Available online at www.sciencedirect.com
ScienceDirect
Biomolecular Engineering 24 (2007) 273–282
Biomolecular Engineering
www.elsevier.com/locate/geneanabioeng

Improving cluster-based missing value estimation of DNA microarray data

Lígia P. Brás, José C. Menezes *

Centre for Chemical & Biological Engineering, Department of Chemical and Biological Engineering, IST, Technical University of Lisbon, Av. Rovisco Pais, P-1049-001 Lisbon, Portugal

Received 27 October 2006; received in revised form 21 February 2007; accepted 12 April 2007

Abstract

We present a modification of the weighted *K*-nearest neighbours imputation method (KNNimpute) for missing values (MVs) estimation in microarray data based on the reuse of estimated data. The method was called iterative KNN imputation (IKNNimpute) as the estimation is performed iteratively using the recently estimated values.

The estimation efficiency of IKNNimpute was assessed under different conditions (data type, fraction and structure of missing data) by the normalized root mean squared error (NRMSE) and the correlation coefficients between estimated and true values, and compared with that of other cluster-based estimation methods (KNNimpute and sequential KNN). We further investigated the influence of imputation on the detection of differentially expressed genes using SAM by examining the differentially expressed genes that are lost after MV estimation.

The performance measures give consistent results, indicating that the iterative procedure of IKNNimpute can enhance the prediction ability of cluster-based methods in the presence of high missing rates, in non-time series experiments and in data sets comprising both time series and non-time series data, because the information of the genes having MVs is used more efficiently and the iterative procedure allows refining the MV estimates. More importantly, IKNN has a smaller detrimental effect on the detection of differentially expressed genes.

Keywords: Missing value estimation; *K*-nearest neighbours; Gene expression data; DNA microarray data

1. Introduction

DNA microarrays are a high-throughput technology that allows for the simultaneously monitoring of the mRNA levels of thousands of genes in particular cells or tissues, giving a global view of gene expression (Lockhart and Winzeler, 2000; Schena et al., 1995; Schulze and Downward, 2001).

The data generated in a set of microarray experiments are usually gathered in a matrix with genes in rows and experimental conditions in columns. Frequently, these matrices contain missing values (MVs). This is due to the occurrence of imperfections during the microarray experiment (e.g. insufficient resolution, spotting problems, deposition of dust or scratches on the slide, hybridization failures) that create suspect values, which are usually thrown away and set as missing (Alizadeh et al., 2000). The in situ synthesized Affymetrix GeneChips and the spotted cDNA (or oligonucleotide) microarrays are the two most commonly used types of microarray technology. The redundancy in design used in a GeneChip (i.e. a gene is represented by a set of approximately 20 probe pairs) prevents the existence of MVs. This is not the case for spotted cDNA microarrays, where usually each spot is assigned to a unique gene, and the use of double to quadruple spots for a gene is currently an exception. So, the loss at a spot usually leads to the loss of information for a gene, and thus to a MV in the gene expression data matrix. Therefore, in this work we consider the estimation of MVs in gene expression data obtained from spotted cDNA microarrays.

In some microarray data sets, the proportion of MVs is significant. For example, some authors reported that the percentage of gene profiles with at least one MV can be higher than 85% (de Brevern et al., 2004). The presence of missing gene expression values constitutes a problem for downstream data analyses, since many of the methods employed (e.g. classification and model-based clustering techniques) require

* Corresponding author. Tel.: +351 218 417 347; fax: +351 218 419 197.
E-mail addresses: ligia.bras@ist.utl.pt (L.P. Brás), bsel@ist.utl.pt (J.C. Menezes).

1389-0344/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.bioeng.2007.04.003

complete matrices. Due to economic reasons or biological sample availability, repeating the microarray experiments in order to obtain a complete gene expression matrix is usually unfeasible, so other alternatives have to be considered. The simple approaches usually applied to handle missing gene expression entries include removing the genes with MVs before the analysis (case deletion), or replacing the MVs of a gene with the average of the observed values over that gene (mean substitution; Schafer and Graham, 2002). Another common approach is to replace missing $\log_2$ transformed gene expression ratios by zeros (Alizadeh et al., 2000). These approaches have disadvantages: case deletion procedures may bias the results if the remaining cases are unrepresentative of the entire sample (Little and Rubin, 1987), while both mean and zero substitutions distort relationships among variables and artificially reduce the variance of the variable in question (Little and Rubin, 1987; Schafer and Graham, 2002), since the same value is used to replace missing entries in a given gene.

To overcome these drawbacks, Troyanskaya et al. (2001) proposed a method called weighted *K*-nearest neighbour imputation (KNNimpute) that reconstructs the MVs using a weighted average of *K* most similar genes. Overall, this estimation method is more robust than others, such as replacement by zero, row average or singular value decomposition, to the fraction of missing elements and to the type of data for which estimation is executed, performing better in non-time series data or noisy data (Troyanskaya et al., 2001). As an improvement of KNN imputation, Kim et al. (2004) proposed a sequential KNN imputation method (SKNNimpute) that uses the estimated values sequentially for the later nearest neighbour calculation and estimation.

In a recent work, de Brevern et al. (2004) studied the stability of gene clusters of microarray data including MVs or not, specified by diverse hierarchical clustering algorithms, showing that the MVs (even at a low rate) have important effects on the gene clusters' stability. Thus, the presence of MVs in the data matrix should not be neglected, and MV estimation should be regarded as a pre-processing step essential to obtain proper results from microarray data analyses.

Although other methods have been proposed for estimating gene expression missing data, such as regression-based methods (Bø et al., 2004; Kim et al., 2005; Nguyen et al., 2004; Brás and Menezes, 2006) and Bayesian approaches (Oba et al., 2003), in this work we focus on the cluster-based methods, since these are widely used for the replacement of MVs in microarray data. For example, KNNimpute is the only imputation method available in significance analysis of microarrays (SAM; Tusher et al., 2001), prediction analysis for microarrays (PAM; Tibshirani et al., 2002) and microarray analysis of variance (MAANOVA; Kerr et al., 2000).

We propose an iterative procedure for the prediction of gene expression MVs called iterative KNN imputation (IKNNimpute), and compare its performance with that of other clustering-based imputation methods (KNNimpute and SKNNimpute) for various rates of MVs and type of missing structure using publicly available microarray data sets.

The methods are evaluated by comparing their estimates for the artificial missing entries with the true values, using measures such as normalized root mean squared errors, correlation coefficients and bias. Though such approach gives important measures of performance, a more fundamental and functional question that should further be addressed is the effect of the methods' estimates on the final output of different analysis methods, such as clustering algorithms or statistical algorithms for the differential analysis of gene expression. In the literature, such evaluations are lacking, and only a few cases can be found (for example, see de Brevern et al., 2004; Ouyang et al., 2004; Scheel et al., 2005; Jörnsten et al., 2005). In our study, the impact of the imputation methods' estimates on significance analysis for differential expression is also performed by comparing the lists of differentially expressed genes obtained using the statistical method known as SAM (Tusher et al., 2001). We opted to focus on the effects of imputation on differential expression, since, although cluster analysis of microarray data is capable of discovering coherent patterns of gene expression, it gives little information about statistical significance, i.e., about whether changes in gene expression are experimentally significant.

2. Materials and methods

2.1. Notation

Throughout this paper, microarray data are represented by matrices with rows corresponding to genes and columns to experimental conditions. In particular, $\boldsymbol{G}$ represents the original data matrix (with real MVs), while $\boldsymbol{X}$ is a gene expression matrix with p genes and n experiments (with $p \gg n$) that may contain missing entries. The ith row of $\boldsymbol{X}$ represents the expression profile of the ith gene in the n experiments, whereas x_{ij} denotes the expression level of gene i in sample j.

Using the notation of Nguyen et al. (2004), a gene with MVs is called *target* gene, and the genes with available information for estimating its missing entries constitute the set of *candidate* genes.

We also make use of the missing indicator matrix, $\boldsymbol{R}$, defined by Rubin (1976) to track the missing and non-missing entries of $\boldsymbol{X}$. If the expression value x_{ij} is available, the ijth element of $\boldsymbol{R}$, r_{ij}, is equal to 1, otherwise it is zero.

2.2. Weighted KNN imputation and SKNN imputation

In cluster-based estimation, MVs are estimated by combining the expression levels of *K*-nearest genes chosen based on a given similarity measure. Thus, KNN predictions are based on the intuitive assumption that objects close in distance are potentially similar. Both the measure to use for computing similarities between genes and the number of nearest neighbours (K) must be determined.

For a given target gene $\boldsymbol{x}_i$, KNNimpute (Troyanskaya et al., 2001) calculates a weighted Euclidean distance d_{ik} between the target gene i and each candidate gene k using the expression:

$$d_{ik} = \sqrt{\frac{\sum_{j=1}^{n} r_{ij} r_{kj} (x_{kj} - x_{ij})^2}{\sum_{j=1}^{n} r_{ij} r_{kj}}} \tag{1}$$

where r_{ij} is the element in the ith row and jth column of the missing indicator matrix $\boldsymbol{R}$. The missing entry j of target gene i is then estimated by the weighted average of the expression values of the K most similar genes in experiment j:

$$\hat{y}_{ij} = \sum_{k=1}^{K} w_{ik} x_{kj} \tag{2}$$

where w_{ik} is the weight for the kth neighbour gene of target gene i normalized by the sum of the inverse weighted Euclidean distance for all K neighbours (i.e. the contribution of each neighbour gene is weighted by the similarity of its expression to that of the target gene):

$$w_{ik} = \frac{1/d_{ik}}{\sum_{k=1}^{K} 1/d_{ik}} \quad (3)$$

SKNNimpute (Kim et al., 2004) was proposed as an improvement to KNNimpute, differing from the latter method in two main points: (a) MVs are estimated sequentially starting with the gene having the smallest missing rate, and (b) SKNNimpute uses the estimated values for estimating the MVs of the remaining genes. In SKNNimpute, $\boldsymbol{X}$ is split into two sets by considering the genes with no MVs ($\boldsymbol{X}^{\text{complete}}$) and the genes comprising MVs ($\boldsymbol{X}^{\text{incomplete}}$). The former matrix is used as the candidate set, while the latter contains the target genes to be estimated following the order of their missing rate. Applying the KNN principle, the target gene's missing entries are filled according to Eq. (2). However, once the estimation of a given target gene is completed, the candidate set is updated with that gene, so that it can be used for the next estimation round.

In KNNimpute, the set of candidate genes is constructed for each missing position of a given target gene. Thus, in each estimation, the candidate matrix may contain MVs. Therefore, the pairs of vectors target gene/candidate gene may have different lengths. Ignoring this fact would be assuming that expression levels are equal in both vectors, so that vectors with more MVs would present smaller distances. Therefore, as presented in Eq. (1), only the jointly available positions between both target and candidate genes are used to compute the Euclidean distance, and the number of such positions is used as weight (weighted Euclidean distance). In SKNNimpute, all MVs in a given target gene are estimated simultaneously using the selected neighbour genes, since the latter genes were taken from $\boldsymbol{X}^{\text{complete}}$. Moreover, a simple Euclidean distance can be used. Consequently, SKNNimpute offers an advantage over KNNimpute in terms of speed.

2.3. IKNN imputation

As referred above, one of the main differences between KNNimpute and SKNNimpute is due to the fact that in SKNNimpute the set of candidate genes is continuously updated after completing each target gene, making it possible to use former target genes as candidate genes. Reformulating the concept of reusing the estimated data in the estimation process, we develop a new method that we called iterative KNN imputation (IKNNimpute). IKNNimpute is based on an iterative procedure that involves the following steps:

Step 1. Initialisation: replace all the MVs in $\boldsymbol{X}$ by the estimates given by row (gene) averages, obtaining a complete matrix $\boldsymbol{X}^{\text{complete }(0)}$.

Step 2. hth estimation cycle ($h = 1,\ldots$):

(1) For each target gene i in $\mathbf{X}$:
   a. Using $\boldsymbol{X}^{\text{complete }(h-1)}$, construct the matrix of candidate genes that comprises all genes, except the one that is currently being estimated (i.e. the target gene).
   b. Compute the Euclidean distance between the target gene and each candidate gene, and select the K nearest genes.
   c. Impute the MVs in target gene i simultaneously by using a weighted average of the expression levels of the K-nearest genes—Eq. (2).
(2) After imputing all target genes, $\boldsymbol{X}^{\text{complete }(h)}$ is obtained.
(3) Determine the sum of squared differences between the estimated positions of the complete matrices $\boldsymbol{X}^{\text{complete }(h-1)}$ and $\boldsymbol{X}^{\text{complete }(h)}$ obtained from the two last iterations:

$$\delta^{(h)} = \sum_{j=1}^{N} \left(\hat{y}_j^{(h-1)} - \hat{y}_j^{(h)}\right)^2,$$

Step 3. If $\delta^{(h)} < \tau$, stop. Otherwise, return to step (2) and iterate until the convergence criterion τ is reached.

Herein, we considered $\tau = 10^{-3}$. In general, the convergence criterion was reached in two iterations ($h = 2$).

IKNNimpute differs from SKNNimpute in the way of constructing the set of candidate genes and in the way of reusing the estimated data. Specifically, the initial step of gene average substitution performed in IKNNimpute provides the possibility of using the maximum number of genes as candidates for estimating the MVs of a given target gene. Furthermore, the iterative procedure allows a refinement of the predictions. As in SKNNimpute, the missing positions of a target gene can be estimated at once, which is computationally more efficient compared to KNNimpute.

2.4. Data

In this study, we used four publicly available data sets. Two of the data sets (data sets TS1 and TS2) come from a study of the cell cycle regulated genes in *Saccharomyces cerevisiae* (Spellman et al., 1998), and consist of time series cDNA microarray data. The data set called TS1 contains data from a *cdc15*- and *cdc28*-based synchronisation, while data set TS2 only comprises the *cdc28*-based synchronisation data. TS2 has a dimension ratio n/p five times smaller than that of TS1, but similar to that of the third and fourth data sets. The third data set belongs to a study of gene expression regulated by the calcineurin/Crz1p-signalling pathway in *S. cerevisiae* (Yoshimoto et al., 2002). This data set is denoted by MIX, since it can be classified as a mixed experiment, comprising both time course and non-time course data. The fourth data set comes from a study of human cancer cell lines (Ross et al., 2000), and is termed by NTS, since it corresponds to non-time series data. All data sets were downloaded from the supplementary Internet pages accompanying the papers, and consist of cDNA microarray experiments. Table 1 presents the dimensions of the data matrices before (original data set) and after removing all genes with MVs (complete data set). Prior to the analysis, data were logarithmically (base 2) transformed (except for the cases where data sets were already downloaded in $\log_2$ scale).

2.5. Missing data set-up

To evaluate the methods' accuracy, we introduced artificial missing entries to a complete (i.e. without MVs) expression matrix, $\boldsymbol{X}^{\text{complete}}$, constructed from the real data set ($\boldsymbol{G}$) by discarding the missing elements. Two different procedures were considered (A and B).

In procedure A, the test set $\boldsymbol{X}$ was constructed by randomly removing (marking as missing) a specific percentage of the entries (1, 5 and 10%) of $\boldsymbol{X}^{\text{complete}}$. These percentages were chosen based on the values of missing rate commonly encountered in real experimental microarray data sets (see below).

Given that the probes are arrayed at random in the chips, one can expect that the missing signals caused by effects such as irregularities in the spot production, hybridization failure, dust on the chip, spatial noise, etc., will have a random distribution. However, in some cases where the signal is too low, the image processing software used for spotted cDNA microarrays flags out signals that cannot be distinguished from the background, or that have too irregular shape. Thus, in such cases, missing entries are not introduced at random, but instead the missing pattern depends on the signal intensity. In general, we should then expect that a mixture of missing at random and not at random will be present in a given microarray data set. At this view, procedure B intends to reproduce realistic missing data patterns, i.e. an abnormally high frequency of MVs in some arrays (columns) of a real microarray data matrix. Therefore, in procedure B, we assigned MVs to the elements in the p rows of the complete matrix by randomly sampling p rows (genes) of $\boldsymbol{G}$, and using their missing positions. This led to a similar missing structure for the test data set $\boldsymbol{X}$ as that of the original set $\boldsymbol{G}$. Table 1 presents the structure of missing entries in $\boldsymbol{G}$ for the different data sets analyzed in this work, showing that the total missing rate in the original data sets was below 10%. Similar total missing rates have been reported in other microarray data sets. For example, de Brevern et al. (2004) examined the content of MVs in eight series of microarray experiments, reporting that the percentage of MVs varied from 0.8 to 10.6%. Table 1 shows that in all data sets, only a small number of genes (inferior to 1.5%) have more than 50% of their entries missing, while there is no array with more than 50% entries missing. Thus, according to Table 1, the total percentage of missing elements in the test data sets generated by procedure B is approximately 8%, 6%, 4% and 4%, respectively, for experiments TS1, TS2, MIX and NTS.

Table 1
Dimension and missing pattern of the data sets: dimension of the data matrices before (original data set ***G***) and after (complete data set) removing the missing elements; ratio between the number of experiments and genes in the complete data sets (*n*/*p*); pattern of missing entries in the original data matrices (overall missing rate and distribution of MVs among genes and among arrays)

	TS1	TS2	MIX	NTS
Dimension ($p \times n$)				
Original data set (***G***)	6178 × 41	6178 × 17	6166 × 24	9712 × 64
Complete data set	869 × 41	1383 × 17	4380 × 24	6115 × 64
n/*p* (complete data set)	0.05	0.01	0.006	0.01
Total missing rate in ***G*** (%)	8.3	6.1	3.8	3.9
% of genes in ***G*** with:				
<5% missing entries	82.5	22.4	84.8	82.2
5–10% missing entries	9.3	66.8	4.4	6.2
10–20% missing entries	5.0	9.1	4.3	5.9
20–50% missing entries	2.5	0.9	5.2	4.6
≥50% missing entries	0.7	0.8	1.3	1.1
% of arrays in **G** with:				
<5% missing entries	87.9	64.8	75.0	73.5
5–10% missing entries	12.1	23.5	25.0	20.3
10–20% missing entries	0.0	0.0	0.0	4.7
20–50% missing entries	0.0	11.7	0.0	1.5
≥50% missing entries	0.0	0.0	0.0	0.0

To obtain results unbiased with regard to the portion of the data that is missing, we run five independent rounds of procedures A and B.

We use the following notation (here exemplified for dataset NTS) to identify the type and rate of MVs in each data matrix: $NTS^{1\%}$ represents the matrix obtained after randomly assigning as missing 1% of the entries of NTS using procedure A; while NTS^{uneq} represents the NTS matrix with unequally distributed missing entries introduced according to procedure B.

KNNimpute and SKNNimpute were run on R (version 2.0.1., 2004), a free software environment for statistical computing and graphics (Gentleman and Ihaka, 1996), while IKNNimpute was implemented in MATLAB (version 6.5.0. Natick, Massachusetts, USA; The MathWorks Inc., 2002). The R code for KNNimpute is given in the package *impute* that can be downloaded from the Bioconductor project (Gentleman et al., 2004). The code for SKNNimpute was gently given by K. Kim. The code of IKNN method is available upon request for both R and MATLAB.

2.6. Parameter sets

In cluster-based methods, the number of nearest neighbours, *K*, must be selected. Troyanskaya et al. (2001) addressed this question in KNNimpute, reporting the best results for *K* in the range 10–20. Therefore, we decided to perform multiple estimation tests using $K = 5$, 10, 15 and 20. Moreover, we implemented a procedure for selecting *K* automatically in IKNNimpute, making it a parameter-free method. This procedure was implemented as follows: at least one position among the non-missing elements of the genes with MVs was set as missing, and estimated using different *K* values. The optimum value for the parameter was chosen as the one originating the smallest prediction error. The obtained value for *K* was then employed for estimating the MVs by the IKNNimpute algorithm.

2.7. Evaluation of the methods

For every data set, each estimation method was applied to recover the introduced MVs, and the accuracy of the method was evaluated by calculating the error between actual (y_j) and estimated values ($\hat{y}_j$) using the normalized root mean squared error (NRMSE):

$$\mathrm{NRMSE} = \frac{1}{\sigma_y}\sqrt{\frac{\sum_{j=1}^{N}(y_j - \hat{y}_j)^2}{N}} \quad (4)$$

where σ_y is the standard deviation for the *N* true values that correspond to all the missing entries in the test matrix.

Moreover, in order to assess the capability of the different estimation methods in preserving the data structure of each experiment, we computed the squared Pearson correlation coefficients (R^2) between true and estimated values for each column (experiment) of the data set.

The bias on the methods, i.e. a consistent under- or overestimation of the true values was also tested using the Wilcoxon signed rank test (Siegel and Castellan, 1988). Considering the residuals $\varepsilon_j = y_j - \hat{y}_j$ of a given estimation method, we tested the null hypothesis (H_0) that states that negative and positive residuals are equally likely.

To compare the performance of two different estimation methods (or the same estimation method run using different *K*-values) one can compare their mean squared errors of prediction (MSEP), which is equivalent to performing a test comparing the variances of two groups of samples. Herein, we used the Levene's test (Levene, 1960), whereby the data values (residuals) were transformed and subject to an analysis of variance to produce the usual *F*-statistic for a test of whether the means vary significantly between the samples. As a data transformation, we opted to apply the absolute deviation from the sample median (instead of the sample mean), since it provides good robustness against many types of non-normal data, retaining good power (Manly, 1998). The null hypothesis states that the variance (or MSEP, in our case) is equal across both methods, while the alternative hypothesis states that the variances are different between the two estimates.

In both tests (Wilcoxon signed rank test and Levene's test), we considered a significance level of 5%, so H_0 was rejected if the obtained *p*-value was inferior or equal to 0.05.

We further assessed the performance of the methods by comparing the list of differentially expressed genes based on the imputed matrix with the same list based on the true full data set. Specifically, we counted the number of genes in the latter list that were lost when analyzing the imputed data set. While there are several methods for detecting differentially expressed genes, herein we considered a common approach called SAM. SAM is a statistical method developed by Tusher et al. (2001). SAM gives a score to each gene on the basis of change in gene expression relative to the standard deviation across experimental conditions. These scores are calculated again after permuting the data set several times. The original and permuted-based values are approximately equal for the majority of the genes, while for a few genes, the difference between the two scores exceeds a given threshold; hence, these genes are supposed potentially significant and called differentially expressed. The percentage of genes identified by chance (the false

discovery rate, FDR) is estimated on the basis of permuted data sets. SAM is freely available as an R package called *samr*, available at http://www-stat.stanford.edu/~tibs/SAM. In our study, for detecting differentially expressed genes, we used a median false discovery rate of 1% in SAM.

3. Results and discussion

The NRMSE was used to assess the accuracy of each method under different conditions, i.e., type of data (time series, mixed and non-time series), proportion of MVs (1, 5 and 10%), and missing structure (equally and unequally distributed MVs). Results are presented in Fig. 1.

3.1. Model parameter

Many authors refer that the estimation ability of KNNimpute depends on the number of nearest neighbour genes, *K*. This parameter is dependent on the data type and missing rate, but has no theoretical way, however, to be appropriately determined. In this context, we decided to evaluate if choosing the correct value for *K* was in fact such a relevant issue in cluster-based estimation methods and if the use of an automatic procedure to select *K* was worthwhile. Therefore, we studied the influence of the value of *K* on prediction ability by performing Levene's tests on the transformed residuals obtained using different *K* values.

For all cluster-based methods, results from Levene's test show that for TS experiments (data sets TS1 and TS2), the range $K = 5$–20 gives statistically equivalent prediction errors. In data sets TS1 and MIX, this optimum range narrows for higher missing rates in favour of higher *K* values ($K = 10$–20). For the non-time series data, in general, there is evidence of a better estimation performance when a small value for *K* is utilized ($K = 5$). Thus, in the presence of a weaker data similarity structure (NTS data), incorporating farthest neighbours reduces the prediction accuracy, since the information they bring is irrelevant comparatively to the noise they introduce in the MVs estimation process.

We implemented a procedure to automatically estimate the optimum number of nearest neighbouring genes (*K*) within

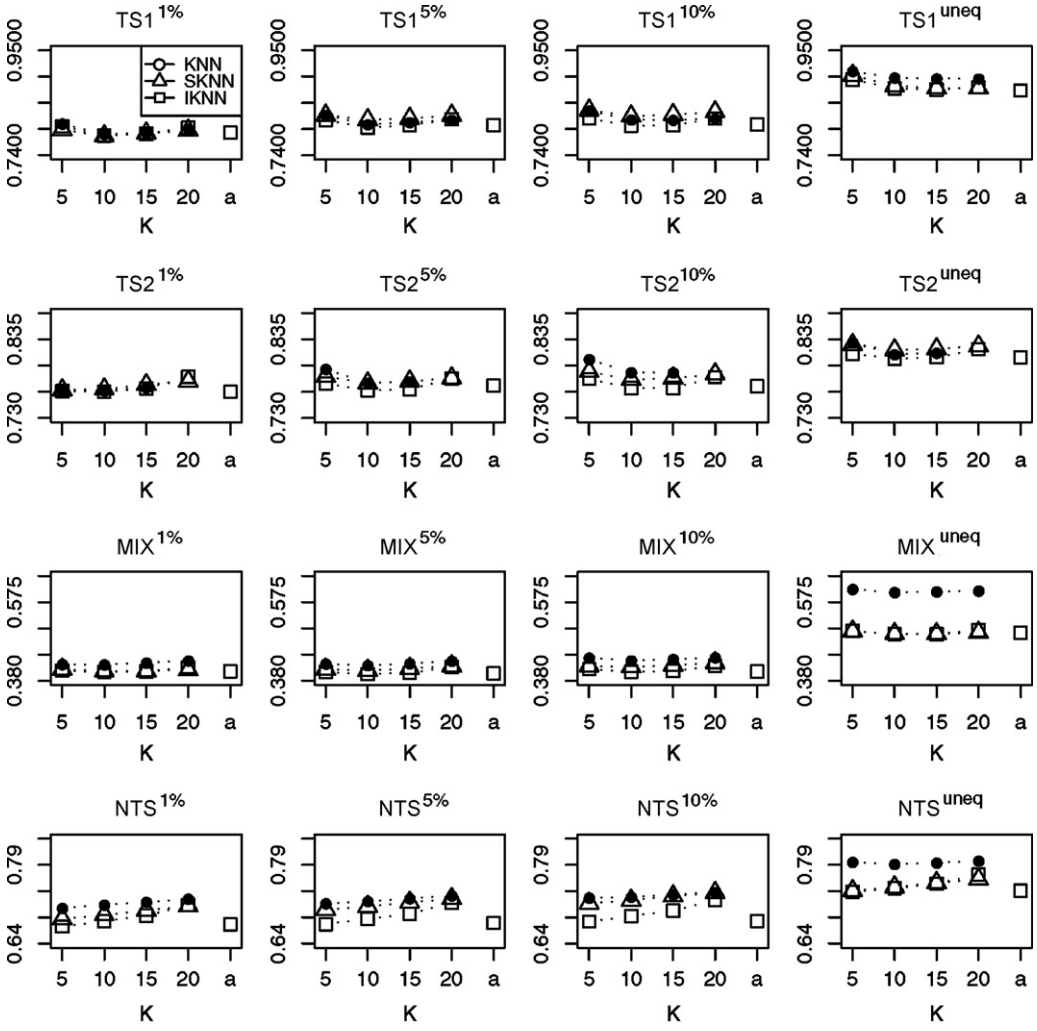

Fig. 1. Normalized root mean squared error (NRMSE) calculated based on the missing value estimates obtained using different parameters (*x*-axis) in the cluster-based methods for the different data sets. "$K = a$" corresponds to the case where *K* was automatically selected inside IKNN method.

IKNNimpute. For all the types of experiments, we found that the value automatically estimated for K (data not shown) belongs to the previously reported optimum range determined using a fixed K value in IKNNimpute. So, the increase in computation time required by a procedure such as the one herein implemented is dispensable, and K can just be set to any value in the range $K = 5$–20 (or to be more conservative, in the range $K = 10$–15) for TS and MIX data, while K should be set equal to 5 when dealing with NTS data.

3.2. Comparison between cluster-based methods

Considering the model parameter value that originated a minimal NRMSE for each type of microarray data, missing rate and pattern, the prediction performance of KNNimpute, SKNNimpute and IKNNimpute were compared by applying the Levene's test on the transformed residuals. The obtained p-values are shown in Table 2, and indicate that SKNNimpute and IKNNimpute have statistically smaller MSEPs than KNNimpute when applied on MIX, NTS and $TS1^{uneq}$ data ($p < 0.04$). The three cluster-based methods display similar estimation ability in the other TS test data sets ($7.86 \times 10^{-2} < p < 9.87 \times 10^{-1}$). Moreover, results indicate that the strategy utilized in IKNNimpute for information reuse can surpass that employed in SKNNimpute when dealing with MIX and NTS data (especially in the presence of a higher proportion of MVs) and for $TS1^{10\%}$ ($p < 0.03$), while giving statistically similar prediction errors in the other cases (Fig. 1 and Table 2).

The methods were also evaluated in terms of bias using the Wilcoxon signed ranks test. At a 5% significance level, the cluster-based methods behave similarly in terms of generating biased or unbiased estimates. Specifically, the methods originate biased estimates when applied to NTS experiments ($p < 1.2 \times 10^{-5}$), and to TS data with high rate of MVs or unequally distributed MVs ($p < 3.8 \times 10^{-2}$). Unbiased estimates were obtained for TS experiments with a total missing rate of 1% and for MIX experiments.

As a further characterization of the estimation efficiency, we evaluated the capability of the methods to retain the data structure of each array by determining the R^2 value between estimated values and true values for each column (array) of the data matrix. Results indicate that in the presence of unevenly distributed missing entries or on MIX (Fig. 2) and NTS (Fig. 3), experiments SKNNimpute and IKNNimpute offer an advantage over KNNimpute. Moreover, IKNNimpute is more capable of reconstructing the original structure of the data for higher missing rates, particularly when dealing with NTS data.

Despite the fact that the above measures (NRMSE and R^2) are important measures of performance, it is more essential to assess the effect of the methods' estimates for MVs on final outputs of the analysis of microarray data. Therefore, for MIX and NTS data sets, for which the IKNN method has a statistically significant advantage over the other cluster-based methods in terms of prediction ability, we applied the permutation-based SAM procedure to obtain the list of genes differentially expressed for the true complete data sets and after MV estimation by KNNimpute, SKNNimpute and IKNNimpute. Then, we examined the differentially expressed genes that are lost due to MV estimation by comparing the lists of differentially expressed genes. Results are presented in Figs. 4 and 5. For comparison, we have also included in the plots, the results respecting the replacement of missing entries by gene average substitution.

Fig. 4 displays the percentage of lost differentially expressed genes. This percentage increases with the content of missing entries in the data. Depending on the total missing rate, 3.0–9.8% or 1.8–12.5% (respectively for MIX and NTS data) of the genes in the list of differentially expressed genes for the true complete MIX or NTS data sets are lost after estimating the missing entries using clustering-based methods. These values

Table 2
Probability level (p-value) of Levene's test based on transformed residuals for the estimation of MVs using different cluster-based methods (p-values ≤ 0.05 are shown in bold)

Data sets	KNNimpute vs. SKNNimpute	KNNimpute vs. IKNNimpute	SKNNimpute vs. IKNNimpute
$TS1^{1\%}$	8.63×10^{-1}	9.58×10^{-1}	9.05×10^{-1}
$TS1^{5\%}$	2.80×10^{-1}	5.61×10^{-1}	9.61×10^{-2}
$TS1^{10\%}$	5.21×10^{-1}	9.24×10^{-2}	$\mathbf{2.04 \times 10^{-2}}$
$TS1^{uneq}$	$\mathbf{2.93 \times 10^{-2}}$	$\mathbf{1.16 \times 10^{-2}}$	7.29×10^{-1}
$TS2^{1\%}$	7.26×10^{-1}	9.87×10^{-1}	7.38×10^{-1}
$TS2^{5\%}$	9.83×10^{-1}	4.43×10^{-1}	4.30×10^{-1}
$TS2^{10\%}$	7.10×10^{-1}	7.86×10^{-2}	1.62×10^{-1}
$TS2^{uneq}$	7.32×10^{-1}	6.50×10^{-1}	4.26×10^{-1}
$MIX^{1\%}$	$\mathbf{3.81 \times 10^{-2}}$	$\mathbf{2.26 \times 10^{-2}}$	8.35×10^{-1}
$MIX^{5\%}$	$\mathbf{5.61 \times 10^{-4}}$	$\mathbf{8.09 \times 10^{-9}}$	$\mathbf{2.09 \times 10^{-2}}$
$MIX^{10\%}$	$\mathbf{4.81 \times 10^{-10}}$	**0**	$\mathbf{6.37 \times 10^{-5}}$
MIX^{uneq}	**0**	**0**	8.25×10^{-1}
$NTS^{1\%}$	$\mathbf{3.53 \times 10^{-5}}$	$\mathbf{2.34 \times 10^{-10}}$	$\mathbf{2.90 \times 10^{-2}}$
$NTS^{5\%}$	$\mathbf{6.39 \times 10^{-8}}$	**0**	**0**
$NTS^{10\%}$	$\mathbf{4.91 \times 10^{-11}}$	**0**	**0**
NTS^{uneq}	**0**	**0**	4.84×10^{-1}

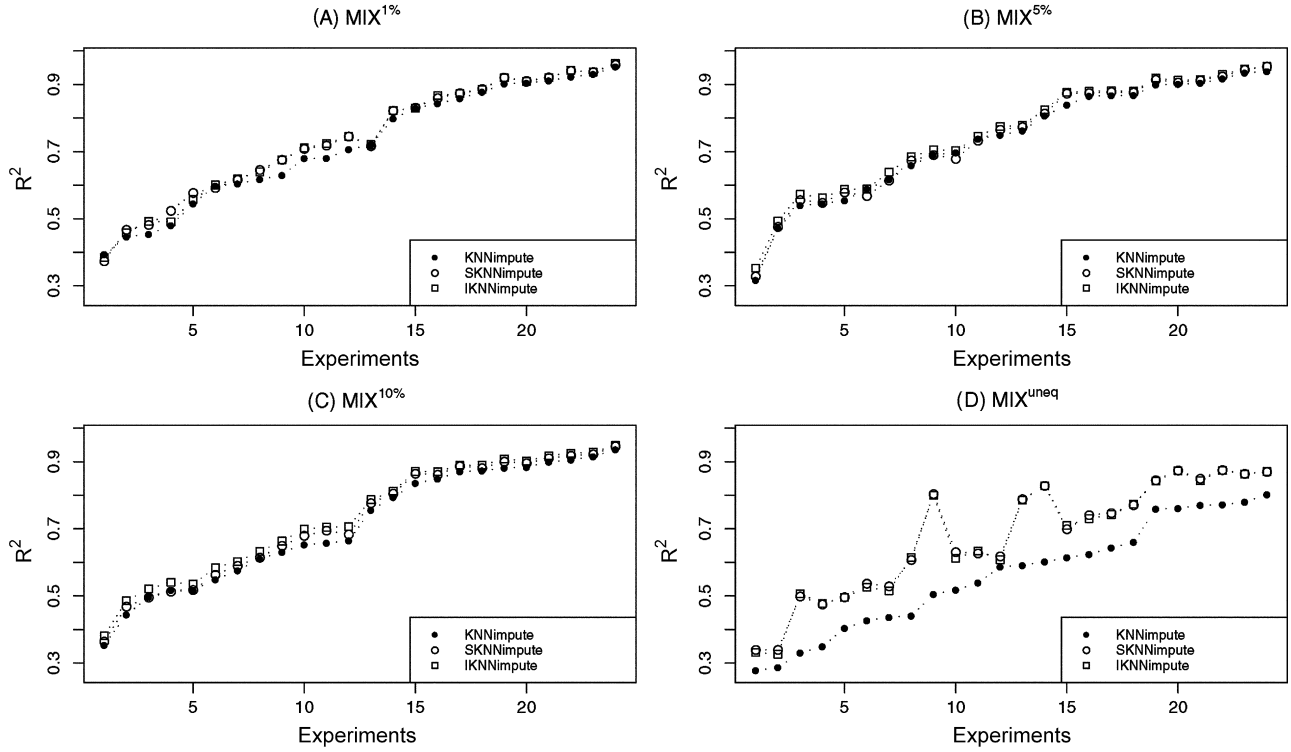

Fig. 2. Correlation (R^2) between true and estimated values within each array when the methods are applied on MIX experiments. To facilitate the visualization, arrays (x-axis) were sorted by increasing the R^2 values of KNNimpute. (A) MIX$^{1\%}$, (B) MIX$^{5\%}$, (C) MIX$^{10\%}$ and (D) MIX$^{\text{uneq}}$ data sets.

represent a significant improvement in comparison with the range 5.1–24.0% (MIX) or 3.3–25.0% (NTS) obtained when the missing entries are substituted by gene averages. The differences between the three cluster-based methods are modest. Nevertheless, the results indicate the better performance of IKNN method for higher missing rates. Besides, in general the results of SAM procedure have a lower variability between runs when the MVs are estimated by IKNNimpute.

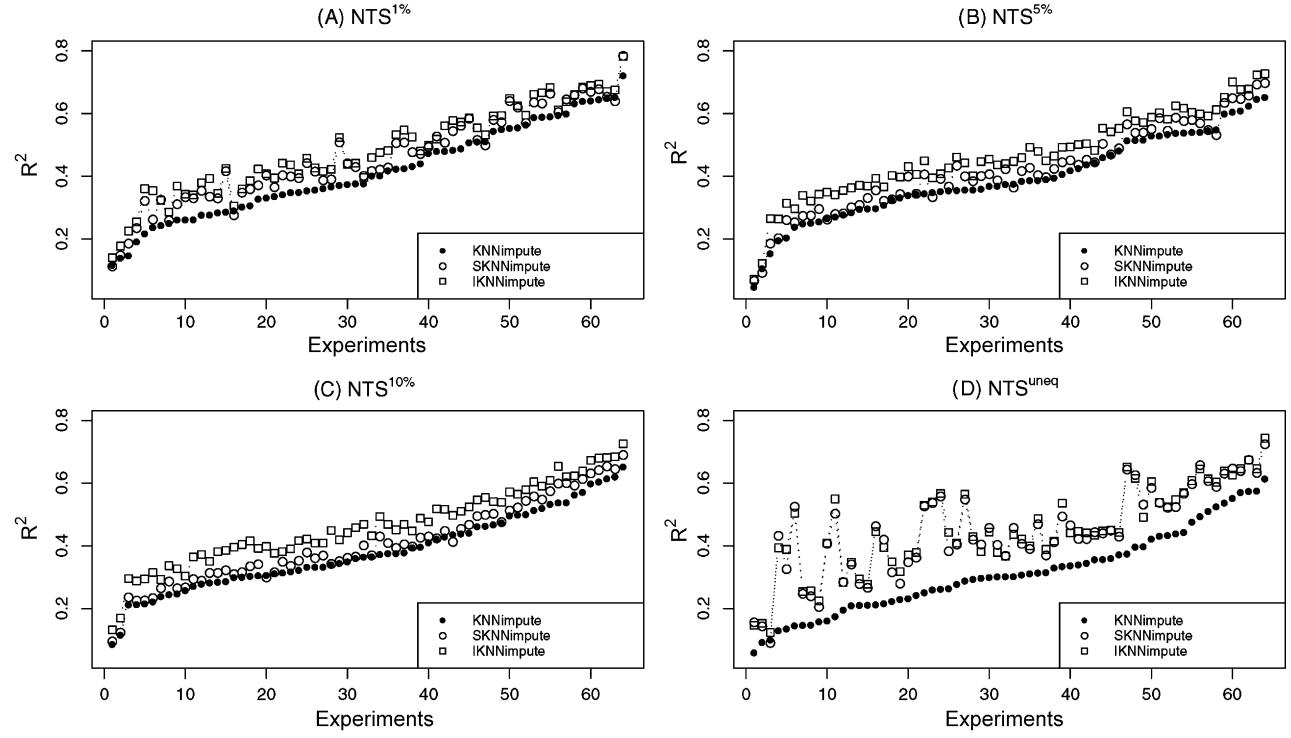

Fig. 3. Correlation (R^2) between true and estimated values within each array when the methods are applied on NTS experiments. To facilitate the visualization, arrays (x-axis) were sorted with respect to the R^2 values of KNNimpute. (A) NTS$^{1\%}$, (B) NTS$^{5\%}$, (C) NTS$^{10\%}$ and (D) NTS$^{\text{uneq}}$ data sets.

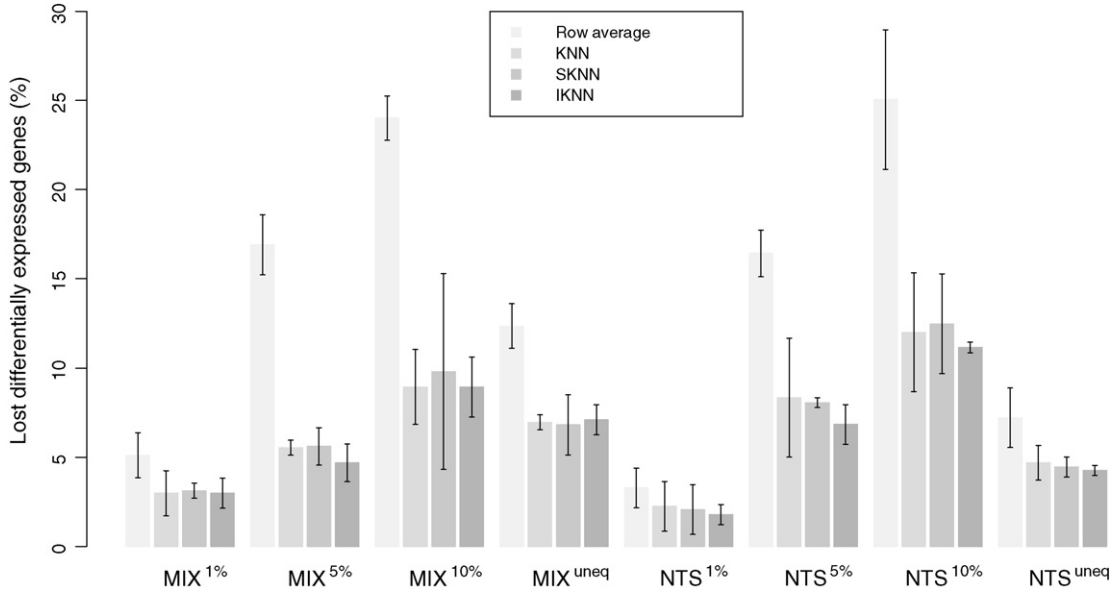

Fig. 4. Percentage of lost differentially expressed genes when analyzing the data sets using SAM. The bars correspond to the average between repeated runs, and the vertical lines represent two standard deviations from the average of repeated runs.

To evaluate the importance of the loss of differentially expressed genes, we further examined the position of the lost genes in the list of differentially expressed genes obtained for the true complete data sets. The ratio between the ranks of the lost genes after replacing the missing entries by gene average substitution, KNN, SKNN or IKNN methods and the length of the original list is presented in Fig. 5, in the form of box plots, for each data set and missingness. The lower the rank of a given gene, the more significant that gene is, and more severe is its loss. As expected, the ranks of the lost genes (expressed as ratios in Fig. 5) decrease with increasing missing rate. Gene average substitution of MVs originates severe losses in the detection of differential expression and KNN-based MVs estimates are capable to recover some of the lost genes, especially those in the top 50% of the original list. For example, for MIX data with more than 1% missing rate, when gene average substitution is applied to estimate the missing entries, almost 30% of the lost differentially expressed genes belong to the top 50% original list; this percentage drops to less than 10% when IKNN method is applied instead.

The pattern of the missing entries in the data matrix is of relevance. Although for MIX^{uneq} and NTS^{uneq} data sets (both of which present a total missing rate of approximately 4%; see Section 2.5) the percentage of lost differentially expressed genes is approximately comparable to the values obtained for 5% of MVs randomly assigned (or at least smaller than the values

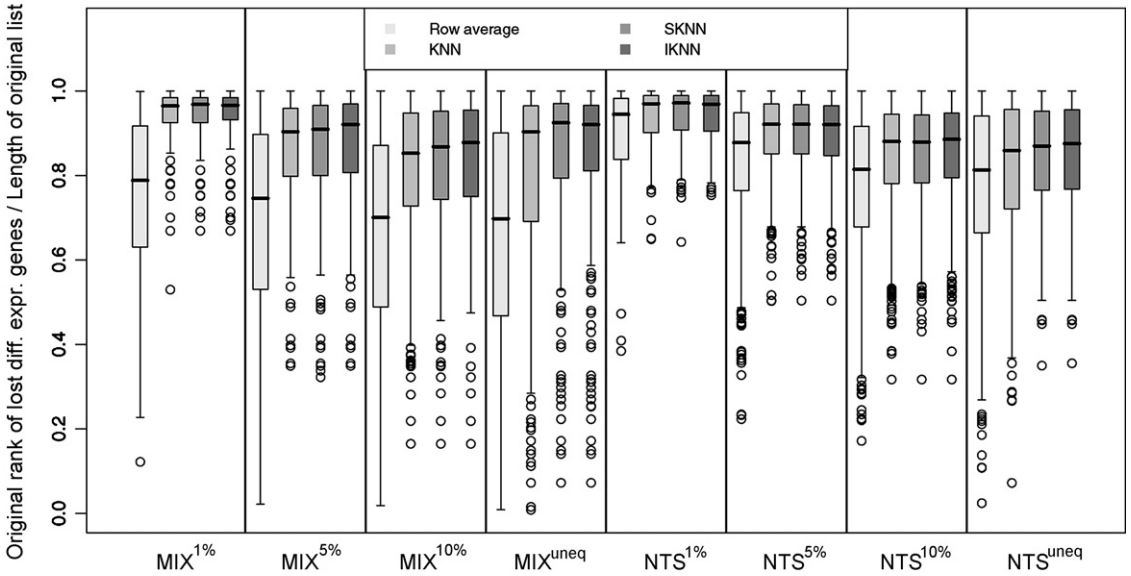

Fig. 5. Box plots of ratio between the original rank of the lost differentially expressed genes and the length of the list of differentially expressed genes obtained for the true full data set.

obtained for 10% missing at random; Fig. 4), there are more lost genes with a low rank than when imputing 5 or 10% at random.

In general, regardless of the cluster-based method used to estimate the missing entries, the effect on detection of differential expression using SAM is similar. However, there are slight improvements when IKNN method is applied for 5 and 10% missing at random, since the median rank of lost differentially expressed genes is higher, and there are fewer genes located at topmost positions of the original list of differentially expressed genes.

4. Conclusion

In this work, we propose a cluster-based method for estimating missing values in DNA microarray data, which was called iterative KNN imputation (IKNNimpute), since it involves the iterative use of estimated data. The prediction performance of IKNNimpute was assessed and compared with that of other cluster-based methods (KNNimpute and SKNNimpute) over different types of data sets (time series, mixed and non-time course experiments) with different proportions (1, 5 and 10%) and patterns (equally and unequally distributed) of missing data, and using different values for the parameter K (i.e. number of nearest neighbouring genes).

In general, using a small number of nearest genes ($K = 5$ or 10) is enough in cluster-based methods because the information brought in by farthest neighbours is irrelevant when compared to the noise they introduce in the MVs estimation process, leading to a decrease in accuracy. Furthermore, we evaluated the possibility of using an automated procedure to select the value for K, and thus overcome the need to perform parameter adjustments in advance, concluding that the expense of time required by such procedure is dispensable.

Using the NRMSE and R^2 between true and estimated MVs as measures of performance, we found that the proposed procedure for the re-utilization of estimated data in IKNN imputation can outperform the one employed in SKNNimpute for the estimation of MVs in MIX and NTS data sets (especially for higher missing rates) and in $TS1^{10\%}$. This is due to the fact that IKNNimpute allows using the information of the genes having MVs more efficiently, and the iterative procedure allows refining the MV estimates.

The methods were additionally compared on the basis of the effect of MV imputation on the detection of differentially expressed genes using SAM procedure, for MIX and NTS experiments. Although the advantage of IKNN method over KNN and SKNN for MIX and NTS data with higher missing rates is small when the outputs of SAM are compared, the variability of the results across runs is lower when IKNNimpute estimates are used, which may suggest that the results for KNN and SKNN might be overoptimistic.

Acknowledgements

The author L.P. Brás would like to thank Foundation for Science and Technology in Portugal for financial support (POSI BD/10302/2002).

References

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

Bø, T.H., Dysvik, B., Jonassen, I., 2004. LSimpute: accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Res. 32, e34.

Brás, L.P., Menezes, J.C., 2006. Dealing with gene expression missing data. IEE Proc. Syst. Biol. 153, 105–119.

de Brevern, A.G., Hazout, S., Malpertuy, A., 2004. Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. BMC Bioinform. 5, 114.

Gentleman, R., Ihaka, R., 1996. R: a language for data analysis and graphics. J. Comput. Graph. Stat. 5, 299–314.

Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J., Zhang, J., 2004. Bioconductor: open software development for computational biology and bioinformatics. Genome Biol. 5, R80.

Jörnsten, R., Wang, H.-Y., Welsh, W.J., Ouyang, M., 2005. DNA microarray data imputation and significance analysis of differential expression. Bioinformatics 21, 4155–4161.

Kerr, M.K., Martin, M., Churchill, G.A., 2000. Analysis of variance for gene expression microarray data. J. Comput. Biol. 7, 819–837.

Kim, K.Y., Kim, B.J., Yi, G.S., 2004. Reuse of imputed data in microarray analysis increases imputation efficiency. BMC Bioinform. 5, 160.

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

Levene, H., 1960. Robust tests for the equality of variance. In: Olkin, I.,Ghurye, S.G., Hoefding, W., Madow, W., Mann, H.B. (Eds.), Contributions to probability and statistics: essays in honor of Harold Hotelling. Stanford University Press, Palo Alto, CA, pp. 278–292.

Little, R., Rubin, D., 1987. Statistical analysis with missing data. Wiley, New York.

Lockhart, D.J., Winzeler, E.A., 2000. Genomics, gene expression and DNA arrays. Nature 405, 827–836.

Manly, B.F.J., 1998. Randomization, bootstrap and Monte Carlo methods in biology. Chapman & Hall, London.

Nguyen, D.V., Wang, N., Carroll, R.J., 2004. Evaluation of missing value estimation for microarray data. J. Data Sci. 2, 347–370.

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

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

Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S.S., de Rijn, M.V., Waltham, M., Pergamenschikov, A., Lee, J.C.F., Lashkari, D., Shalon, D., Myers, T.G., Weinstein, J.N., Botstein, D., Brown, P.O., 2000. Systematic variation in gene expression patterns in human cancer cell lines. Nat. Genet. 24, 227–235.

Rubin, D., 1976. Inference and missing data. Biometrika 63, 581–592.

Schafer, J., Graham, J., 2002. Missing data: our view of the state of the art. Psychol. Methods 7, 147–177.

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

Schena, M., Shalon, D., Davis, R., Brown, P., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270, 467–470.

Schulze, A., Downward, J., 2001. Navigating gene expression using microarrays—a technology review. Nat. Cell Biol. 3, E190–E195.

Siegel, S., Castellan, N.J., 1988. Nonparametric Statistics for Behavioral Sciences. McGraw-Hill, New York, pp. 87–95.

Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B., 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol. Biol. Cell 9, 3273–3297.

Tibshirani, R., Hastie, T., Narasimhan, D., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proc. Natl. Acad. Sci. U. S. A. 99, 6567–6572.

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

Tusher, V., Tibshirani, R., Chu, G., 2001. Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl. Acad. Sci. U. S. A. 98, 5116–5121.

Yoshimoto, H., Saltsman, K., Gasch, A.P., Li, H.X., Ogawa, N., Botstein, D., Brown, P.O., Cyert, M.S., 2002. Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in *Saccharomyces cerevisiae*. J. Biol. Chem. 277, 31079–31088.