BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 1 2007, pages 107–113
doi:10.1093/bioinformatics/btl555

Data and text mining

Towards clustering of incomplete microarray data without the use of imputation

Dae-Won Kim[1,*], Ki-Young Lee[2], Kwang H. Lee[3] and Doheon Lee[4]

[1]School of Computer Science and Engineering, Chung-Ang University, Seoul City, Republic of Korea, [2]Department of Bioengineering, University of California at San Diego, California, USA, [3]Advanced Information Technology Research Center and [4]Department of BioSystems, KAIST, Daejeon, Republic of Korea

Received on May 11, 2006; revised on October 8, 2006; accepted on October 24, 2006
Advance Access publication October 31, 2006
Associate Editor: Satoru Miyano

ABSTRACT

Motivation: Clustering technique is used to find groups of genes that show similar expression patterns under multiple experimental conditions. Nonetheless, the results obtained by cluster analysis are influenced by the existence of missing values that commonly arise in microarray experiments. Because a clustering method requires a complete data matrix as an input, previous studies have estimated the missing values using an imputation method in the preprocessing step of clustering. However, a common limitation of these conventional approaches is that once the estimates of missing values are fixed in the preprocessing step, they are not changed during subsequent processes of clustering; badly estimated missing values obtained in data preprocessing are likely to deteriorate the quality and reliability of clustering results. Thus, a new clustering method is required for improving missing values during iterative clustering process.

Results: We present a method for Clustering Incomplete data using Alternating Optimization (CIAO) in which a prior imputation method is not required. To reduce the influence of imputation in preprocessing, we take an alternative optimization approach to find better estimates during iterative clustering process. This method improves the estimates of missing values by exploiting the cluster information such as cluster centroids and all available non-missing values in each iteration. To test the performance of the CIAO, we applied the CIAO and conventional imputation-based clustering methods, e.g. *k*-means based on KNNimpute, for clustering two yeast incomplete data sets, and compared the clustering result of each method using the *Saccharomyces* Genome Database annotations. The clustering results of the CIAO method are more significantly relevant to the biological gene annotations than those of other methods, indicating its effectiveness and potential for clustering incomplete gene expression data.

Availability: The software was developed using Java language, and can be executed on the platforms that JVM (Java Virtual Machine) is running. It is available from the authors upon request.

Contact: dwkim@cau.ac.kr

1 INTRODUCTION

DNA microarray technology has allowed for the monitoring of the transcript abundance of thousands of genes in parallel under a variety of conditions. Since the diauxic shift (DeRisi *et al*., 1997), sporulation (Chu *et al*., 1998), and the cell cycle (Cho *et al*., 1998) in the yeast *Saccharomyces cerevisiae* were explored, many experiments have been analyzed by various methods to monitor the gene expression levels of various organisms during some biological process. Of the analysis methods proposed to date, clustering has emerged as one of the most popular techniques. Since Eisen *et al*. (1998) first used the hierarchical clustering method to find groups of coexpressed genes, numerous methods have been studied for clustering gene expression data: self-organizing map (Tamayo *et al*., 1999), *k*-means clustering (Tavazoie *et al*., 1999), simulated annealing (Luckshin and Fuchs, 2001), graph-theoretic approach (Xu *et al*., 2001), mutual information approach (Steuer *et al*., 2002), fuzzy *c*-means clustering (Dembele and Kastner, 2003), kernel hierarchical clustering (Qin *et al*., 2003), diametrical clustering (Dhilon *et al*., 2003), quantum clustering with singular value decomposition (Horn and Axel, 2003), bagged clustering (Dudoit and Fridlyand, 2003), CLICK (Sharan *et al*., 2003) and GK (Kim *et al*., 2005).

However, the analysis results obtained by clustering methods will be influenced by missing values in microarray experiments, and thus it is not always possible to correctly analyze the clustering results due to the incompleteness of data sets. The problem of missing values have various causes, including dust or scratches on the slide, image corruption and spotting problems (Troyanskaya *et al*., 2001; Bo *et al*., 2004). Ouyang *et al*. (2004) pointed out that most of the microarray experiments contain some missing entries and >90 % of rows (genes) are affected.

To convert incomplete microarray experiments to a complete data matrix that is required as an input for a clustering method, we must handle the missing values before calculating clustering. To this end, typically we have either removed the genes with missing values or estimated the missing values using an imputation prior to cluster analysis. Of the methods proposed to date, several imputation methods have been demonstrating their effectiveness in building the complete matrix of clustering: missing values are replaced by zeros (Alizadeh *et al*., 2000) or by the average expression value over the row (gene). Troyanskaya *et al*. (2001) presented two correlation-based imputation methods: a singular-value-decomposition-based method (SVDimpute) and weighted K-nearest neighbors (KNNimpute). Besides, a classical expectation maximization approach (EMimpute) exploits the maximum likelihood of the covariance of the data for estimating the missing values (Bo *et al*., 2004; Ouyang *et al*., 2004).

However, a common limitation of existing approaches for clustering incomplete microarray data is that the estimation of missing

*To whom correspondence should be addressed.

values must be calculated in the preprocessing step of clustering. Once the estimates are found, they are not changed during the subsequent steps of clustering. Thus badly estimated missing values during data preprocessing can deteriorate the quality and reliability of clustering results, and therefore drive the clustering method to fall into a local minimum; it prevents missing values from being imputed by better estimates during the iterative clustering process.

To minimize the influence of bad imputation, in the present study we developed a CIAO (Clustering Incomplete data using Alternating Optimization) method for clustering incomplete microarray data, which iteratively finds better estimates of missing values during clustering process. An incomplete gene expression data set is used as an input without any prior imputation. This method preserves the uncertainty inherent in the missing values for longer before final decisions are made, and is therefore less prone to fall into local optima in comparison to conventional imputation-based clustering methods. To achieve this, a method for measuring the distance between a cluster centroid and an incomplete row (a gene with missing values) is proposed, along with a method for estimating the missing attributes using all available information in each iteration. The remainder of this paper is organized as follows: Section 2 describes the formulation of the CIAO method; Section 3 highlights the potential of the CIAO method through several tests on the yeast data sets; and Section 4 presents our concluding remarks.

2 METHOD

The objective of the CIAO method is to classify a set of data points $X = \{x_1, x_2, \ldots, x_n\}$ in a p dimensional space into k disjoint and homogeneous clusters represented as $C = \{C_1, C_2, \ldots, C_k\}$. Here each data point $x_j = [x_{j1}, x_{j2}, \ldots, x_{jp}]$ $(1 \le j \le n)$ is the expression vector of the j-th gene over p different environmental conditions or samples. A data point with some missing conditions or samples is referred to as an incomplete gene; a gene x_j is incomplete if x_{jl} is missing for $\exists 1 \le l \le p$, i.e. an incomplete gene $x_1 = [0.75, 0.73, ?, 0.21]$ where x_{13} is missing. A gene expression data set X is referred to as an incomplete data set if X contains at least one incomplete gene expression vector.

To find better estimates of missing values and improve the clustering result during iterative clustering process, in each iteration we exploit the information of current clusters such as cluster centroids and all available non-missing values. For example, a missing value x_{jl} is estimated using the corresponding l-th attribute value of the cluster centroid to which x_j is closest in each iteration. To improve the estimates during each iteration, the proposed method attempts to optimize the objective function with respect to the missing values, which is often referred to as the alternating optimization (AO) scheme. The objective of the proposed method is obtained by minimizing the function J_m:

$$\min \left\{ J_m(U,V) = \sum_{i=1}^{k} \sum_{j=1}^{n} (\mu_{ij})^m D_{ij} \right\} \tag{1}$$

where

$$D_{ij} = \|x_j - v_i\|^2 \tag{2}$$

is the distance between x_j and v_{ij},

$$V = [v_1, v_2, \ldots, v_k] \tag{3}$$

is a vector of the centroids of the clusters $C_1, C_2, \ldots, C_k$,

$$U = [\mu_{ij}] = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{k1} & \mu_{k2} & \cdots & \mu_{kn} \end{bmatrix} \tag{4}$$

is a fuzzy partition matrix of X satisfying the following constraints,

$$\mu_{ij} \in [0,1], \quad 1 \le i \le k, \quad 1 \le j \le n,$$

$$\sum_{i=1}^{k} \mu_{ij} = 1, \quad 1 \le j \le n, \tag{5}$$

$$0 < \sum_{j=1}^{n} \mu_{ij} < n, \quad 1 \le i \le k.$$

and

$$m \in [1, \infty) \tag{6}$$

is a weighting exponent that controls the membership degree μ_{ij} of each data point x_j to the cluster C_i. As $m \to 1$, J_1 produces a hard partition where $\mu_{ij} \in \{0,1\}$. As m approaches infinity, J_∞ produces a maximum fuzzy partition where $\mu_{ij} = 1/k$. This fuzzy k-means-type approach has advantages of differentiating how closely a gene belongs to each cluster (Dembele and Kastner, 2003) and being robust to the noise in microarray data (Futschik, 2003) because it makes soft decisions in each iteration through the use of membership functions.

Under this formulation, missing values are regarded as optimization parameters over which the functional J_m is minimized. To obtain a feasible solution by minimizing Equation (1), the distance D_{ij} between an incomplete gene x_j and a cluster centroid v_i must be calculated as:

$$D_{ij} = \frac{p}{\sum_{l=1}^{p} \omega_{jl}} \sum_{l=1}^{p} (x_{jl} - v_{il})^2 \omega_{jl} \tag{7}$$

where

$$\omega_{jl} = \begin{cases} 1 & \text{if } x_{jl} \text{ is non-missing} \\ 1 - \exp(-t/\tau) & \text{if } x_{jl} \text{ is missing} \end{cases} \tag{8}$$

We differentiate the missing attribute values from the non-missing values in calculating D_{ij}. The fraction part in Equation (7) indicates that D_{ij} is inversely proportional to the number of non-missing attributes used where p is the number of attributes. ω_{jl} indicates the confidence degree with which l-th attribute of x_j contributes to D_{ij}; specifically, $\omega_{jl} = 1$ if x_{jl} is non-missing and $0 \le \omega_{jl} < 1$ otherwise. The exponential decay, $\exp(-t/\tau)$, represents the reciprocal of the influence of the missing attribute x_{jl} on discrete time t where τ is a time constant. At the initial iteration ($t = 0$), w_{jl} has a value of 0. As time t (i.e. the number of iterations) increases, the exponent part decreases fast, and thus w_{jl} approaches 1. Let us consider an incomplete data point $x_1 = [0.75, 0.73, ?, 0.21]$ where initially x_{13} is missing. Suppose that x_{13} is estimated as a value of 0.52 after two iterations; then x_1 has a vector of [0.75,0.73,0.52,0.21]. From this vector, we see that x_{13} participates in calculating the distance to cluster centroids less than the other three values because it is now being estimated. Besides, the influence of x_{13} to D_{i1} is increased as the iteration continues because its estimate is improved by an iterative optimization.

Using D_{ij} in Equation (7) the saddle point of J_m is obtained by considering the constraint Equation (5) as the Lagrange multipliers:

$$\nabla J_m(U,V,\lambda) = \sum_{i=1}^{k} \sum_{j=1}^{n} (\mu_{ij})^m D_{ij} + \sum_{j=1}^{n} \lambda_j \left[\sum_{i=1}^{k} \mu_{ij} - 1 \right] \tag{9}$$

and by setting $\nabla J_m = 0$. If $D_{ij} > 0$ for all i, j and $m > 1$, then (U,V) may minimize J_m only if,

$$\mu_{ij} = \left[\sum_{z=1}^{k} \left(\frac{D_{ij}}{D_{iz}} \right)^{2/(m-1)} \right]^{-1}, \tag{10}$$

$$1 \le i \le k; \quad 1 \le j \le n$$

and

$$v_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^m x_j}{\sum_{j=1}^{n} (\mu_{ij})^m}, \quad 1 \le i \le k. \tag{11}$$

This solution also satisfies the remaining constraints of Equation (5). Along with the optimization of the cluster centroids and membership degrees in Equations (10) and (11), missing values are optimized during each iteration to minimize the functional J_m. In this study, we optimize the missing values by minimizing the function $J(x_j)$ presented by Hathaway and Bezdek (2001):

$$J(x_j) = \sum_{i=1}^{k} (\mu_{ij})^m \|x_j - v_i\|_A^2 \tag{12}$$

By setting $\nabla J = 0$ with respect to the missing attributes of x_j, a missing value x_{jl} is calculated as:

$$x_{jl} = \frac{\sum_{i=1}^{k} (\mu_{ij})^m v_{il}}{\sum_{i=1}^{k} (\mu_{ij})^m}, \quad 1 \le i \le k. \tag{13}$$

By Equation (13), x_{jl} is estimated by the weighted mean of all cluster centroids in each iteration. At the initial iteration, x_{jl} is initialized with the corresponding attribute of the cluster centroid to which x_j has the highest membership degree.

Algorithm 1. *The CIAO Method*

Given incomplete microarray data $X = \{x_1, \ldots, x_n\}$, $x_j \in R^p$, the number of clusters (k), the weighting exponent (m), and the termination criterion (ε), this method finds k disjoint and homogeneous clusters.

(1) Initialize $U_t = [\mu_{ij}^{(t)}]$ (initially, $t \leftarrow 0$) of x_j belonging to cluster C_i for $1 \le i \le k$, $1 \le j \le n$ such that $\sum_{i=1}^{k} \mu_{ij} = 1.0$. Choose the initial values for cluster centroids V_0 and missing attributes.

(2) Compute the distances between x_j and $v_i^{(t)}$ for $1 \le i \le k$, $1 \le j \le n$ using:

$$D_{ij} = \frac{p}{\sum_{l=1}^{p} \omega_{jl}} \sum_{l=1}^{p} (x_{jl} - v_{il})^2 \omega_{jl}$$

where

$$\omega_{jl} = \begin{cases} 1 & \text{if } x_{jl} \text{ is non-missing} \\ 1 - \exp(-t/\tau) & \text{if } x_{jl} \text{ is missing.} \end{cases}$$

(3) Update U_{t+1} by the following procedure. For each x_j, $1 \le j \le n$,
(a) if $D_{ij} > 0$, $1 \le i \le k$, then update the membership of x_j at $t+1$ by:

$$\mu_{ij}^{(t+1)} = \left[\sum_{z=1}^{k} \left(\frac{D_{ij}}{D_{iz}} \right)^{2/(m-1)} \right]^{-1},$$

(b) if $D_{ij} = 0$ for some $i \in I \subseteq 1, \ldots, k$, then for all $i \in I$, set $\mu_{ij}^{(t+1)}$ to be between [0,1] such that:

$$\sum_{i \in I} \mu_{ij}^{(t+1)} = 1, \text{and}$$
$$\text{set} \quad \mu_{ij}^{(t+1)} = 0 \text{ for other } i \notin I.$$

(4) Update the centroids $V_{t+1} = [v_1^{(t+1)}, \ldots, v_k^{(t+1)}]$ for $1 \le i \le k$ using:

$$v_i^{(t+1)} = \frac{\sum_{j=1}^{n} (\mu_{ij}^{(t+1)})^m x_j}{\sum_{j=1}^{n} (\mu_{ij}^{(t+1)})^m}.$$

(5) Update the estimates of missing attributes in x_{jl}, $1 \le l \le p$ using:

$$x_{jl}^{(t+1)} = \frac{\sum_{i=1}^{k} (\mu_{ij}^{(t+1)})^m v_{il}^{(t+1)}}{\sum_{i=1}^{k} (\mu_{ij}^{(t+1)})^m}, \quad 1 \le i \le k.$$

(6) If $\|V_{t+1} - V_t\| \le \varepsilon$, then stop; otherwise, $t \leftarrow t+1$ and go to Step 2.

Algorithm 1 shows the procedural steps of the CIAO method for clustering $n \times p$ gene expression data where n is the number of genes and p is the number of experiments (attributes). This method iteratively improves a sequence of sets of clusters until no further improvement in $J_m(U,V)$ is possible. It loops through the estimates for $V_t \rightarrow U_{t+1} \rightarrow V_{t+1}$ and terminates on $\| V_{t+1} - V_t \| \le \varepsilon$. Equivalently, the initialization of the algorithm can be done on U_0, and the iterates become $U_t \rightarrow V_{t+1} \rightarrow U_{t+1}$, with the termination criterion $\| U_{t+1} - U_t \| \le \varepsilon$. This way of alternating optimization using membership computation makes the present method be less prone to fall into local minima than conventional clustering methods.

Theorem 1. *The CIAO method given in Algorithm* 1 *converges in a finite number of iterations.*

Proof. We first show that a saddle point of J_m appears at most once by the CIAO method before it stops. Suppose that this is not true, i.e., $U_{t_1} = U_{t_2}$ for some t_1 and t_2 where $t_1 \neq t_2$. By the alternating optimization scheme, we get V_{t_1+1} and V_{t_2+1} as optimal solutions for $U = U_{t_1}$ and $U = U_{t_2}$, respectively. Therefore, we have

$$J_m(U_{t_1}, V_{t_1+1}) = J_m(U_{t_2}, V_{t_1+1}) \quad (\textit{since } U_{t_1} = U_{t_2})$$
$$= J_m(U_{t_2}, V_{t_2+1}) \tag{14}$$

However, the sequence $J_m(\cdot,\cdot)$ generated by the CIAO method is strictly decreasing (Selim and Ismail, 1984*). Hence Equation (*14*) is false and $U_{t_1} \neq U_{t_2}$. Since there are a finite number of saddle points of J_m (Selim and Ismail*, 1984*), hence the algorithm will converge in a finite number of iterations.*

A similar proof concerning the convergence of the *k*-means-type algorithms to a local minimum has been stated by Selim and Ismail 1984.

3 RESULTS

3.1. Data sets and implementation parameters

To test the effectiveness with which the CIAO clusters incomplete microarray data, we applied the CIAO and conventional imputation-based clustering methods to two published yeast data sets and compared the performance of each method.

The data sets employed were the yeast cell-cycle data set of Cho *et al*. (1998) and the yeast sporulation data set of Cho *et al*. (1998). The Cho data set contains the expression profiles of 6200 yeast genes measured at 17 time points over two complete cell cycles. We used the same selection of 2945 genes made by Tavazoie *et al*. (1999) in which the data for two time points (90 and 100 min) were removed. The Chu data set consists of the expression levels of the yeast genes measured at seven time points during sporulation. Of the 6116 gene expressions analyzed by Eisen *et al*. (1998), 3020 significant genes obtained through 2-fold change were used. These three data sets were preprocessed for the test by randomly removing 5–25% of the data in order to create incomplete matrices.

To cluster these incomplete data sets with conventional methods, we first estimated the missing values using the widely used KNNimpute (Troyanskaya *et al*., 2001) and EMimpute (Bo *et al*., 2004; Ouyang *et al*., 2004). For the estimated matrices yielded by each imputation method, we used CLUST 3.0 (Eisen *et al*., 1998) software that implements many clustering methods, of which we investigated the results of the *k*-means method. In these experiments, the parameters used in the CIAO were $\varepsilon = 0.001$, and m, τ values were variously tested. The KNNimpute was tested with $K = 20$; this

value was chosen because it has been overwhelmingly favored in previous studies (Troyanskaya *et al*., 2001).

The choice of the number of clusters are of importance in cluster analysis. However, the problem of the automatic determination of the optimal number of clusters still remains as a hard issue. In the tests reported here, we analyzed the performance of each approach with the number of clusters of $k = 5$, which has been widely used in the two yeast data sets; for the cell-cycle data set, the number of clusters was set to be $k = 5$ in many studies (Cho *et al*. 1998; Yeung *et al*., 2001; Gibbons and Roth 2002). For the sporulation data set, the number of clusters was reported around five (Chu *et al*., 1998; Datta and Datta, 2003).

3.2 Comparison of clustering performance

To show the performance of an imputation, most of the imputation methods proposed to date, including KNNimpute and EMimpute, have examined the root mean squared error (RMSE) between the true values and the imputed values. However, as Bo *et al*. (2004) pointed out, the RMSE is limited to the study the impact of missing value imputation on cluster analysis. To make this study more informative regarding how large an impact the imputation method has on cluster analysis, in the present work the clustering results obtained using the alternative imputations were evaluated by comparing gene annotations using the *z*-score (Gibbons and Roth, 2002; Bo *et al*., 2004). Besides, we analyzed the cluster qualities using the figure of merit (FOM) for an internal validation (Yeung *et al*., 2001).

The *z*-score (Gibbons and Roth, 2002) is calculated by investigating the relationship between clusters produced and the known attributes of the genes in those clusters. To achieve this, this score uses the *Saccharomyces* Genome Database (SGD) annotation of the yeast genes, along with the gene ontology developed by the Gene Ontology Consortium (Ashburner *et al*., 2000; Issel *et al*., 2002). The computation of *z*-score is based on mutual information between a clustering result and the SGD gene annotation; indicating relationships between clustering and annotation. A higher score of *z* represents that the corresponding clustering result is better than random; genes are better clustered by function, indicating a more biologically significant clustering result.

The FOM of Yeung *et al*. (2001) estimates the predictive power of a clustering method based on the jackknife approach Yeung *et al*., (2001). The method measures the root mean square deviation in the left-out condition of the individual gene expression level relative to their within-cluster means. As each condition is used as the validation condition, it calculates the sum of FOMs over all the conditions. Meaningful clusters exhibit less variation in the remaining conditions than clusters formed by random. Thus, a lower value of FOM represents a well-clustered result, representing that a clustering method has high predictive power.

Figure 1 shows the average *z*-scores achieved by the imputation-based *k*-means and CIAO methods over 30 runs for the yeast sporulation and cell-cycle data sets. The *z*-scores of the three methods are plotted with respect to the percentages of missing values (0–25%). The number of neighbors in the KNNimpute was $K = 20$, and the parameters of CIAO were $m = 3.0$ and $\tau = 100$. For the sporulation data set, the *k*-means method using KNNimpute gave *z*-scores from 8.5 to 50.9. The *z*-scores of the EMimpute-based *k*-means method were ranged from 38.9 to 50.7. Compared to these methods, the CIAO method provided better clustering performance for all

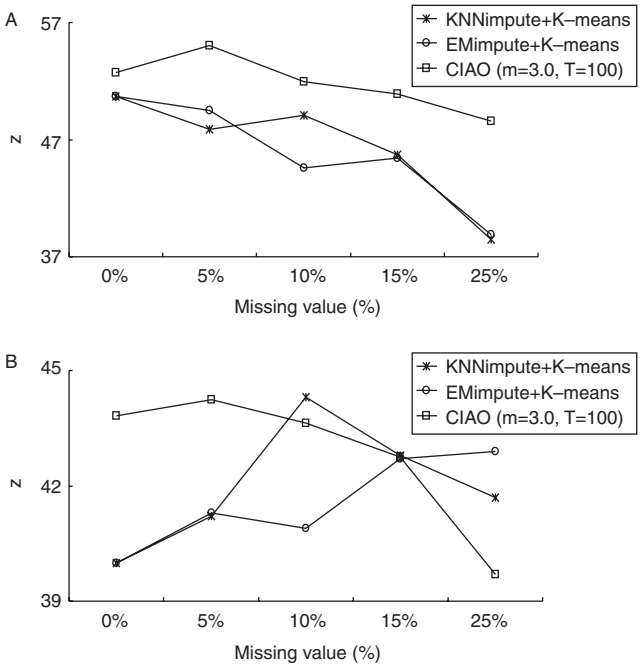

Fig. 1. Comparison of the clustering performance of the imputation-based clustering methods and CIAO for two yeast data sets: (A) Comparison of *z*-scores for the yeast sporulation data set of Chu *et al*. (1998). (B) Comparison of *z*-scores for the yeast cell-cycle data set of Cho *et al*. (1998). The *k*-means method was tested on the data obtained by KNNimpute and EMimpute. The horizontal axis represents the percentages of missing values given, the vertical axis represents the *z*-score.

missing values; the *z*-scores were varied from 48.6 to 55.1 and the standard deviation were ranged from 4 to 7. At no missing value (0%), it was observed that the three methods showed similar *z*-scores. For the cell-cycle data set, the CIAO method provided better clustering performance than other methods at low missing values, giving $z = 44.2$ at 5% and $z = 43.6$ at 10%. Interestingly, at 0% missing value, we see that CIAO gives better *z*-score than other imputation-based methods, which is explained in Figure 2. The best *z*-scores of KNNimpute and EMimpute-based *k*-means methods were $z = 44.3$ and $z = 42.9$, respectively.

Figure 2 shows the comparison of average *z*-scores of CIAO method over 30 runs for different *m* values. The CIAO method uses *m* to control the membership degree μ_{ij} of each datum x_j to the cluster C_i. Although the choice of *m* is of importance in the fuzzy cluster analysis, there is no general agreement on what value to use for the optimal *m* except for the attempt of Dembele and Kastner (2003). In this study we empirically tested various *m* values and reported their influence on the clustering results. Figure 2A shows the clustering performance of CIAO for five $m = 1.1$, 1.5, 2.0, 2.5 and 3.0 values for the sporulation data. Of *m* values considered, CIAO gave the best *z*-scores at $m = 3.0$; it provided more stable performance over the percentage of missing values than other choices. The CIAO with $m = 1.5$ showed the most ineffective performance. For the cell-cycle data (Figure 2(B)), we see that CIAO with $m = 3.0$ also gave the most stable clustering performance. Similar clustering results were obtained at $m = 1.1$, 1.5 and 2.0. In addition, we observe that CIAO with different *m* values

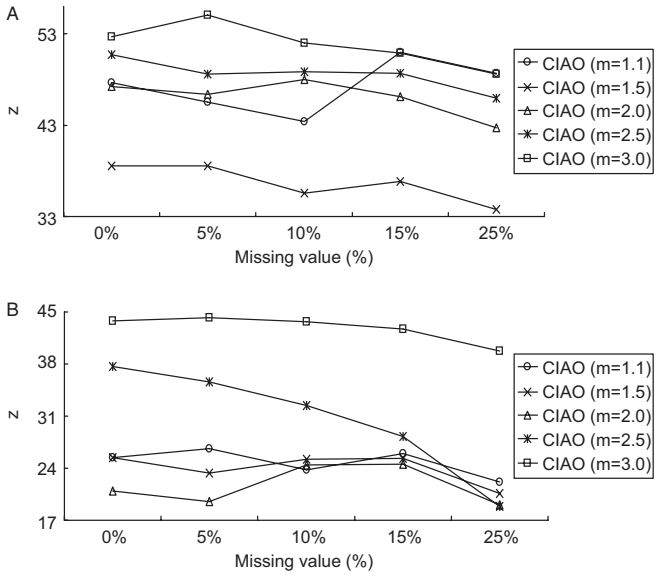

Fig. 2. Comparison of the clustering performance of CIAO method for different m values. (A) Z-scores of CIAO with T = 100 and various m's for the sporulation data. (B) Z-scores of CIAO with T = 100 and various m's for the cell-cycle data.

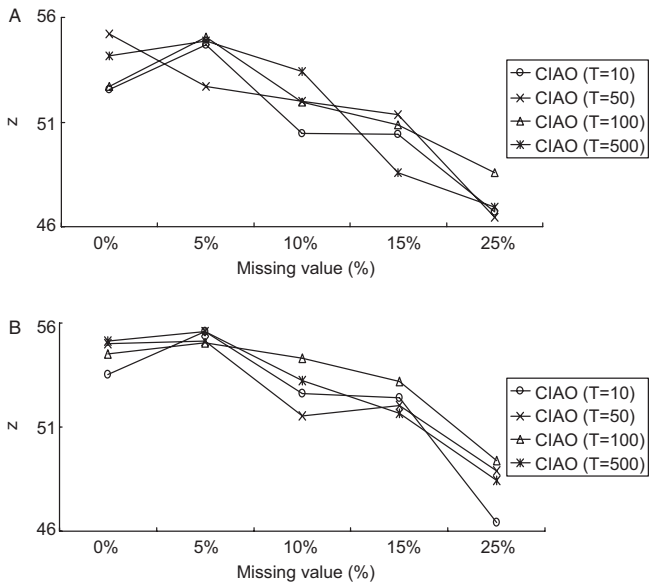

Fig. 3. Comparison of the clustering performance of CIAO method for different τ values. (A) Z-scores of CIAO with m=3.0 and various T's for the sporulation data. (B) Z-scores of CIAO with m=3.0 and various T's for the cell-cycle data.

yielded different z-scores at 0% missing value; it showed better performance with $m = 2.5$ and 3.0 than with other m values. This explains why CIAO in Figure 1 showed better z-scores at 0% missing than the imputation-based k-means methods did. From the result of Figure 2, we see that the choice of $m = 3.0$ shows more stable performance compared with other m values. Moreover, we tested the performance of CIAO for two values ($m = 5.0$ and 10.0) in order to investigate the clustering result of CIAO with $m > 3$. Compared with the case of $m = 3.0$, the CIAO with $m = 5.0$, 10.0 showed similar performance results for the sporulation and cell-cycle data sets. For the sporulation data set, CIAO gave z-scores from 48 to 54 over both $m = 5.0$ and 10.0. For the cell-cycle data set, CIAO yielded z-scores from 37 to 45 over both m values.

Besides the issue of m, CIAO has another parameter, τ, a time constant. We investigated the influence of the choice of τ on the clustering results in Figure 3. For the sporulation data (Figure 3(A)), CIAO with different $\tau = 10$, 50, 100 and 500 values showed similar performances over 5–15% missing values. At 0% missing, the best z-score was obtained at $\tau = 50$ whereas the CIAO with $\tau = 100$ showed better result at 25% missing value. For the cell-cycle data (Figure 3B), CIAO showed similar patterns of z-scores for $\tau = 10$, 50, 100 and 500. We observe that the performance of CIAO is less insensitive to the choice of τ than that of m values.

Table 1 lists the comparison results of the average FOMs of the imputation-based clustering methods and CIAO for the yeast sporulation and cell-cycle data sets over five times. The standard deviations of the methods used were 0.03–0.04 for the sporulation data, 0.1–0.2 for the cell-cycle data. Of the methods considered, the EMimpute-based k-means gave better FOMs than the other methods for the two datasets. The KNNimpute-based k-means and CIAO gave similar FOMs over the missing range. However, as shown in the table, the differences of FOMs were not significant enough to explain the superiority of one method to another. This is the typical limitation of the internal validation measures as pointed out by Yeung *et al*. (2001). The internal validation use information within the given data set only in order to compute the goodness of the clustering results. Figure 4 shows the comparison of RMSE of the imputation methods and CIAO for the incomplete data sets. From the comparison results for the sporulation data, the KNNimpute gave better RMSE at lower missing values whereas CIAO gave better RMSE at higher missing values. The EMimpute shows the most ineffective of the methods considered. As for the cell-cycle data, we see that RMSE of each method increases as the missing value increases. However, as mentioned earlier, RMSE is limited to investigate the impact of the both imputation and clustering together, indicating that better RMSE does not necessarily lead to better z-scores and FOM.

Finally to compare the performance directly, we applied the k-means to the CIAO-imputed data, and applied CIAO clustering to the data imputed by KNNimpute and EMimpute methods. In

Table 1. Comparison of clustering performance of the KNNimpute, EMimpute-based k-means and CIAO methods for the sporulation and cell-cycle data sets

Data set	Method/%missing	5%	10%	15%	25%
Sporulation	KNNimpute+k-means	1.31	1.31	1.30	1.30
	EMimpute+k-means	1.26	1.24	1.28	1.27
	CIAO	1.28	1.33	1.29	1.30
Cell-cycle	KNNimpute+k-means	4.23	4.08	4.23	4.11
	EMimpute+k-means	3.97	3.92	4.06	4.09
	CIAO	4.01	4.06	4.11	4.13

The FOMs of each method are specified.

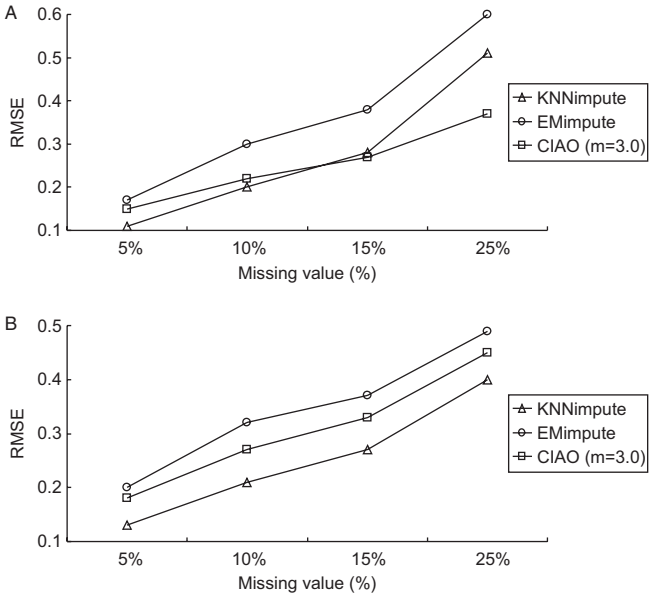

Fig. 4. Comparison of RMSE of the imputation methods and CIAO for two yeast data sets: (A) Comparison of RMSE for the yeast sporulation data set of Chu *et al*. (1998). (B) Comparison of RMSE for the yeast cell-cycle data set of Cho *et al*. (1998).

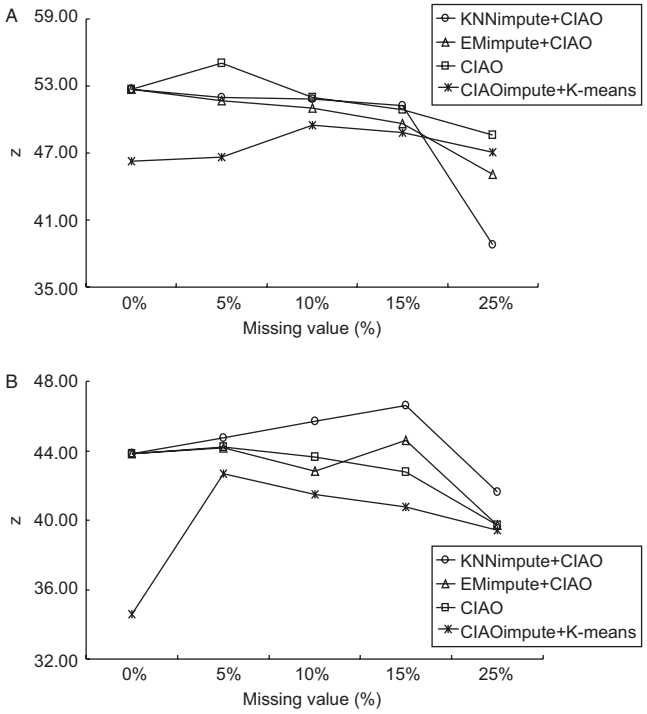

Fig. 5. Performance comparison of CIAO when used as an imputation method only and a stand-alone clustering algorithm, respectively. (A) Z-scores of CIAO as an imputation and a clustering method for the sporulation data. (B) Z-scores of CIAO as an imputation and a clustering method for the cell-cycle data.

Figure 5(A), we see that CIAO clustering showed similar performances at low missing values regardless of the imputation methods. When CIAO clustering was applied to the KNNimputed data at 25% missing value, it gave lower *z*-score than the stand-alone CIAO method. Compared to the KNNimpute/EMimpute-based *k*-means in Fig. 1, the KNNimpute/EMimpute-based CIAO methods showed better clustering results especially at 10 and 15% missing values. Of the methods considered, the *k*-means method applied to the CIAO-imputed data showed the most unstable clustering results. For the cell-cycle data, the KNNimpute/EMimpute-based CIAO showed better clustering performance than the imputation-based *k*-means method as well. The *k*-means method using CIAO-impute data showed the lowest *z*-scores of the methods considered. We see from these tests that the CIAO method shows better performance when it is applied for clustering incomplete data rather than when applied just as an imputation method.

The results of the comparison tests indicate that the CIAO method gave better clustering performance than the other imputation-based methods considered, highlighting the effectiveness and potential of the CIAO method. Furthermore, the KNN/EM/CIAOimpute-based K-means methods often showed non-monotonic shapes. We think that the results stems from the fact that the k-means method is likely to fall into local optima unless the initial centroids are correctly selected.

4 CONCLUSION

Clustering has been used as a popular technique for analysis of large amounts of microarray gene expression data, and many clustering methods have been developed in biological research. However, conventional clustering methods have required a complete data matrix as input even if many microarray data sets are incomplete due to the problem of missing values. In such cases, typically either genes with missing values have been removed or the missing values have been estimated using imputation methods prior to the cluster analysis. In the present study, we focused on the bad influence of the earlier imputation on the subsequent cluster analysis. To address this problem, we have presented the CIAO method of clustering incomplete gene expression data. By taking the alternative optimization approach, the missing values are considered as additional parameters for optimization. The evaluation results based on gene annotations have shown that the CIAO method is the superior and effective method for clustering incomplete gene expression data.

Besides the issues mentioned in present work, several issues require further investigation. The number of clusters is given a priori by a user. We aim to use the cluster validity techniques to develop a method for systematically determining the optimal number of clusters for a given data set. In addition, we initialized missing values with the corresponding attributes of the cluster centroid to which the incomplete data point is closest. Although this way of initialization is considered appropriate, further work examining the impact of different initializations on clustering performance is needed.

ACKNOWLEDGEMENTS

This work was supported by the National Research Laboratory Grant (2005-01450) and the Korean Systems Biology Research Grant (2005-00343) from the Ministry of Science and Technology. We

would like to thank CHUNG Moon Soul Center for BioInformation and BioElectronics for providing research facilities.

Conflict of Interest: none declared.

REFERENCES

Alizadeh,A.A. *et al*. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Ashburner,M. *et al*. (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet*., **25**, 25–29.

Bo,T.H. *et al*. (2004) LSimpute: accurate estimation of missing values in microarray data with least square methods. *Nucleic Acids Res*., **32**, e34.

Cho,R.J. *et al*. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*, **2**, 65–73.

Chu,S. *et al*. (1998) The transcriptional program of sporulation in budding yeast. *Science*, **282**, 699–705.

Datta,S. and Datta,S. (2003) Comparions and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, **19**, 459–466.

Dembele,D. and Kastner,P. (2003) Fuzzy c-means method for clustering microarray data. *Bioinformatics*, **19**, 973–980.

DeRisi,J.L. *et al*. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **282**, 257–264.

Dhilon,I.S. *et al*. (2003) Diametrical clustering for identifying anti-correlated gene clusters. *Bioinformatics*, **19**, 1612–1619.

Dudoit,S. and Fridlyand,J. (2003) Bagging to improve the accuracy of a clustering procedure. *Bioinformatics,* **19**, 1090–1099.

Eisen,M. *et al*. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci. USA*, **95**, 14863–14868.

Fuschik,M.E. (2003) Methods for knowledge discovery in microarray data. Ph.D. Thesis, University of Otago, Dunedin, New Zealand.

Gibbons,F.D. and Roth,F.P. (2002) Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res*., **12**, 1574–1581.

Hathaway,R.J. and Bezdek,J.C. (2001) Fuzzy c-means clustering of incomplete data. *IEEE Trans. Sys. Man Cybernet. B: Cybernetics*, **31**, 735–744.

Horn,D. and Axel,I. (2003) Novel clustering algorithm for microarray expression data in a truncated SVD space. *Boinformatics*, **19**, 1110–1115.

Issel-Tarver,L. *et al*. (2002) *Saccharomyces* Genome Database. *Methods Enzymol.,* **350**, 329–346.

Kim,D.W. *et al*. (2005) Detecting clusters of different geometrical shapes in microarray gene expression data. *Bioinformatics*, **21**, 1927–1934.

Lukashin,A.V. and Fuchs,R. (2001) Analysis of temporal gene expression profiles: clustering by simuulated annealing and determining the optimal number of clusters. *Bioinformatics*, **17**, 405–414.

Ouyang,M. *et al*. (2004) Guassian mixture clustering and imputation of microarray data. *Bioinformatics*, **20**, 917–923.

Qin,J. *et al*. (2003) Kernel hierarchical gene clustering from microarray gene expression data. *Bioinformatics*, **19**, 2097–2104.

Selim,S. and Ismail,M. (1984) K-means type algorithms: a generalized convergence theorem and the caracterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell*., **6**, 284–288.

Sharan,R. *et al*. (2003) CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics*, **19**, 1787–1799.

Steuer,R. *et al*. (2002) The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, **18**, S231–S240.

Tamayo,P. *et al*. (1999) Interpreting patters of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl Acad. Sci. USA*, **96**, 2907–2912.

Tavazoie,S. *et al*. (1999) Systematic determination of genetic network architecture. *Nat. Genet*., **22**, 281–285.

Troyanskaya,O. *et al*. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.

Xu,Y. *et al*. (2001) Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. *Bioinformatics*, **17**, 309–318.

Yeung,K. *et al*. (2001) Validating clustering for gene expression data. *Bioinformatics*, **17**, 309–318.