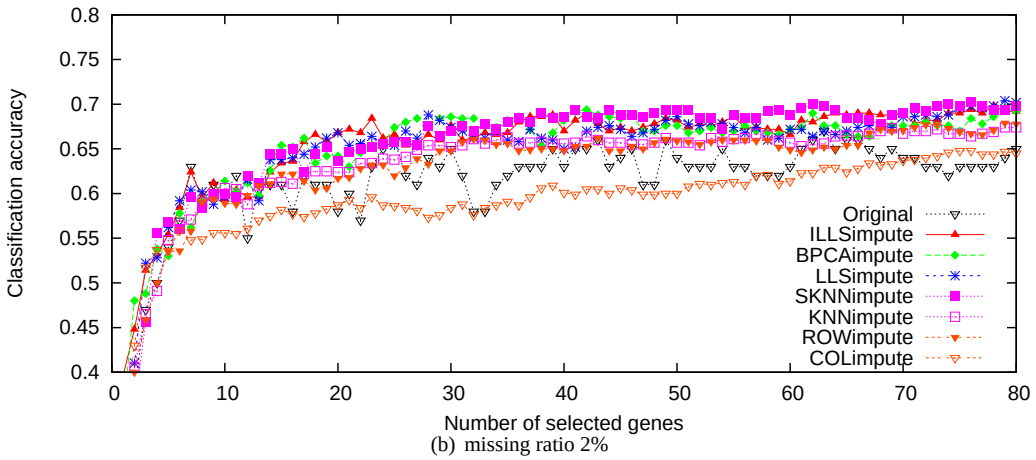


(b) missing ratio 20%

Figure 4.12: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

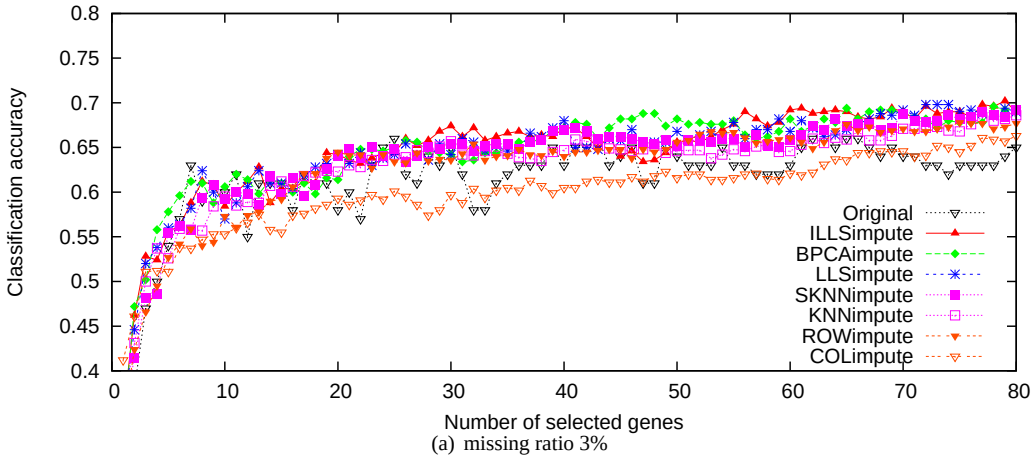


(a) missing ratio 1%

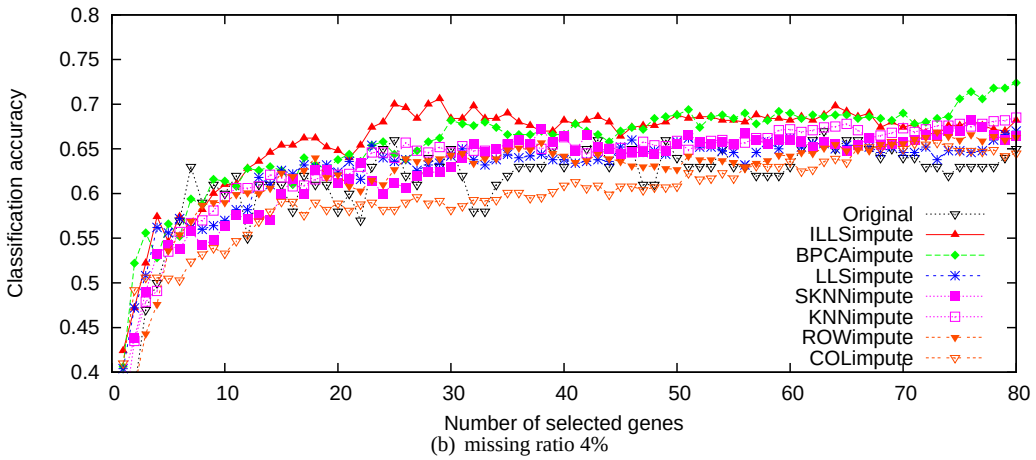


(b) missing ratio 2%

Figure 4.13: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

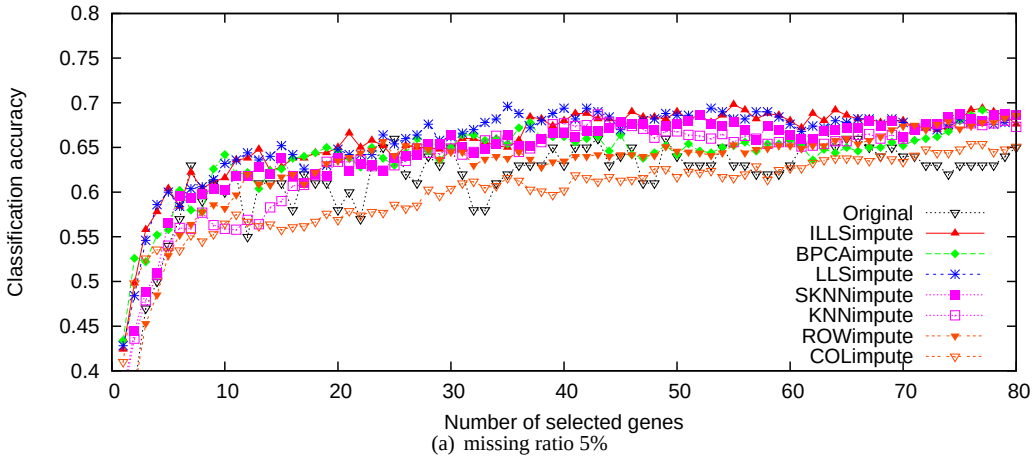


(a) missing ratio 3%

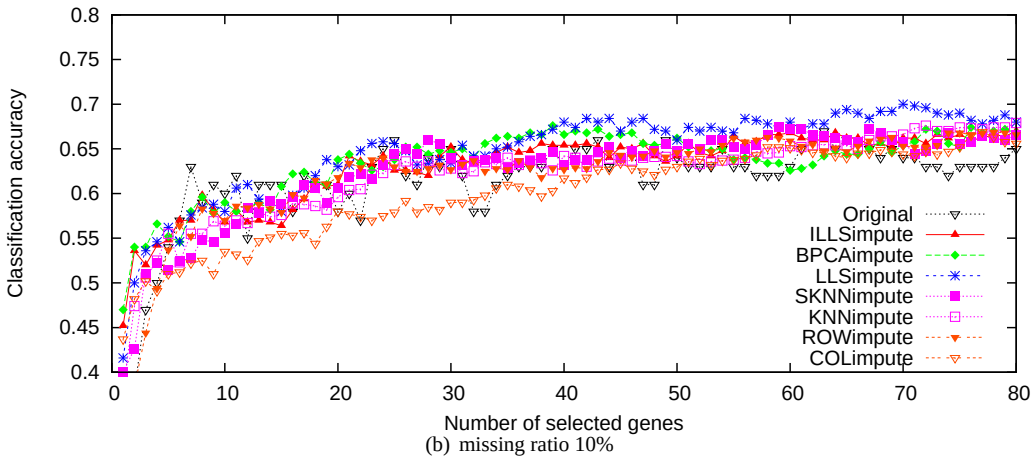


(b) missing ratio 4%

Figure 4.14: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

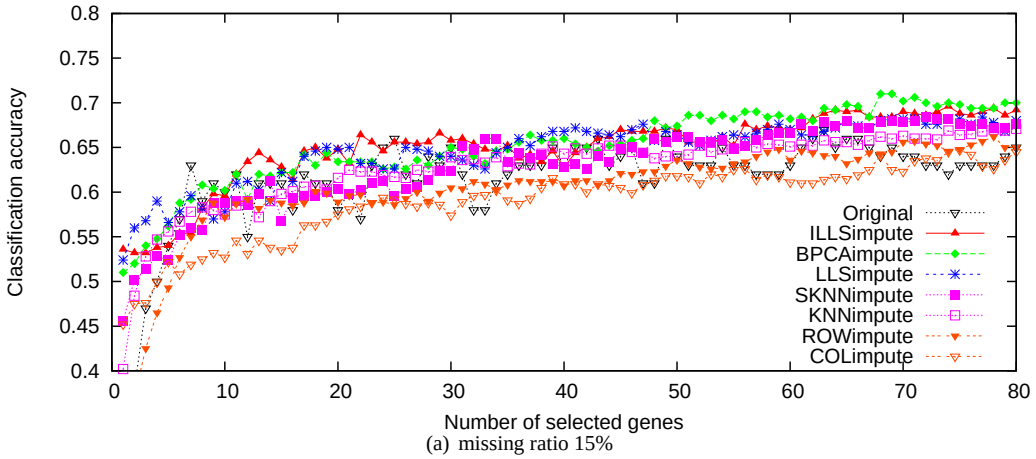


(a) missing ratio 5%

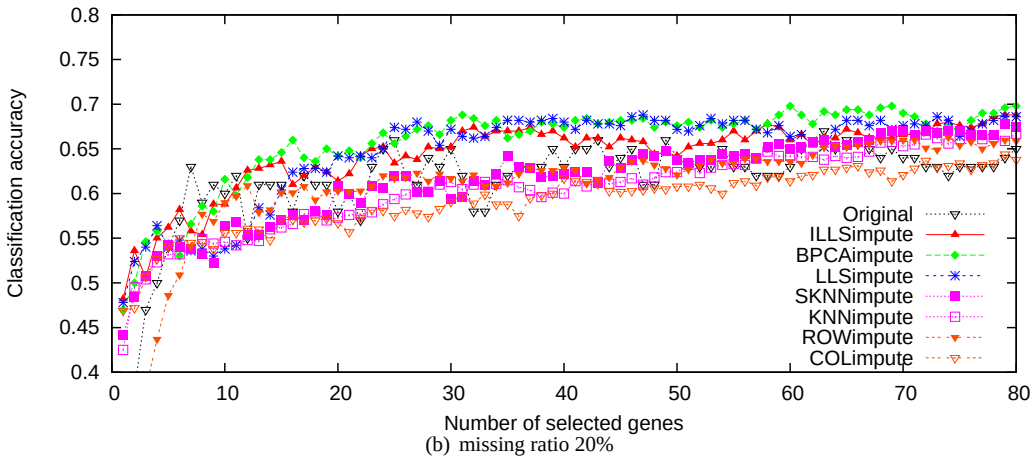


(b) missing ratio 10%

Figure 4.15: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.



(a) missing ratio 15%



(b) missing ratio 20%

Figure 4.16: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

From Figures 4.1–4.16, we can see that using different imputation methods in missing value estimation does affect the subsequent sample classification quality no matter which gene selection method is used. And as the missing ratio increases, the plots of the sample classification accuracies of different missing value imputation methods diverge more from each other. Overall, the qualities of the classification accuracies of some imputation methods, such as ILLSimpute, BPCAimpute, and LLSimpute, consistently outperform the other imputation methods. The performance of SKNNimpute and ROWimpute are better than the performance of KNNimpute, and the performance of KNNimpute is better than COLimpute, which has the lowest performance among all these seven imputation methods. More clearly, Figures 4.4(b), 4.8(b), 4.12(b) 4.16(b) plot the classification accuracies of the seven imputation methods using F-test, Cho, CGS-Ftest, and CGS-Cho, respectively, with missing ratio $r = 20\%$. In Figure 4.4(b), we can see that for the F-test gene selection method, the classification accuracies on the imputed datasets are worse than that achieved on the original dataset. Among the seven imputation methods, the performance order from the best to the worst is ILLSimpute $>$ LLSimpute $>$ BPCAimpute $>$ SKNNimpute $>$ ROWimpute $>$ KNNimpute $>$ COLimpute. In Figure 4.8(b), using the Cho gene selection method, the performances of the seven imputation methods plus the original one are in the order ILLSimpute $>$ LLSimpute $>$ BPCAimpute $>$ Original $>$ SKNNimpute $>$ ROWimpute $>$ KNNimpute $>$ COLimpute. In Figure 4.12(b), ILLSimpute, LLSimpute, Original, and SKNNimpute have similar performance. ROWimpute and KNNimpute perform similarly but not as well as the first three; and COLimpute again has the worst performance. In Figure 4.16(b), the performance order is BPCAimpute $>$ LLSimpute $>$ ILLSimpute $>$ SKNNimpute $>$ KNNimpute $>$ ROWimpute $>$ Original $>$ COLimpute.

By ignoring the detailed gene selection method employed, we calculated the classification accuracy of a missing value imputation method as the average over four values, i.e. the values corresponding to the four gene selection methods, and plotted them on Figures 4.17, 4.18, 4.19 and 4.20. According to these figures, we can say that overall, ILLSimpute, LLSimpute, and BPCAimpute perform equally well whereas BPCAimpute is slightly better than ILLSimpute and LLSimpute when the missing ratio reaches up to 20%. The classification accuracies computed based on the simulated datasets from these three missing value imputation methods are even higher than the classification accuracies achieved based on the original dataset. In the other four imputation methods, SKNNimpute performs better than ROWimpute, ROWimpute performs better than KNNimpute, and KNNimpute performs better than COLimpute, which has the worst performance among all the seven imputation methods.

To test this observation, we do a statistical hypothesis test between each pair of imputation methods based on the classification accuracies calculated from them on gene 40, 60, and 80 being selected and on missing ratio 20%. We collect all the intermediate classification accuracies generated during the 5-fold cross validation and by assuming that each set of the $10 * 10 * 5 = 500$ (10 simulations, 10 cross-validation, 5 folds) classification accuracies from different imputation methods

on the same number genes being selected have the same standard deviation, we use the right-tail t-test, where

$$H_0 : \mu_0 = \mu_1,$$

is the null hypothesis and

$$H_1 : \mu_0 > \mu_1$$

is the alternative hypothesis.

Table 4.1 collects the significant values (p-values) from each t-test. The hypothesis is between μ_0 (imputation methods in rows) and μ_1 (imputation methods in columns). Ranging from 0 to 1, A lower p-value (lower than 0.05) indicates it is more likely that $\mu_0 > \mu_1$ so that the alternative hypothesis H_1 should be accepted and a higher p-value (higher than 0.95) indicates it is more likely $\mu_0 < \mu_1$, and H_0 should be accepted if the p-value is between 0.05 and 0.95. Therefore, according to the p-values in Table 4.1, we can have similar conclusion of the performance of imputation methods as we have from the classification accuracy plots.

Figure 4.21 plots the average NRMSE values of all the seven missing value imputation methods with missing ratio $r = 1\% - 20\%$. Comparing Figure 4.20(b) and Figure 4.21, we can find that the NRMSE values of COLimpute and ROWimpute are quite dissimilar to each other and are far away from the NRMSE values of the other imputation methods, while in sample classification accuracy plots, accuracies of different imputation methods are not that far away from each other. Moreover, according to NRMSE measurement, the imputation quality of KNNimpute is much better than the imputation quality of ROWimpute, while in classification accuracy measurement, it is not. The imputation quality of the other imputation methods in both measurements are in the same order.

4.2.2 The *Carcinomas* Dataset

Similar to the Gliomas dataset, we plot the same set of figures for the Carcinomas dataset. Figures 4.22–4.37 plot the sample classification accuracies of the seven imputation methods with missing ratio $r = 1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%$, and 20% for the four gene selection methods F-test, Cho, CGS-Ftest, and CGS-Cho, respectively. Figures 4.38, 4.39, 4.40, and 4.41 plot the average sample classification accuracies of the seven imputation methods over the four gene selection methods with missing ratio $r = 1\%, 2\%, 3\%, 4\%, 5\%, 10\%, 15\%$, and 20%, respectively. According to these plots, we can see that in general, the performance order from the best to the worst is BPCAimpute $>$ ILLSimpute $>$ LLSimpute $>$ KNNimpute $>$ COLimpute $>$ SKNNimpute $>$ ROWimpute. This order is quite different from the order by NRMSE measurement. Figure 4.42 plots the 20 average NRMSE values of the imputation methods on missing ratio $r = 1\% - 20\%$. According to the NRMSE values in Figure 4.42, the imputation quality order should be BPCAimpute $>$ LLSimpute $>$ ILLSimpute $>$ SKNNimpute $>$ KNNimpute $>$ ROWimpute $>$ COLimpute. Note that in Figure 4.42, the NRMSE values of KNNimpute increase exponentially as the missing ratio increases. We have repeated this particular experiment many times and the same phenomenon was observed. This

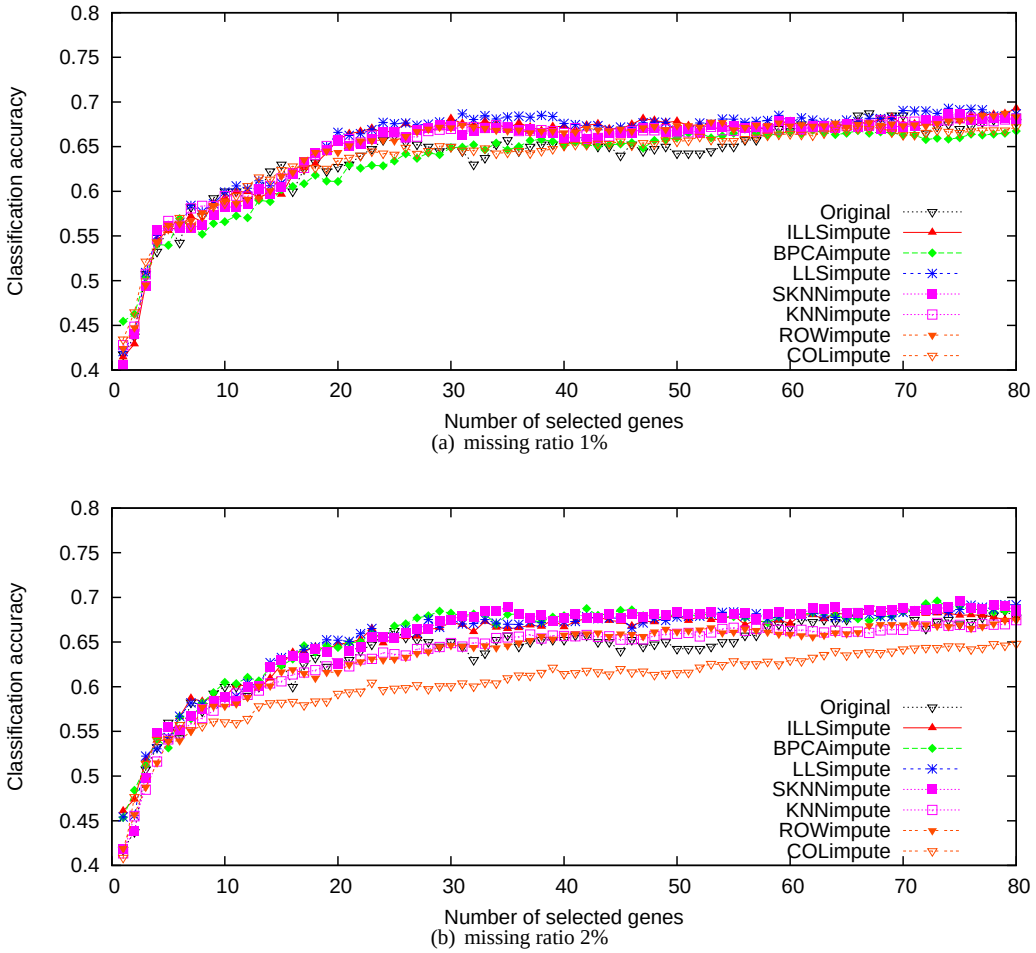


Figure 4.17: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

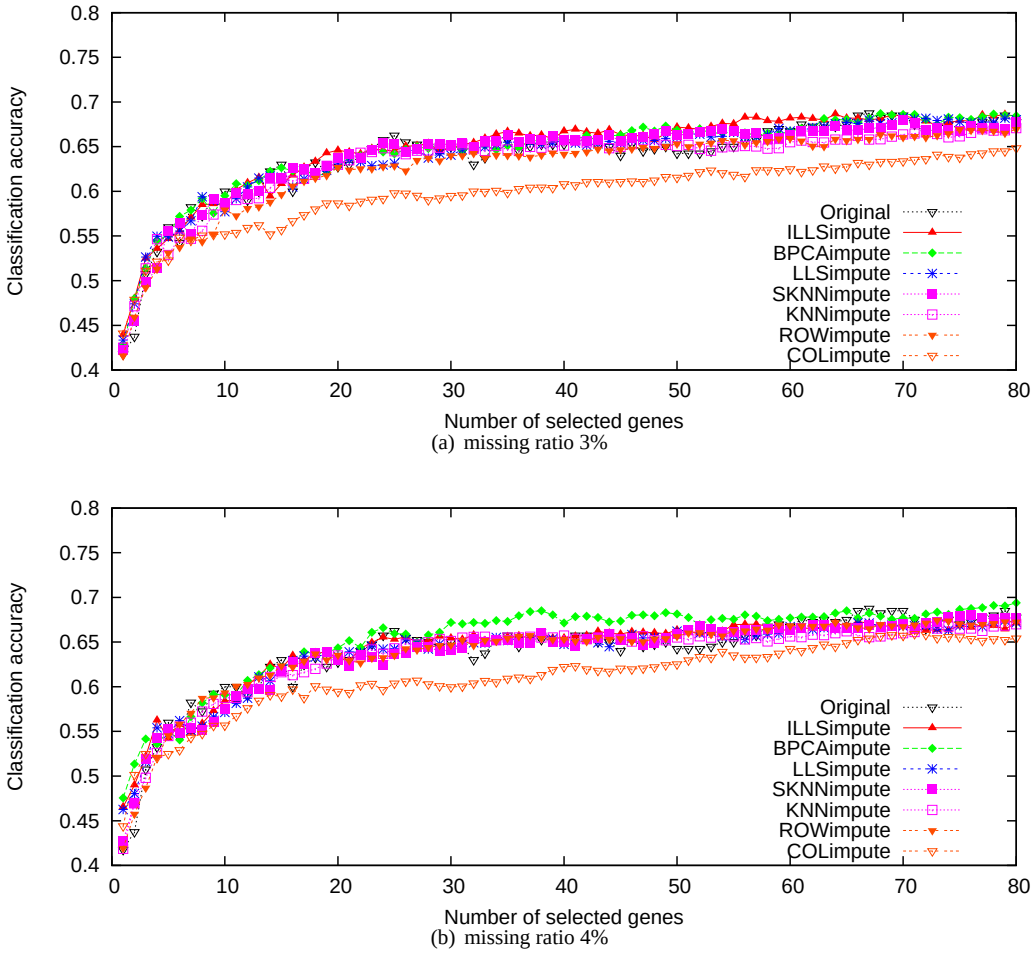


Figure 4.18: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

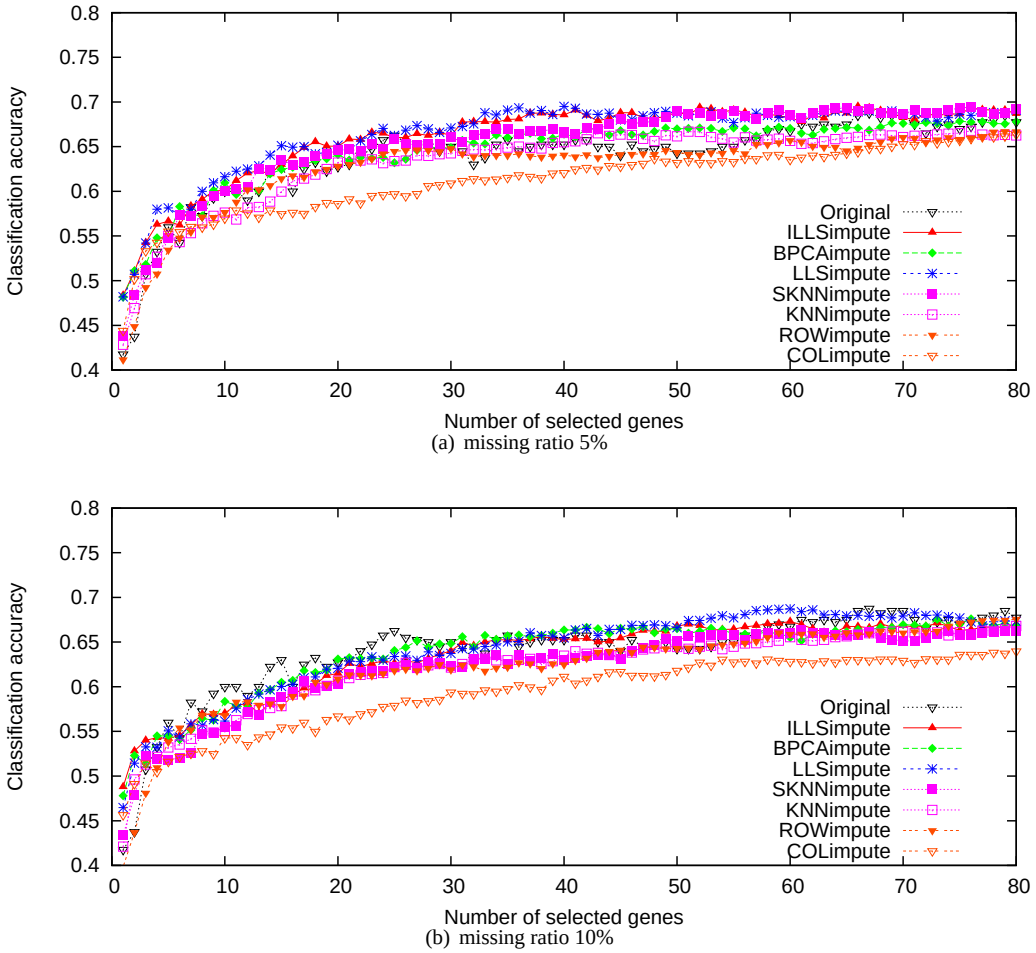
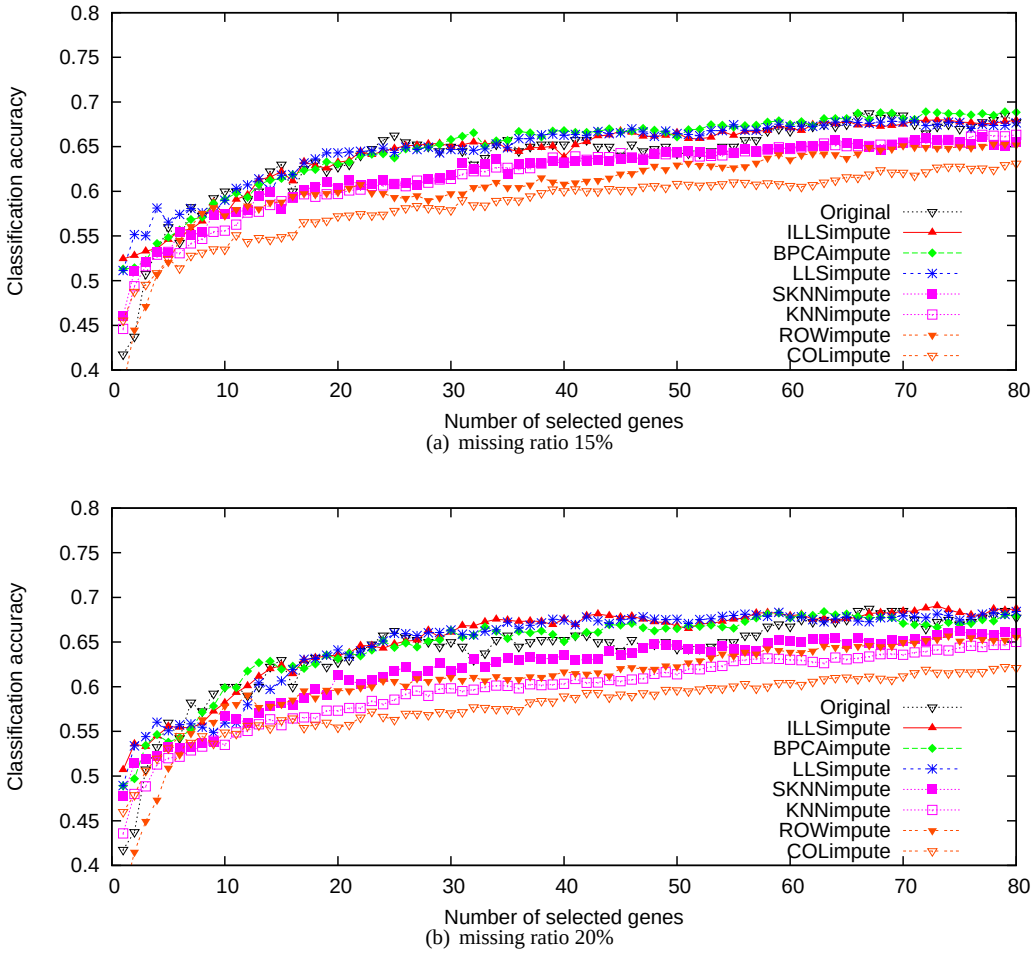


Figure 4.19: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.



(a) missing ratio 15%

(b) missing ratio 20%

Figure 4.20: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Gliomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Gliomas dataset, i.e. $r = 0\%$.

	Gene 40						
	ILLS	BPCA	LLS	SKNN	KNN	COL	ROW
ILLS	0.5000	0.7258	0.3459	0.0000	0.0000	0.0000	0.0000
BPCA	0.2742	0.5000	0.1585	0.0000	0.0000	0.0000	0.0000
LLS	0.6541	0.8415	0.5000	0.0000	0.0000	0.0000	0.0000
SKNN	1.0000	1.0000	1.0000	0.5000	0.0000	0.0000	0.0037
KNN	1.0000	1.0000	1.0000	1.0000	0.5000	0.0000	0.8754
COL	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
ROW	1.0000	1.0000	1.0000	0.9963	0.1246	0.0000	0.5000
	Gene 60						
	ILLS	BPCA	LLS	SKNN	KNN	COL	ROW
ILLS	0.5000	0.9994	0.9732	0.0001	0.0000	0.0000	0.0000
BPCA	0.0006	0.5000	0.0826	0.0000	0.0000	0.0000	0.0000
LLS	0.0268	0.9174	0.5000	0.0000	0.0000	0.0000	0.0000
SKNN	0.9999	1.0000	1.0000	0.5000	0.0005	0.0000	0.0152
KNN	1.0000	1.0000	1.0000	0.9995	0.5000	0.0000	0.8726
COL	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
ROW	1.0000	1.0000	1.0000	0.9848	0.1274	0.0000	0.5000
	Gene 80						
	ILLS	BPCA	LLS	SKNN	KNN	COL	ROW
ILLS	0.5000	0.9966	0.9902	0.0000	0.0000	0.0000	0.0000
BPCA	0.0034	0.5000	0.3528	0.0000	0.0000	0.0000	0.0000
LLS	0.0098	0.6472	0.5000	0.0000	0.0000	0.0000	0.0000
SKNN	1.0000	1.0000	1.0000	0.5000	0.0036	0.0000	0.0807
KNN	1.0000	1.0000	1.0000	0.9964	0.5000	0.0000	0.9006
COL	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000
ROW	1.0000	1.0000	1.0000	0.9193	0.0994	0.0000	0.5000

Table 4.1: P-values (significants) on Gliomas dataset calculated for the right-tail hypothesis test of each pair of imputation methods (row to column) based on the classification accuracies on gene 40, 60, and 80 being selected and missing ratio 20%.

could be due to the fact that as the missing ratio increases, more inaccurate data spots are chosen as missing spots in the simulation process. Their values do not accurately measure the true DNA hybridization intensities while the imputed values could be much closer to the true values. This fact consolidates our conjecture that the NRMSE measurement does have its limitation in measuring the imputation quality, and the sample classification accuracy could be a more suitable measurement for the missing value imputation quality, especially for those datasets having a high percentage of noisy data.

Similarly, to test this observation, we again do the statistical hypothesis test, t-test, between each pair of imputation methods based on the classification accuracies calculated from them on gene 40, 60, and 80 being selected and on missing ratio 20%. In Table 4.2, we can again have similar conclusion concerning the quality of the imputation methods as we have from the classification accuracy plots. In the results of the Carcinomas dataset, we also found the situation where some sample classification accuracies computed based on the simulated dataset are higher than the accuracies computed based on the original dataset.

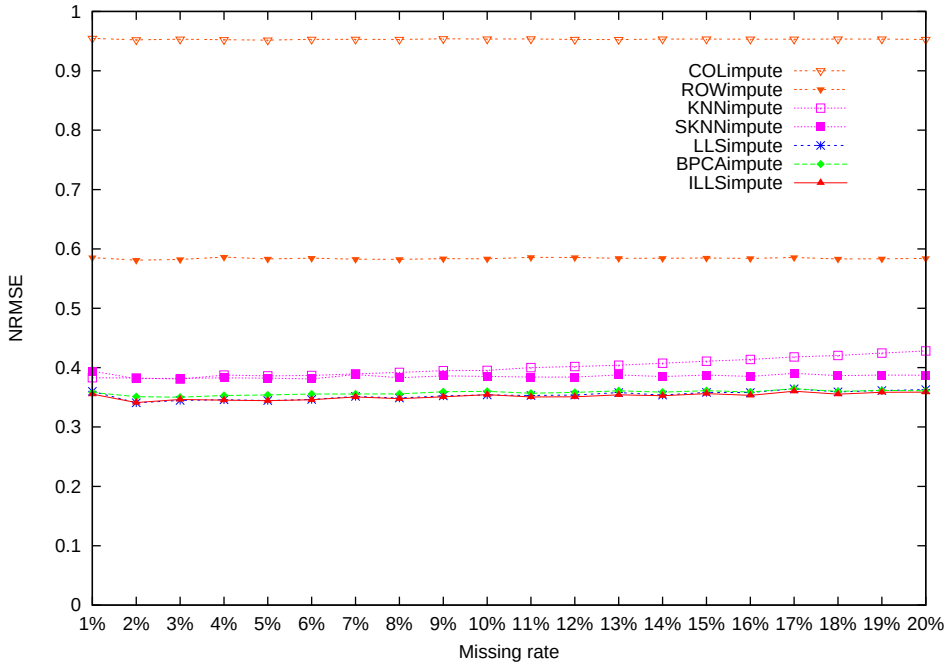
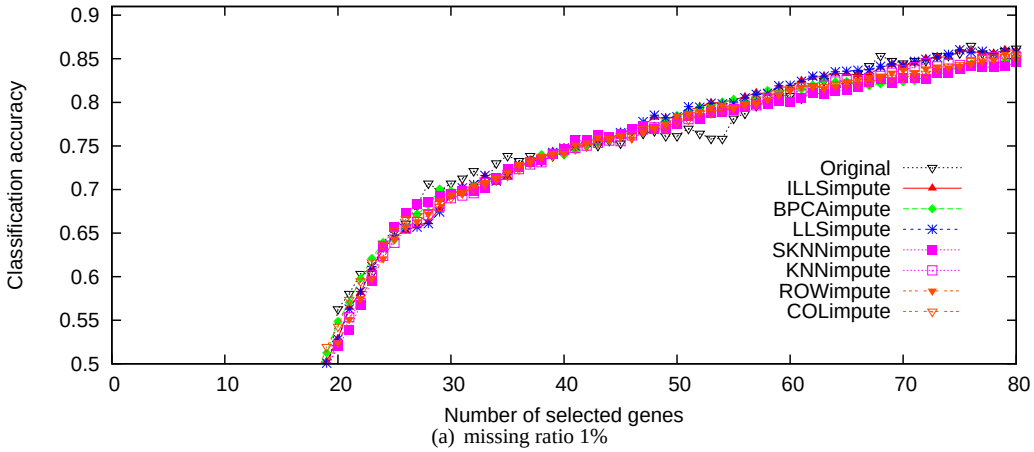
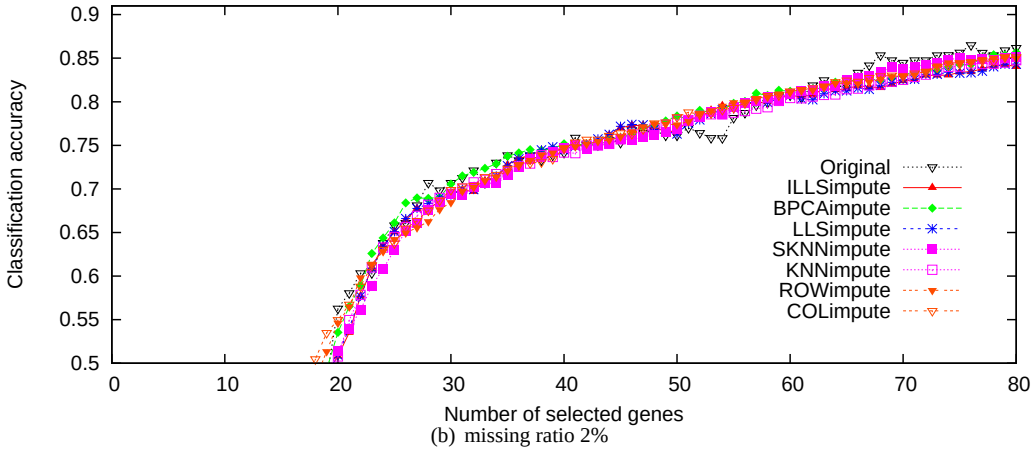


Figure 4.21: The plots of NRMSE values of seven missing value imputation methods ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute on Gliomas dataset.

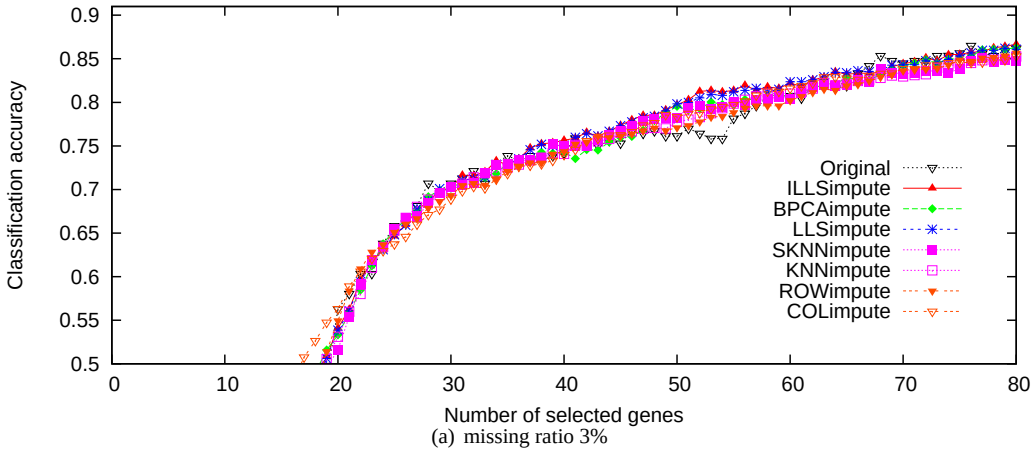


(a) missing ratio 1%

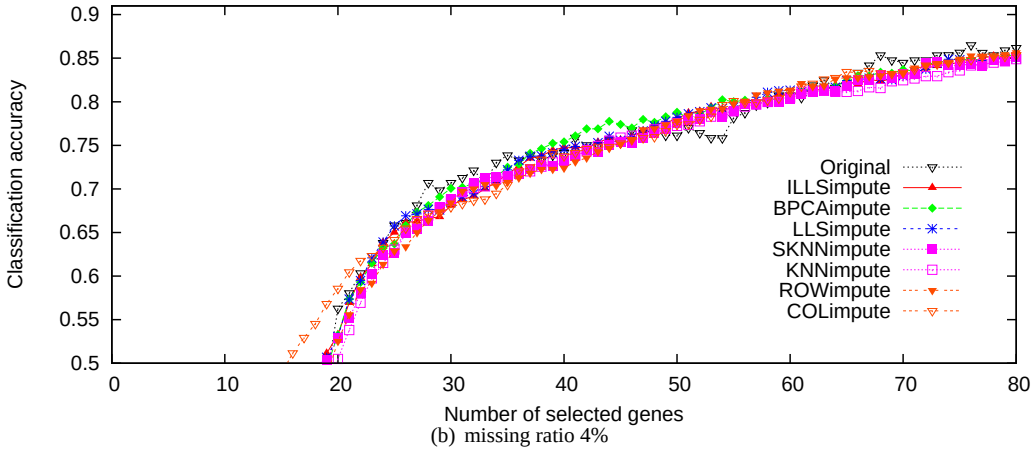


(b) missing ratio 2%

Figure 4.22: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

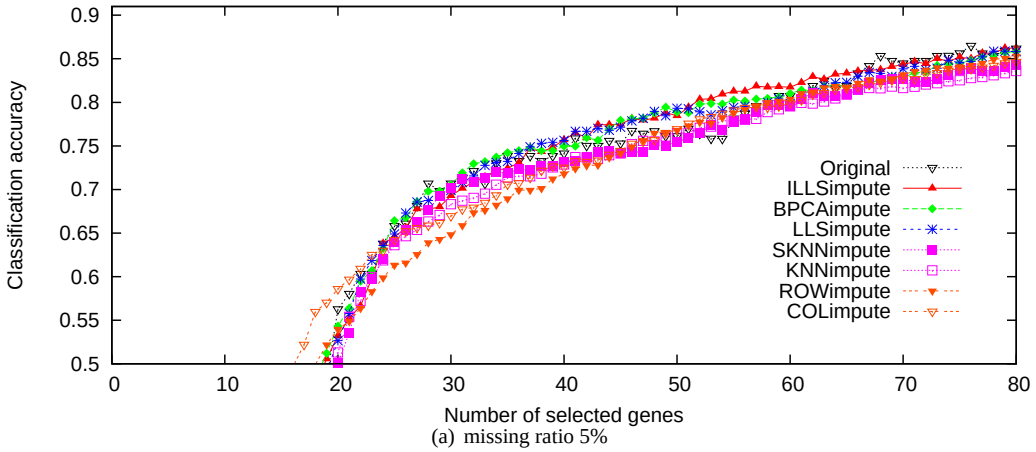


(a) missing ratio 3%

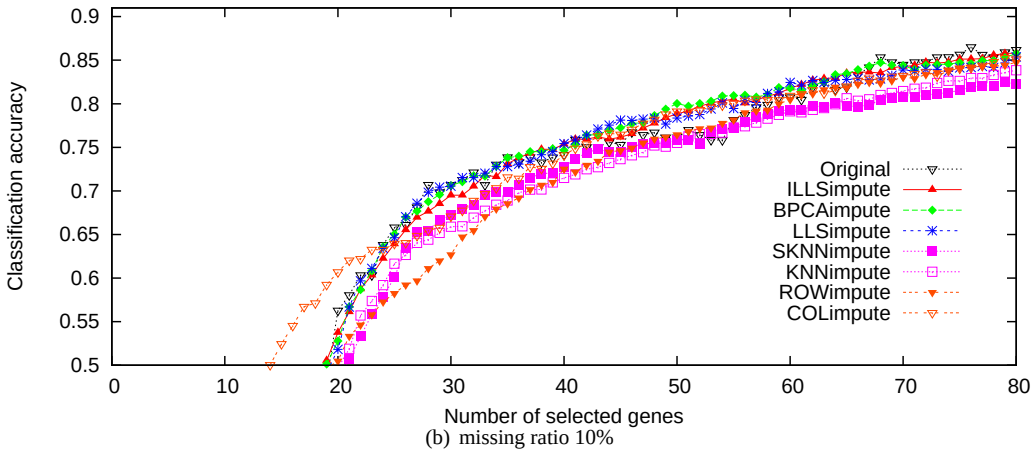


(b) missing ratio 4%

Figure 4.23: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

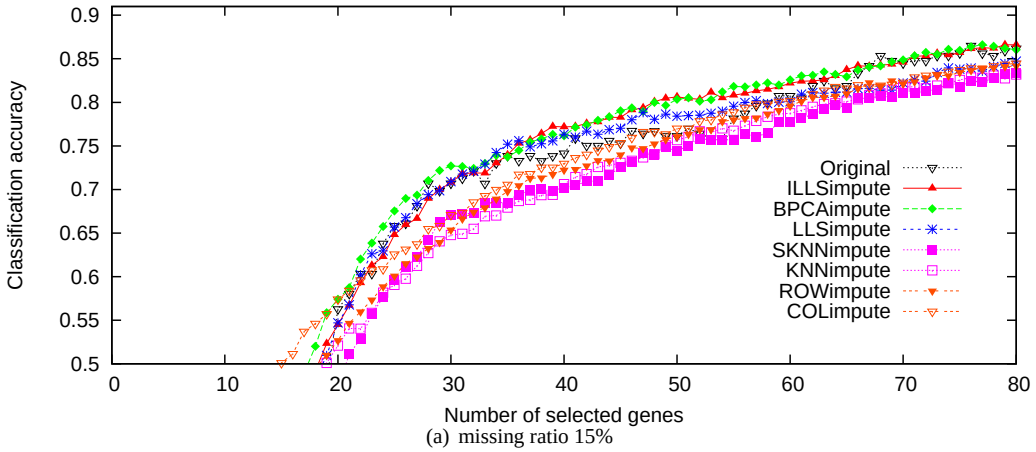


(a) missing ratio 5%

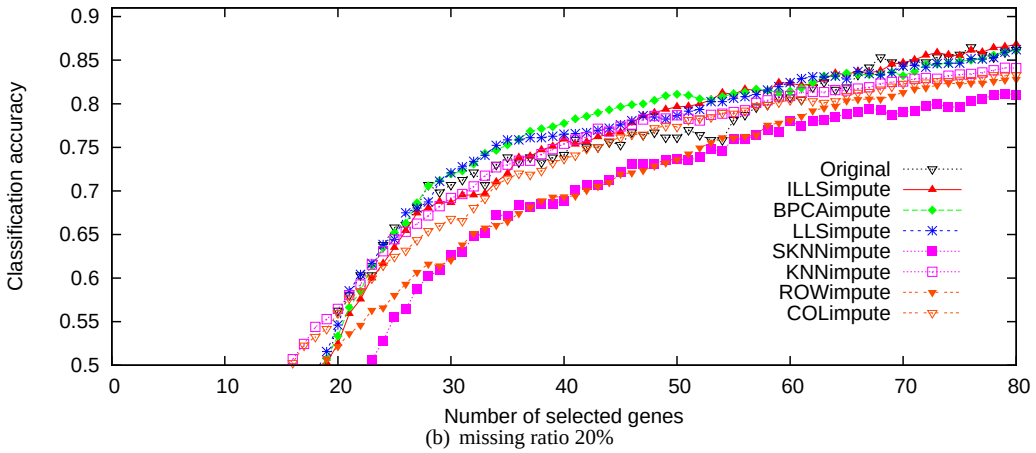


(b) missing ratio 10%

Figure 4.24: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

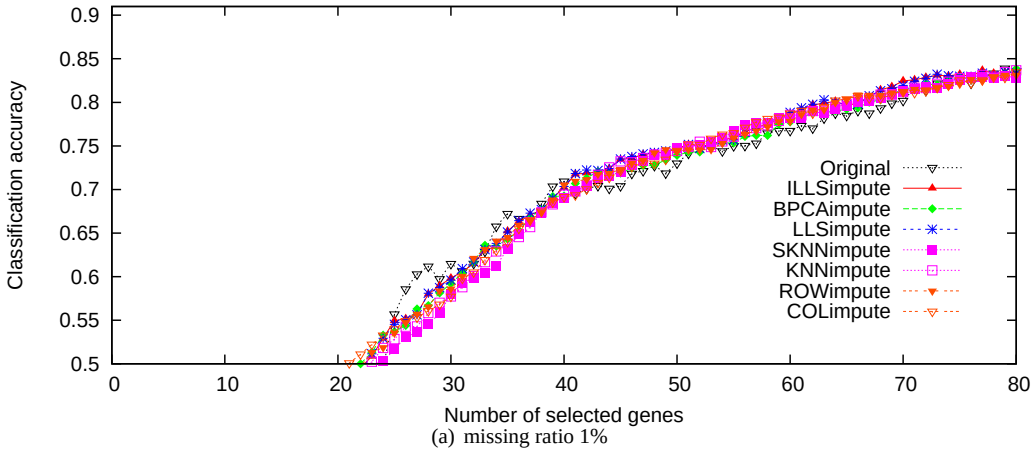


(a) missing ratio 15%

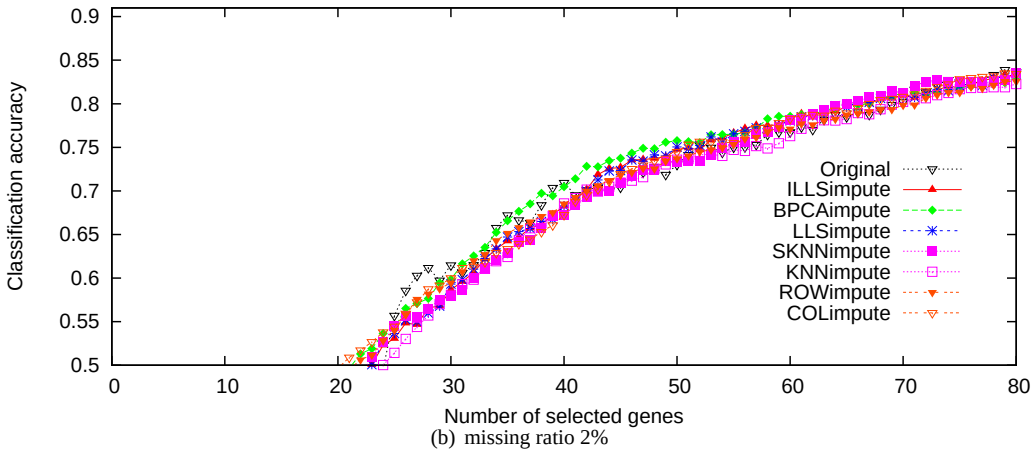


(b) missing ratio 20%

Figure 4.25: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the F-test method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

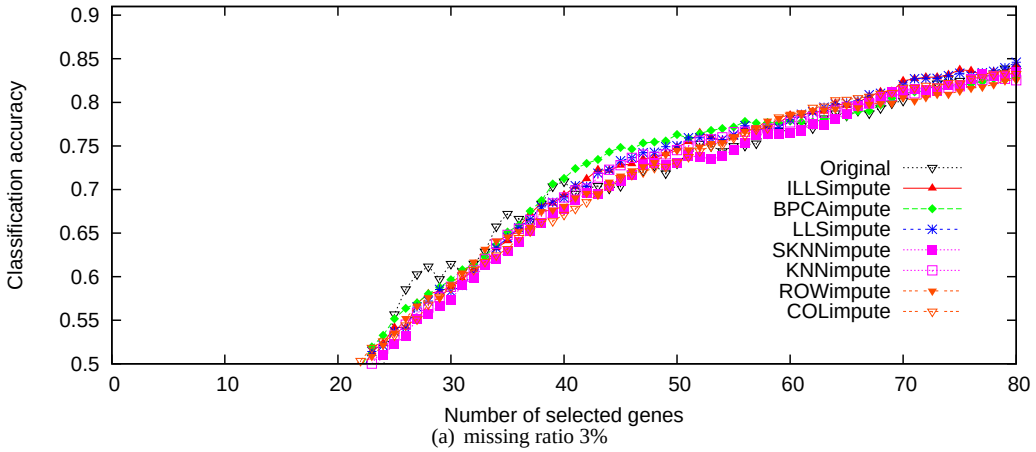


(a) missing ratio 1%

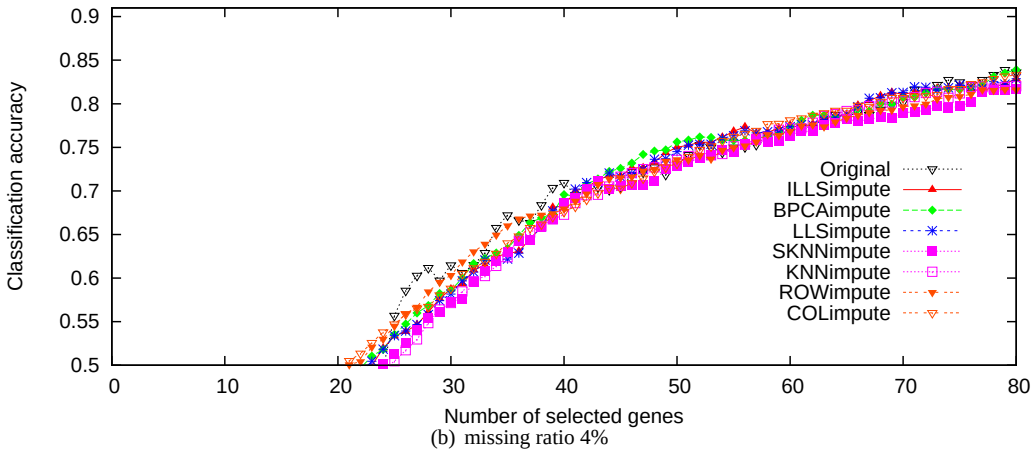


(b) missing ratio 2%

Figure 4.26: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

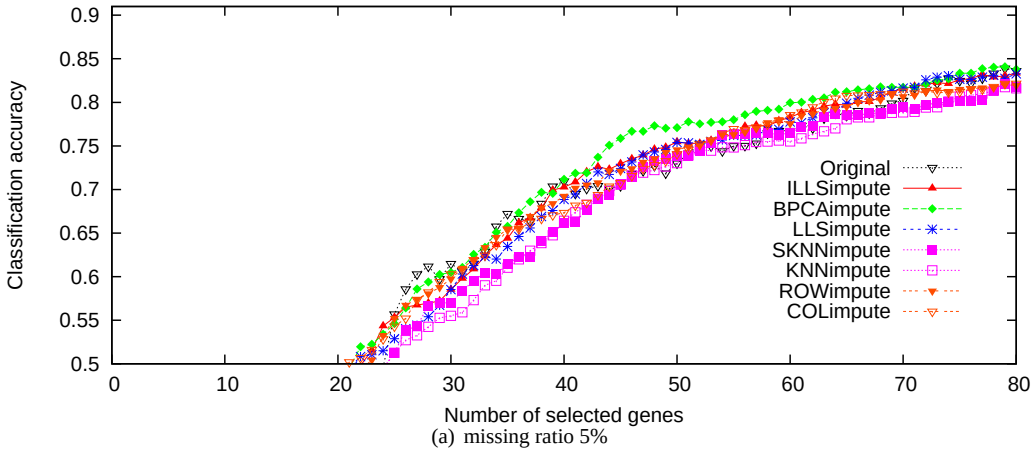


(a) missing ratio 3%

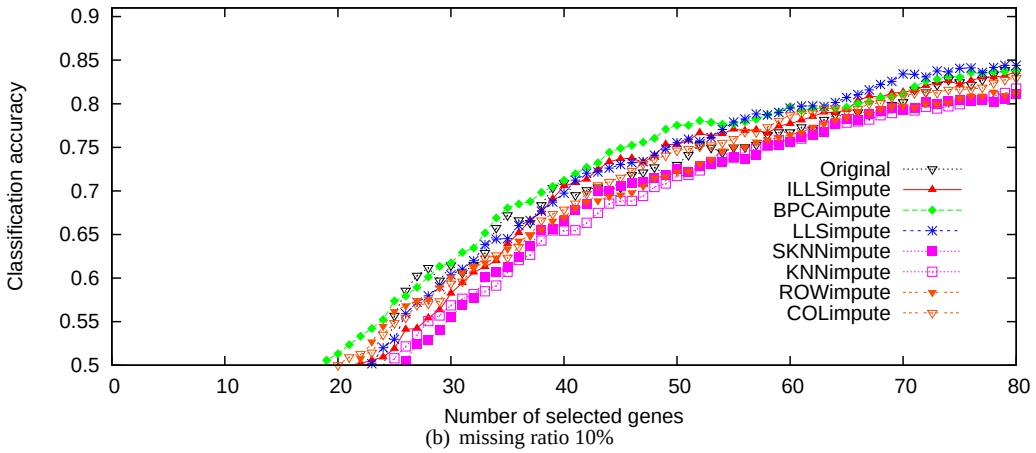


(b) missing ratio 4%

Figure 4.27: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

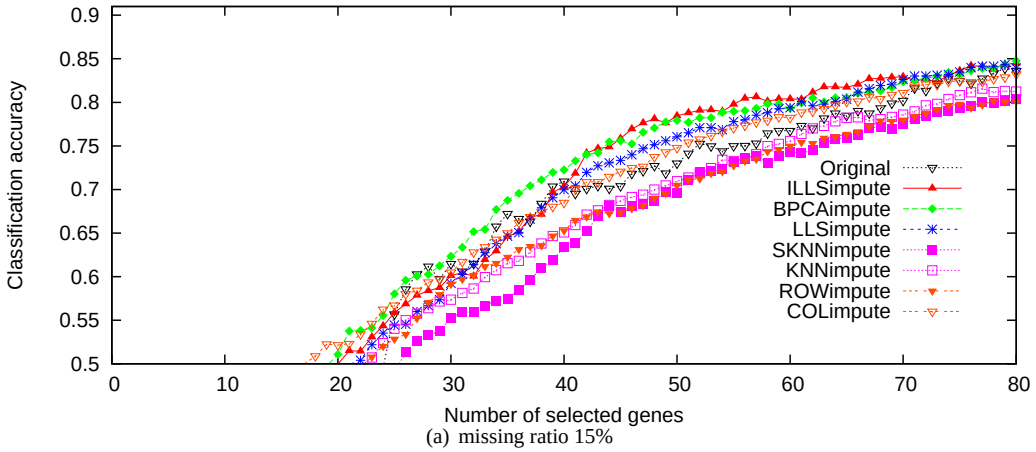


(a) missing ratio 5%

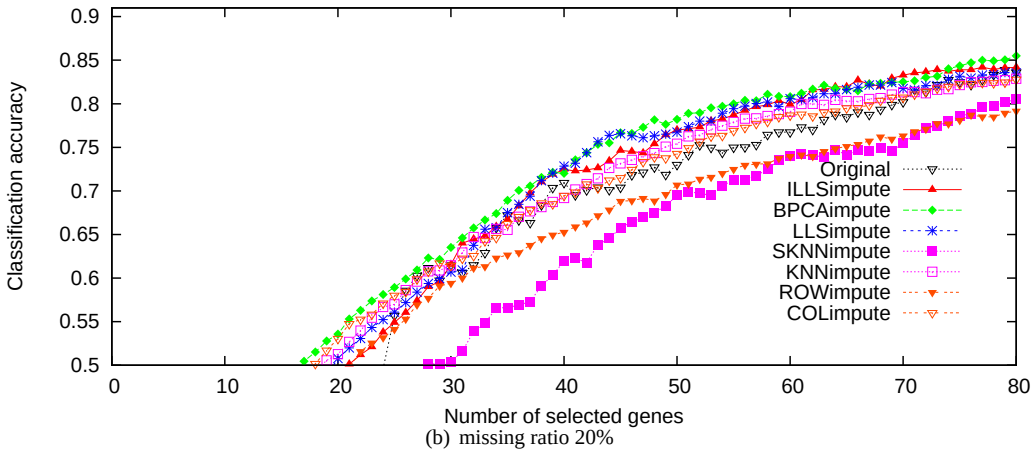


(b) missing ratio 10%

Figure 4.28: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

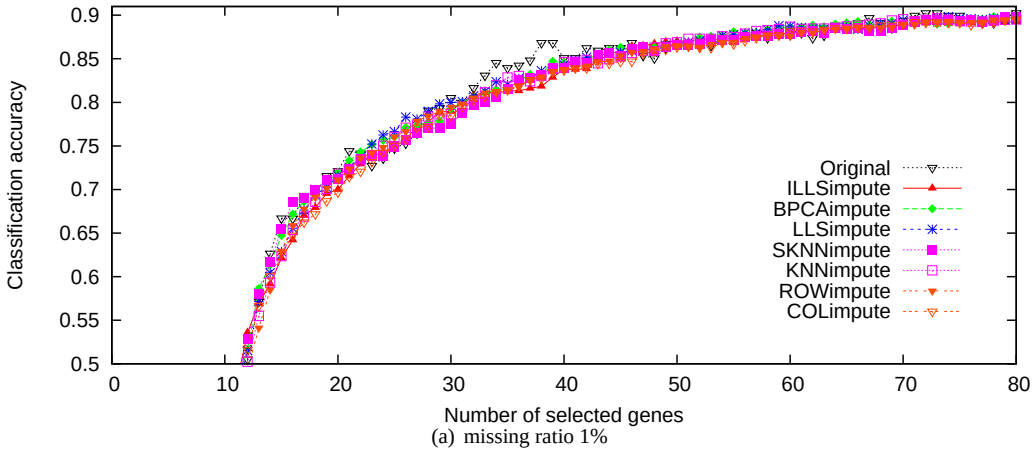


(a) missing ratio 15%

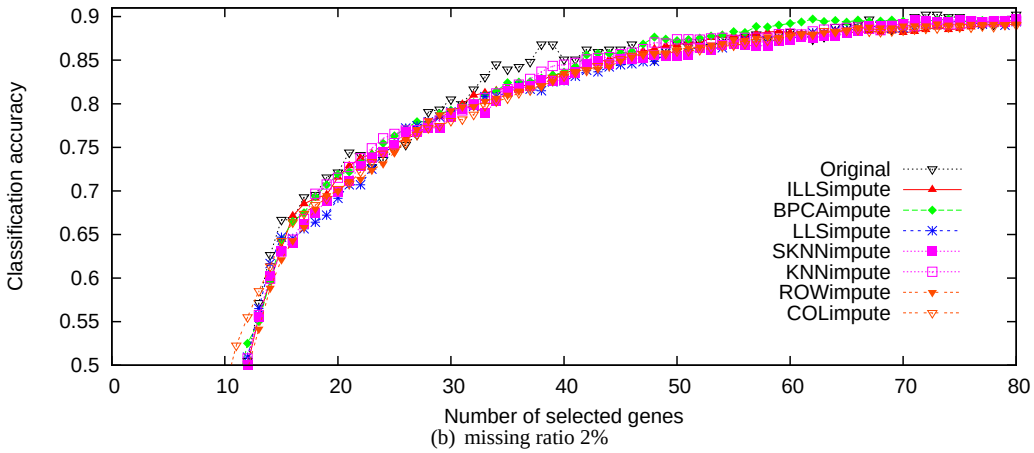


(b) missing ratio 20%

Figure 4.29: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

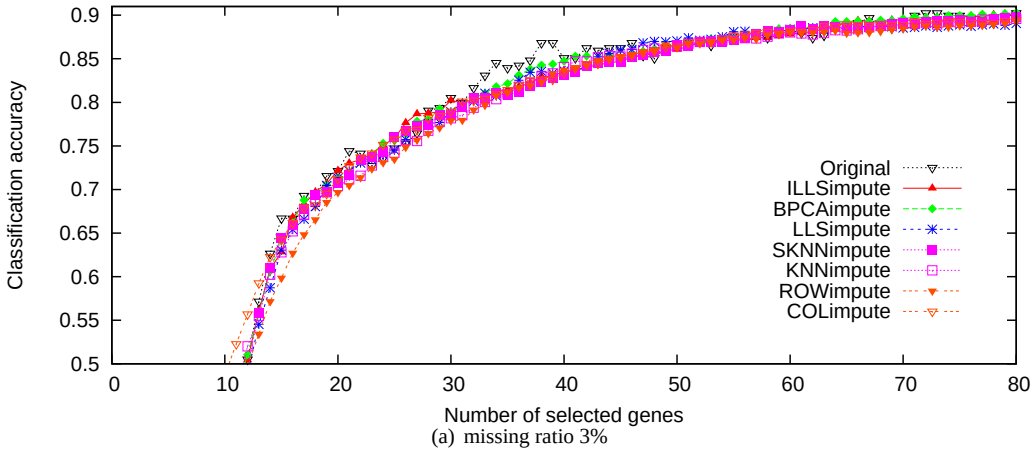


(a) missing ratio 1%

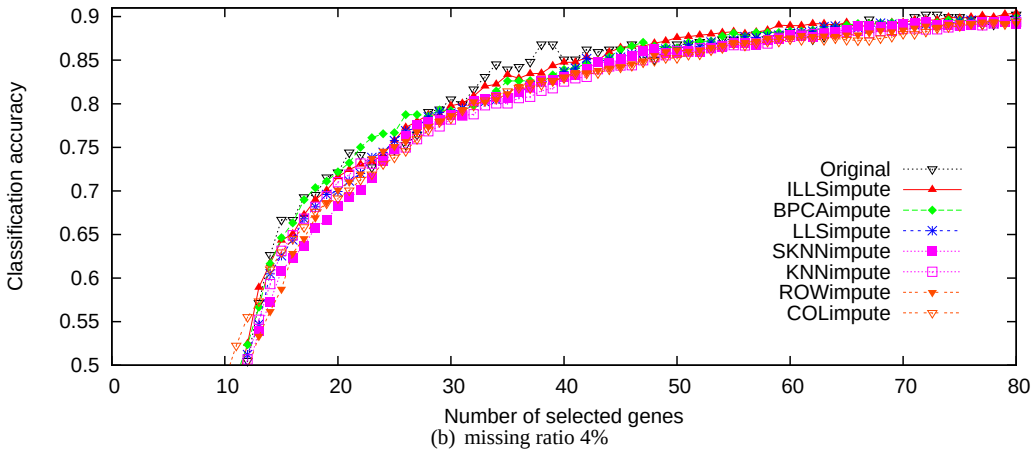


(b) missing ratio 2%

Figure 4.30: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

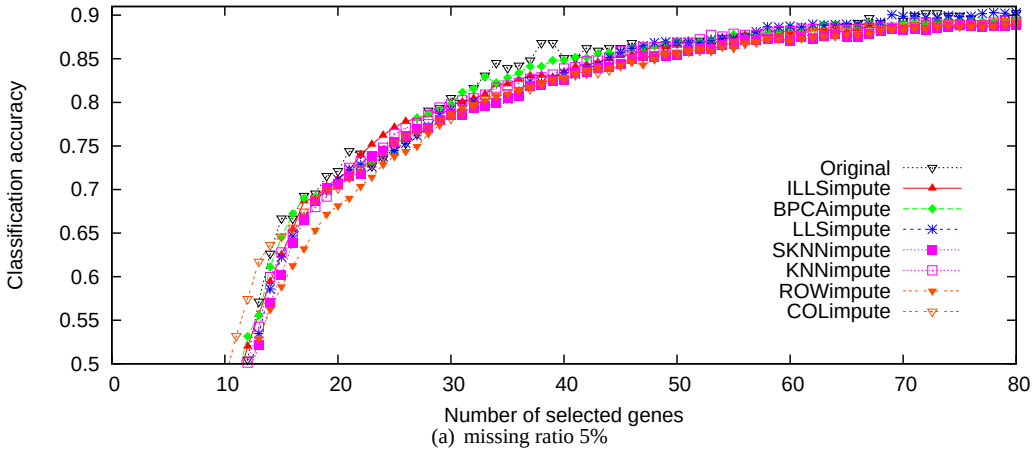


(a) missing ratio 3%

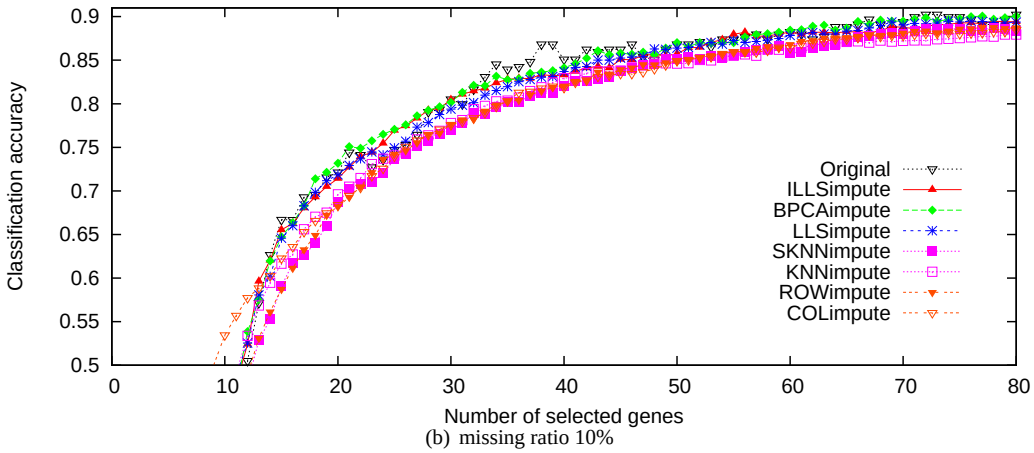


(b) missing ratio 4%

Figure 4.31: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

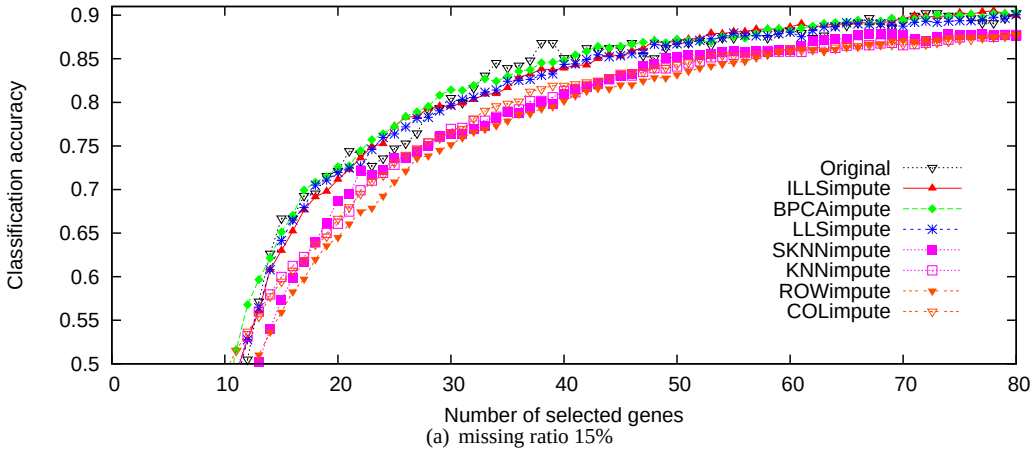


(a) missing ratio 5%

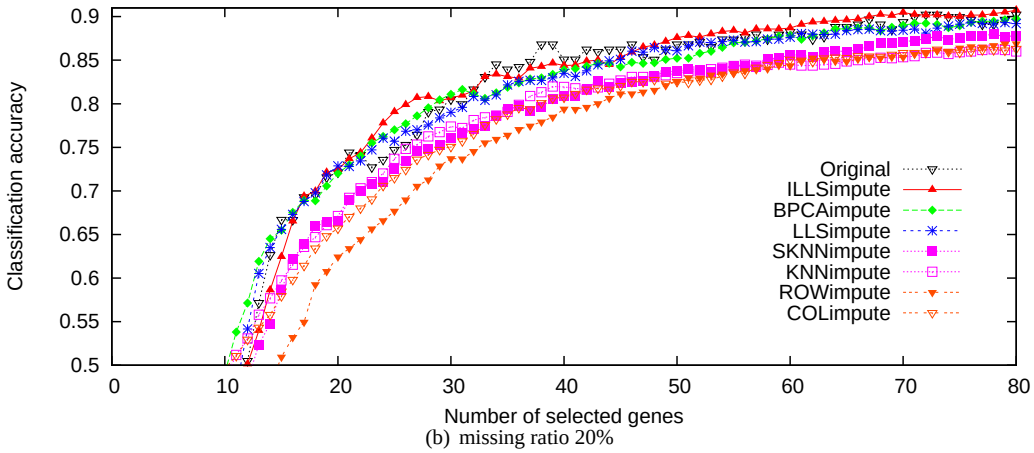


(b) missing ratio 10%

Figure 4.32: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

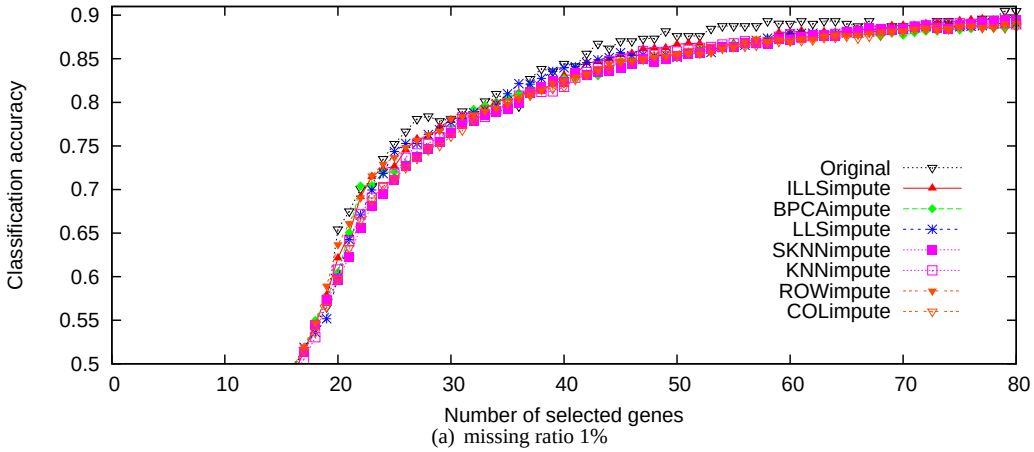


(a) missing ratio 15%

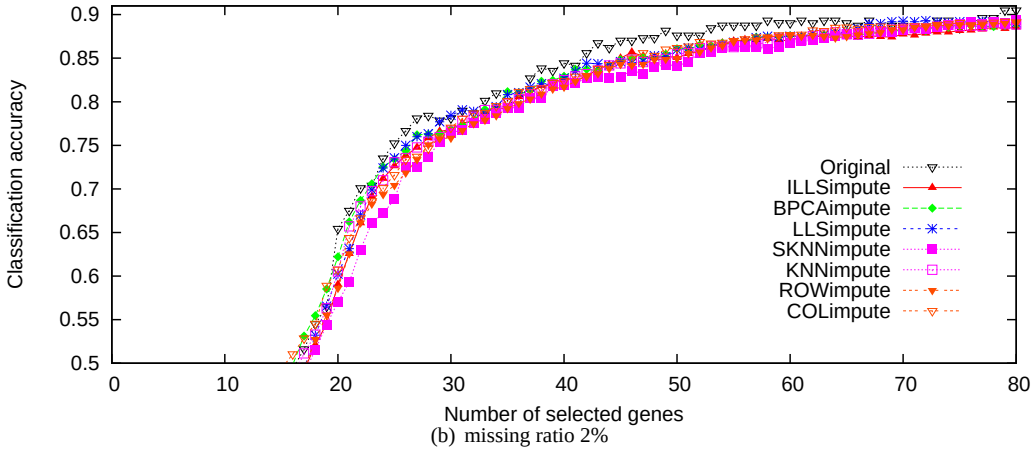


(b) missing ratio 20%

Figure 4.33: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Ftest method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

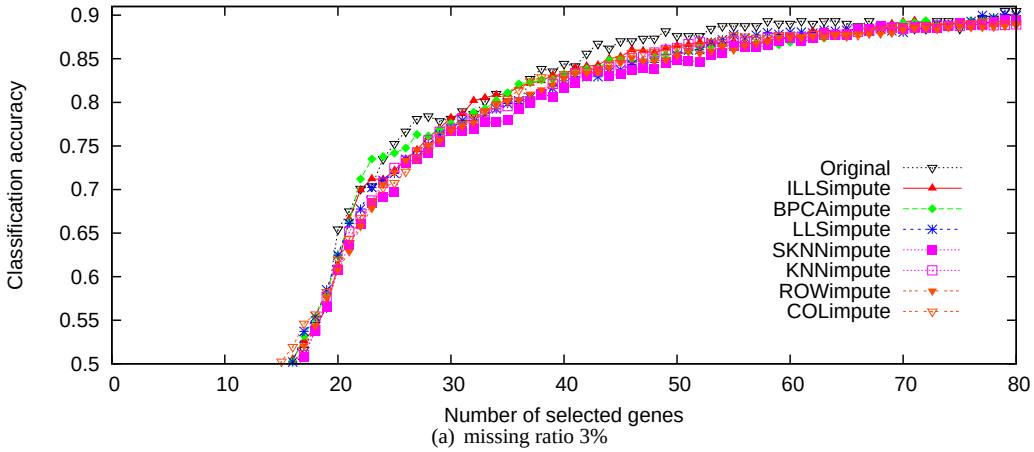


(a) missing ratio 1%

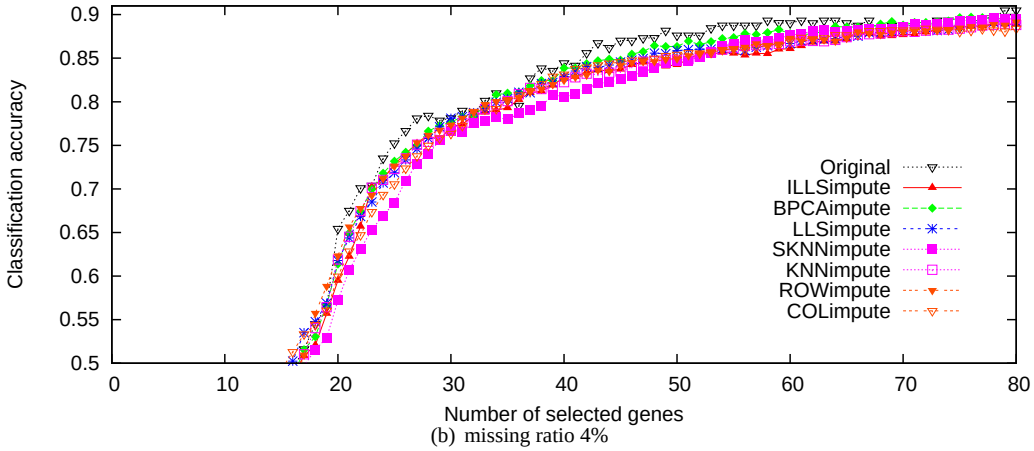


(b) missing ratio 2%

Figure 4.34: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

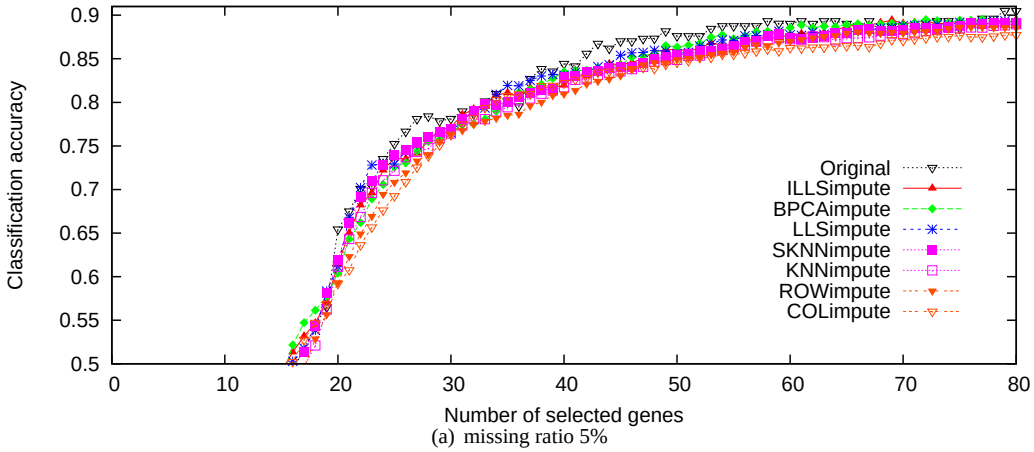


(a) missing ratio 3%

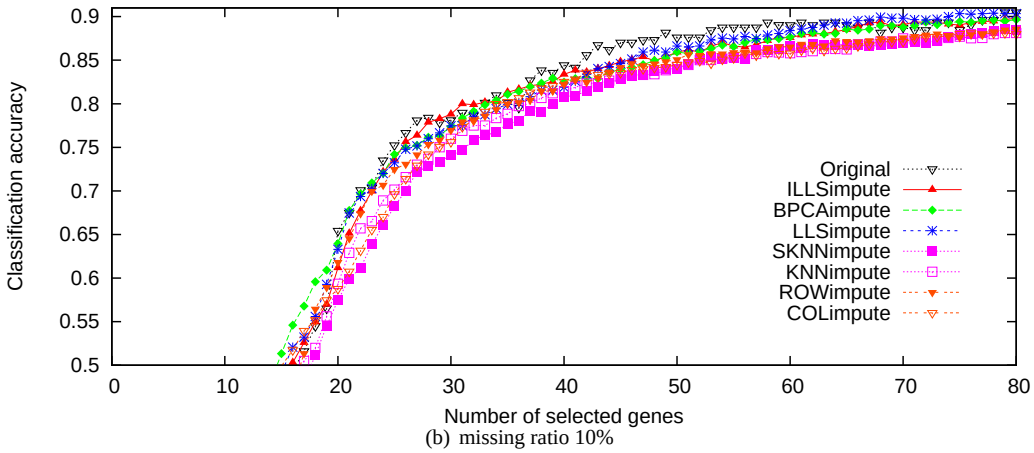


(b) missing ratio 4%

Figure 4.35: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

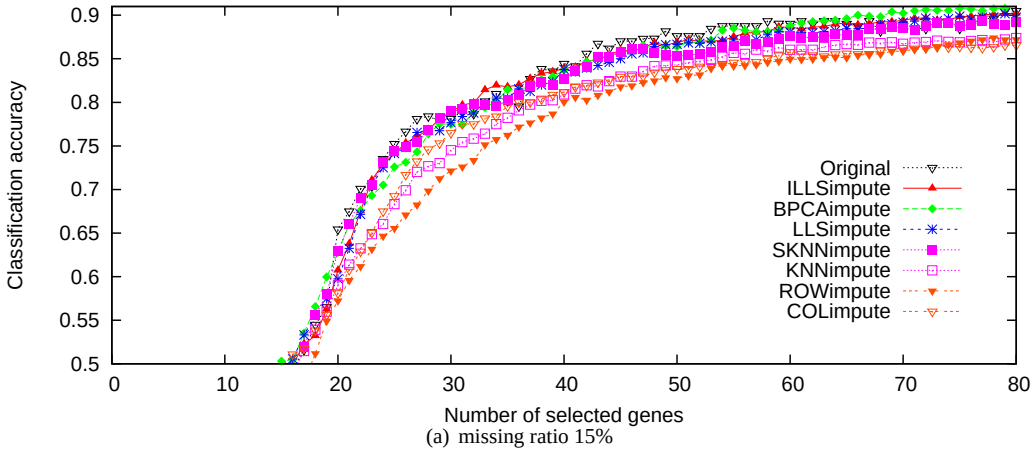


(a) missing ratio 5%

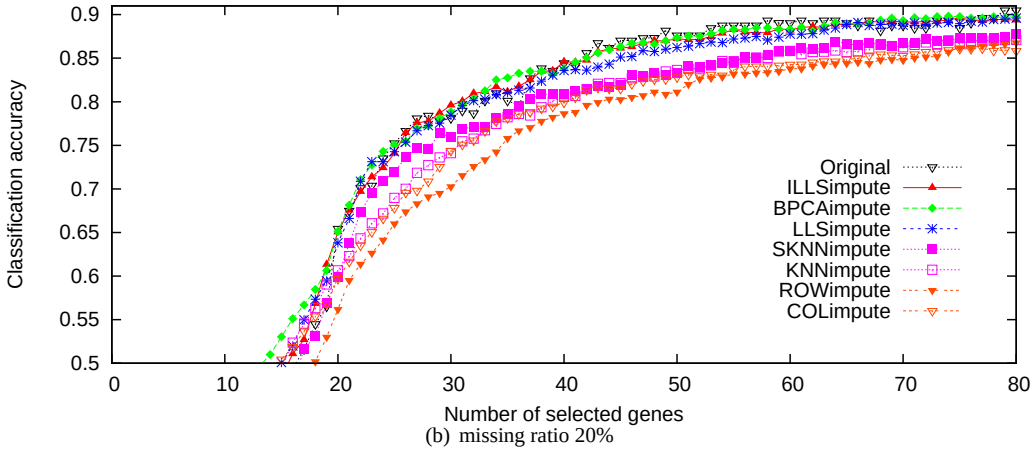


(b) missing ratio 10%

Figure 4.36: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

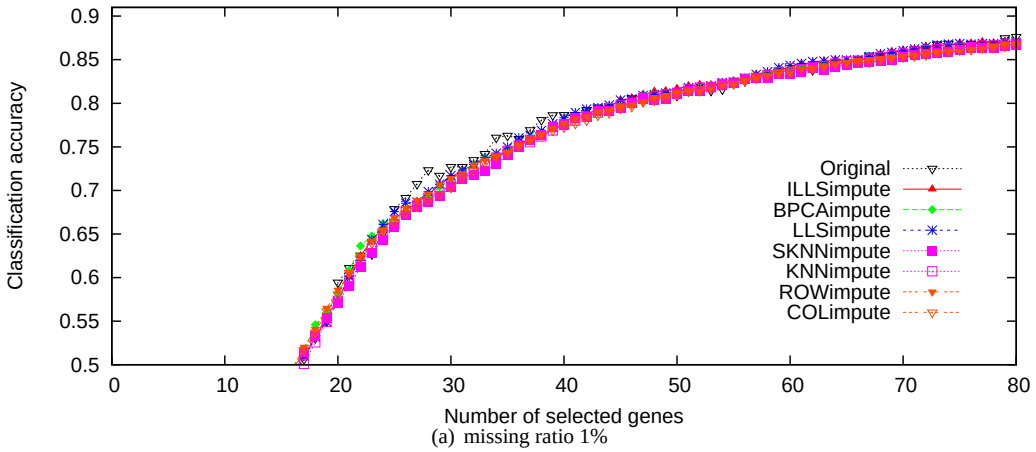


(a) missing ratio 15%

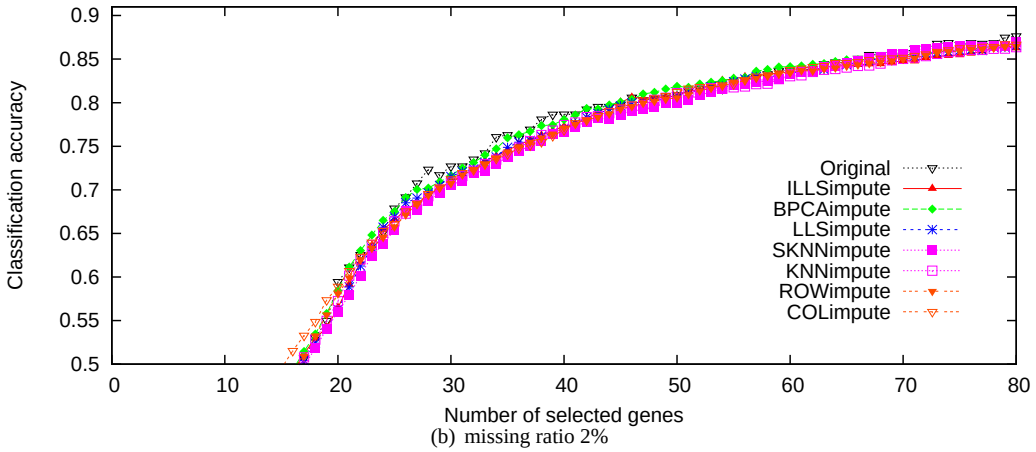


(b) missing ratio 20%

Figure 4.37: The 5-Fold classification accuracies of the KNN-classifier built on the genes selected by the CGS-Cho method, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this set of plots are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

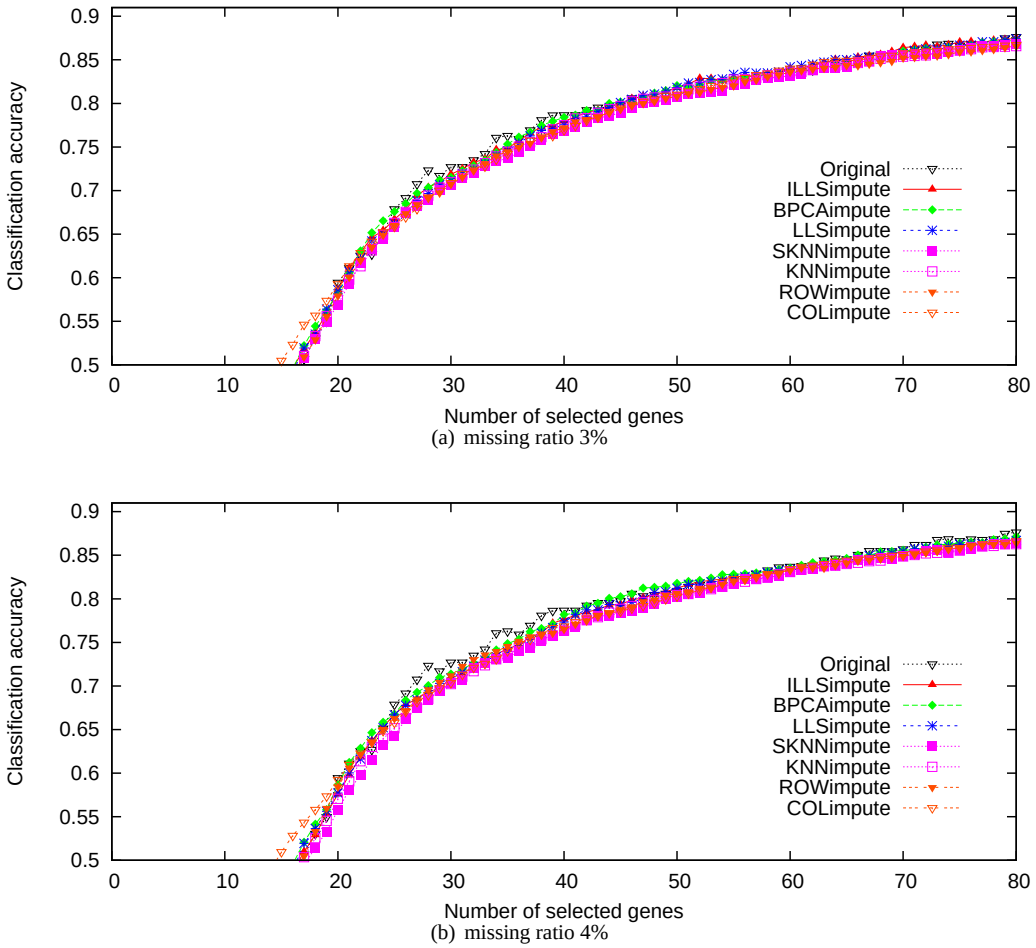


(a) missing ratio 1%



(b) missing ratio 2%

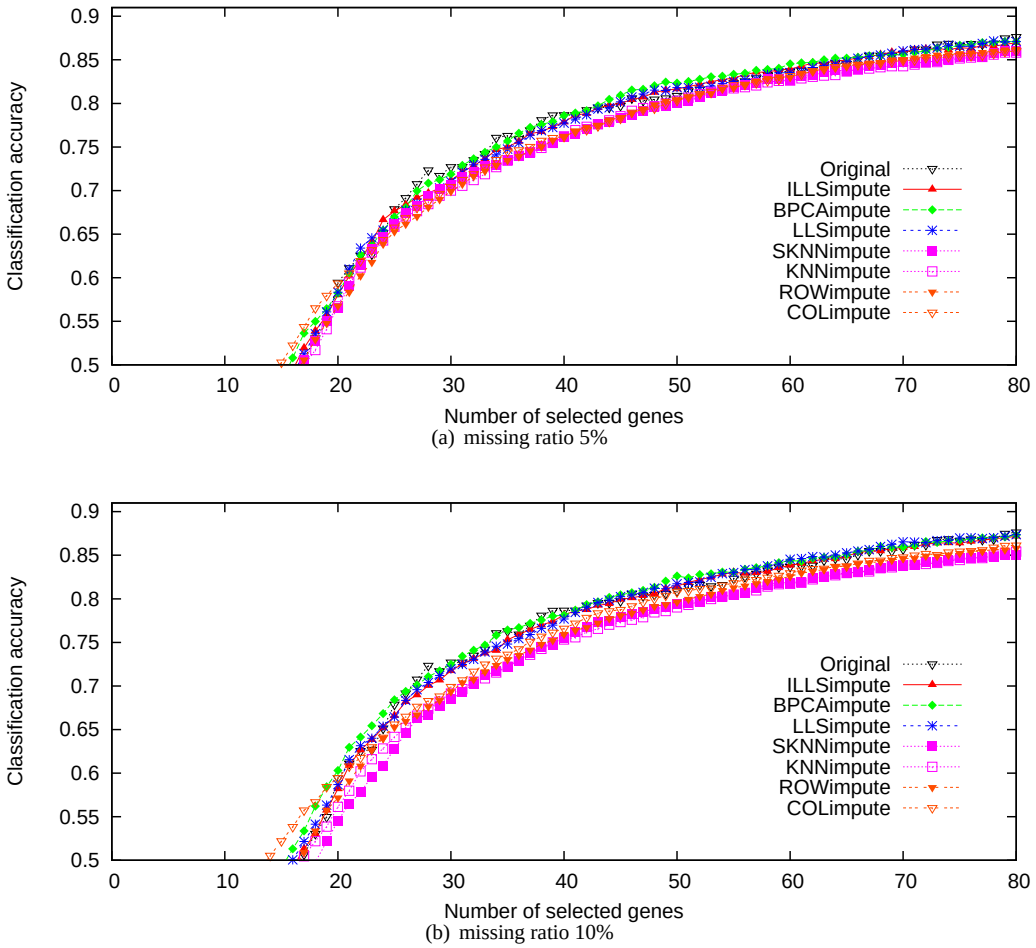
Figure 4.38: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 1% and 2%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.



(a) missing ratio 3%

(b) missing ratio 4%

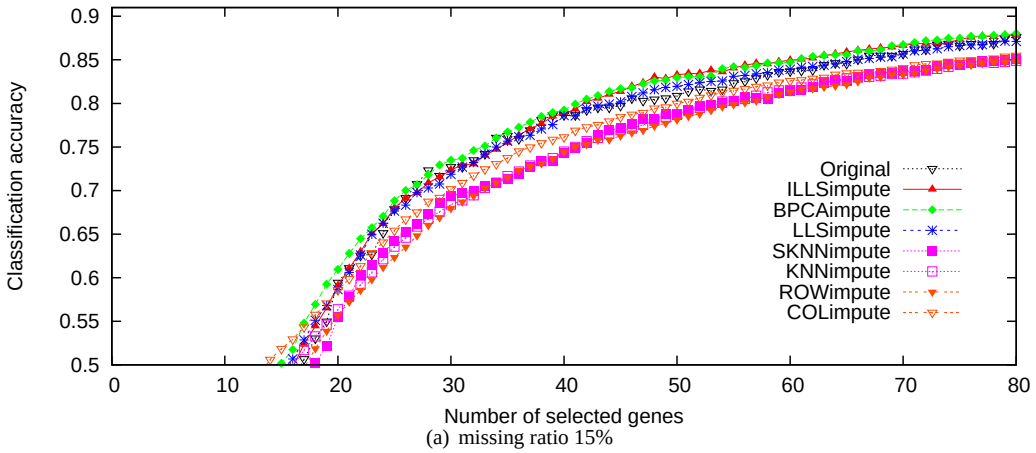
Figure 4.39: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 3% and 4%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.



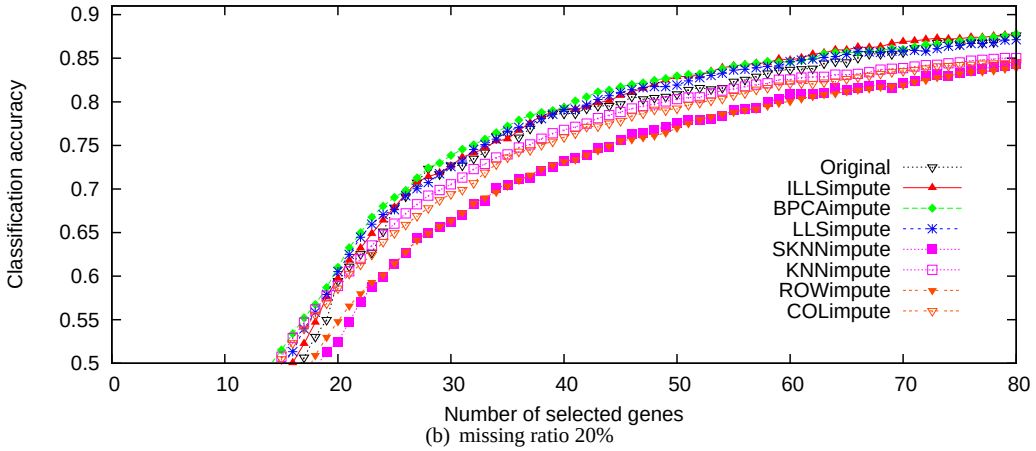
(a) missing ratio 5%

(b) missing ratio 10%

Figure 4.40: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 5% and 10%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.



(a) missing ratio 15%



(b) missing ratio 20%

Figure 4.41: The average 5-Fold classification accuracies of the KNN-classifier over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, on the Carcinomas dataset. The simulated datasets with missing values were imputed by each of ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute. The missing ratios on this plot are 15% and 20%. The Original plots the classification accuracies of the classifier on the original Carcinomas dataset, i.e. $r = 0\%$.

	Gene 40						
	ILLS	BPCA	LLS	SKNN	KNN	COL	ROW
ILLS	0.5000	0.9970	0.2042	0.0000	0.0000	0.0000	0.0000
BPCA	0.0030	0.5000	0.0003	0.0000	0.0000	0.0000	0.0000
LLS	0.7958	0.9997	0.5000	0.0000	0.0000	0.0000	0.0000
SKNN	1.0000	1.0000	1.0000	0.5000	1.0000	0.9971	0.0000
KNN	1.0000	1.0000	1.0000	0.0000	0.5000	0.0000	0.0000
COL	1.0000	1.0000	1.0000	0.0029	1.0000	0.5000	0.0000
ROW	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000
	Gene 60						
	ILLS	BPCA	LLS	SKNN	KNN	COL	ROW
ILLS	0.5000	0.9187	0.7420	0.0000	0.0000	0.0000	0.0000
BPCA	0.0813	0.5000	0.2223	0.0000	0.0000	0.0000	0.0000
LLS	0.2580	0.7777	0.5000	0.0000	0.0000	0.0000	0.0000
SKNN	1.0000	1.0000	1.0000	0.5000	1.0000	0.9988	0.0017
KNN	1.0000	1.0000	1.0000	0.0000	0.5000	0.0000	0.0000
COL	1.0000	1.0000	1.0000	0.0012	1.0000	0.5000	0.0000
ROW	1.0000	1.0000	1.0000	0.9983	1.0000	1.0000	0.5000
	Gene 80						
	ILLS	BPCA	LLS	SKNN	KNN	COL	ROW
ILLS	0.5000	0.9534	0.2499	0.0000	0.0000	0.0000	0.0000
BPCA	0.0466	0.5000	0.0108	0.0000	0.0000	0.0000	0.0000
LLS	0.7501	0.9892	0.5000	0.0000	0.0000	0.0000	0.0000
SKNN	1.0000	1.0000	1.0000	0.5000	1.0000	0.9971	0.0080
KNN	1.0000	1.0000	1.0000	0.0000	0.5000	0.0327	0.0000
COL	1.0000	1.0000	1.0000	0.0029	0.9673	0.5000	0.0000
ROW	1.0000	1.0000	1.0000	0.9920	1.0000	1.0000	0.5000

Table 4.2: P-values (significants) on Carcinomas dataset calculated for the right-tail hypothesis test of each pair of imputation methods (row to column) based on the classification accuracies on gene 40, 60, and 80 being selected and missing ratio 20%.

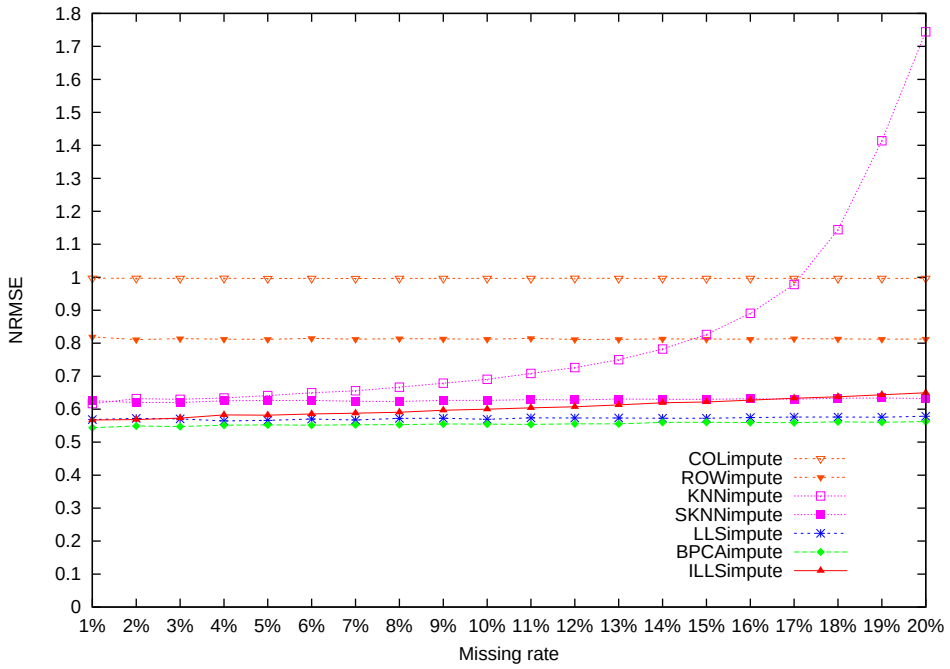


Figure 4.42: The plots of NRMSE values of the seven missing value imputation methods ILLSimpute, BPCAimpute, LLSimpute, SKNNimpute, KNNimpute, ROWimpute, and COLimpute on Carcinomas dataset.

Chapter 5

Conclusions and Discussion

Although adopting different gene selection methods in our study could finally lead to different sample classification accuracies, the collected average classification accuracies, which were taken over four gene selection methods, F-test, Cho, CGS-Ftest, and CGS-Cho, are convincing enough to conclude that ILLSimpute, BPCAimpute, and LLSimpute performed almost equally the best on the two cancer microarray datasets.

The NRMSE measurement is widely adopted in many missing value imputation studies, and we have identified its potential drawback in the microarray missing value study. That is, when using NRMSE for measuring the missing value imputation quality, it is presumed that the observed (original) gene expression dataset contains the gene expression values which all accurately reflect the actual hybridization intensities, and a missing value imputation method is considered to perform well if its imputed values are close to the observed expression values. Note that in practical microarray chips, the boundary between the accepted values and the treated-as-missing values could be vague, which means within the accepted expression values, there could still exist a considerable percentage of values which do not accurately measure the real gene hybridization intensities, although noises in them may not be significant to the level that they should be treated as missing values. When simulating the missing value matrix, the missing spots are randomly chosen, and some of these expression values may actually be inaccurate. Consequently, when a missing value imputation method has a good performance as measured by NRMSE, it could just reflect the fact that the imputed values are closer to the *inaccurate* observed values, rather than to the actual *true* ones. In most of our plots, we find that some classification accuracies computed based on the simulated dataset are higher than the classification accuracies computed based on the original observed dataset. This could be due to the fact that the imputed expression values are closer to the *true* expression values than the observed values. Having this consideration, we believe that the sample classification accuracy is another effective missing value imputation measurement, in addition to NRMSE, typically when the input microarray dataset cannot pass the quality control confidently. Nonetheless, when the expression values of an input microarray dataset are all of high confidence, that is, they do accurately measure the actual DNA hybridization intensities, NRMSE could be a better imputation

quality measurement, considering both its effectiveness and its computational complexity.

Observing from the classification accuracy plots that some imputed dataset-based classification accuracies are even higher than the original dataset-based classification accuracies, which suggests that the quality of the original dataset may not be as good as the imputed dataset, even the imputed dataset is obtained the random missing value simulation, we believe using some algorithm to correct (or smooth) the original missing value datasets could be a very attractive future work. For example, for each gene, along with the sample label information, how much effect a data spot could cause can be examined by first excluding and then imputing its expression value and calculating their respective F-test or Cho score differences, and by setting some threshold, potential missing values on the claimed original $true$ dataset can be identified.

Most of the missing value imputation methods discussed in this dissertation were originally proposed to be applied on microarray data, however, in fields other than microarray study or even the biology area, as long as the there are data correlations in the target dataset, the missing value imputation methods can be applied on them. And to measure the imputation quality, if the equivalent sample class membership information is available, the feature-selection-based (or full-feature-based) sample classification accuracies can be used as well as NRMSE. For example, if a travel plan sales company wants to improve their customer service quality, the current customer features (e.g., income, age, marriage status, interests, etc.) can be collected, which, of course, may contain some missing information. Therefore, the missing value imputation methods can be applied to predict those missing values, and with the known sample class memberships (in this case, the plans the customers chose), the imputation quality can be measured and the best imputation method can be selected. Once the imputation is selected, the missing values for the current customers or for new-coming customers can be determined based on the selected imputation method which can help the company to customize more suitable plans for the customers and that the company profit can be increased as well. This idea can be applied on other research or industrial areas where the similar application scenarios exist. Moreover, if the sample class memberships are highly reliable, the sample classification accuracy could be a more appropriate measurement than NRMSE.

Through this study, we confirm that NRMSE is not the best method for measuring missing value imputation quality, at least in the field of gene expression microarray data. The sample classification accuracy, as we proposed, is a better, more stable, measurement as demonstrated in our extensive simulation experiments. And because the sample classification accuracy method is application oriented, when contradicting to the NRMSE result, it should be the more reliable measurement.

Acknowledgements

I would like to thank my supervisor, Dr. Guohui Lin, for his suggestions, encouragement and support which greatly helped me to finish this thesis. I also would like to thank iCORE, NSERC, and CFI for their funding support.

Bibliography

[1] Affymetrix. Probe design and selection. `http://www.affymetrix.com/technology/design/index.affx`.

[2] 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.Jr. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown, and L.M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[3] 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.

[4] 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. *Proceddings of National Academy of Sciences of the United States of America*, 98:13790–13795, 2001.

[5] 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.

[6] T.H. Bø and I. Jonassen. New feature subset selection procedures for classification of expression profiles. *Genome Biology*, 3:research0017.1–0017.11, 2002.

[7] Z. Cai, M. Heydari, and G. Lin. Iterated local least squares microarray missing value imputation. *Journal of Bioinformatics and Computational Biology*, 4(5):935–57, 2006.

[8] Z. Cai, L. Xu, Y. Shi, M. Salavatipour, R. Goebel, and G. Lin. Using gene clustering to identify discriminatory genes with higher classification accuracy. *IEEE The 6th Symposium on Bioinformatics and Bioengineering (IEEE BIBE 2006)*, pages 135–242, 2006.

[9] J. Choa, D. Leea, J.H. Parkb, and I. Leea. New gene selection method for classification of cancer subtypes considering within-class variation. *European Biochemical Societies (FEBS Letters)*, 551:3–7, 2003.

[10] 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.

[11] X. Gan, A.W.C Liew, and H. Yan. Microarray missing data imputation based on a set theoretic framework and biological knowledge. *Nucleic Acids Research*, 34:5:1608–1619, 2006.

[12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[13] 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.

[14] 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.

[15] 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.

[16] M.L.T. Lee. Analysis of microarray gene expression data. *Kluwer Academic Publishers*, 2004.

[17] C.S.F. Liu L.M. Fu. Multi-class cancer subtype classification based on gene expression signatures with reliability analysis. *FEBS Letters*, 561:186–190, 2004.

[18] 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.

[19] 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.

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

[21] I. Scheel, M. Aldrin, R. Sørum I.K. Glad and, 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.

[22] 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.

[23] A.I Su, J.B Welsh, L.M. Sapinoso, S.G Kern, P. Dimitrov P, H. Lapp, P.G. Schultz, S.M. Powell, C.A. Moskaluk, H.F.Jr. Frierson, and G.M. Hampton. Molecular classification of human carcinomas by use of gene expression signatures. *Cancer Research*, 61(20):7388–7393, 2001.

[24] O. 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.

[25] X. Wang, A. Li, Z. Jiang, and H. Feng. Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme. *BMC Bioinformatics*, 7:32.

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

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