BIOINFORMATICS

Vol. 20 no. 6 2004, pages 917–923
DOI: 10.1093/bioinformatics/bth007

Gaussian mixture clustering and imputation of microarray data

Ming Ouyang[1,*], William J. Welsh[2] and Panos Georgopoulos[1]

[1]Environmental and Occupational Health Sciences Institute, UMDNJ–Robert Wood Johnson Medical School and Rutgers, The State University of New Jersey, 170 Frelinghuysen Road, Piscataway, NJ 08854, USA and [2]Department of Pharmacology, UMDNJ–Robert Wood Johnson Medical School and Informatics Institute, University of Medicine and Dentistry of New Jersey, 675 Hoes Lane, Piscataway, NJ 08854, USA

Received on 15 September 2003; accepted on 4 November 2003
Advance Access publication January 29, 2004

ABSTRACT

Motivation: In microarray experiments, missing entries arise from blemishes on the chips. In large-scale studies, virtually every chip contains some missing entries and more than 90% of the genes are affected. Many analysis methods require a full set of data. Either those genes with missing entries are excluded, or the missing entries are filled with estimates prior to the analyses. This study compares methods of missing value estimation.

Results: Two evaluation metrics of imputation accuracy are employed. First, the root mean squared error measures the difference between the true values and the imputed values. Second, the number of mis-clustered genes measures the difference between clustering with true values and that with imputed values; it examines the bias introduced by imputation to clustering. The Gaussian mixture clustering with model averaging imputation is superior to all other imputation methods, according to both evaluation metrics, on both time-series (correlated) and non-time series (uncorrelated) data sets.

Availability: Matlab code is available on request from the authors.

Contact: ouyang@fidelio.rutgers.edu

1 INTRODUCTION

Microarray gene expression data can be represented as an $m \times n$ matrix, A. The rows correspond to the genes, the columns correspond to the experiments, and the entry $A_{i,j}$ is the expression level of gene i in experiment j. Let A_i be row i of A: the profile of gene i across the experiments. Cluster analysis is commonly applied to microarray data. Clustering methods usually fall into two categories: hierarchical methods (Eisen *et al*., 1998) and relocational methods. Gaussian mixture clustering is a relocational method. Starting from an initial partition of the genes, it iteratively moves genes from one cluster (or component) to another, until the criterion of convergence is met. The number of clusters must be specified in advance. K-means clustering (Hartigan, 1975) corresponds to a special case of Gaussian mixture clustering (Celeux and Govaert, 1992). There are several statistics that estimate the number of clusters, such as the statistic B (Fowlkes and Mallows, 1983), the silhouette statistic (Kaufman and Rousseeuw, 1990), the gap statistic (Tibshirani *et al*., 2001). There are resampling procedures that determine the number of clusters (Levine and Domany, 2001; Yeung *et al*., 2001b; Ben-Hur *et al*., 2002). With Gaussian mixture clustering, the Bayesian information criterion (BIC; Schwarz, 1978) and the Bayes factor (Kass and Raftery, 1995) can be applied to select the number of components.

In microarray experiments, missing entries arise from blemishes on the chips. In large-scale studies involving thousands to tens of thousands of genes and dozens to hundreds of experiments, the problem of missing entries can be severe. Virtually every experiment (column) contains some missing entries and more than 90% of the genes (rows) are affected. Many analysis methods require a full set of data. Either those genes with missing entries are excluded, or the missing entries are filled with estimates prior to the analyses. This study compares methods of missing value estimation. Two evaluation metrics of imputation accuracy are employed. First, the root mean squared error (RMSE) measures the difference between the true values and the imputed values. Second, the number of mis-clustered genes measures the difference between clustering with true values and that with imputed values; it examines the bias introduced by imputation to clustering.

A simple imputation method is to fill the missing entries with zeros (ZEROimpute). With some calculation, the row or column averages (ROWimpute and COLimpute) can be used. Troyanskaya *et al*. (2001) compared ROWimpute, k nearest neighbor imputation (KNNimpute), and singular value decomposition based imputation (SVDimpute). They

*To whom correspondence should be addressed at Informatics Institute, University of Medicine and Dentistry of New Jersey, 675 Hoes Lane, Piscataway, NJ 08854, USA.

found that KNNimpute and SVDimpute were vastly superior to ROWimpute in terms of RMSE. Bar-Joseph *et al*. (2002) described a model-based spline fitting method for time-series data; it can estimate missing entries at observed time points, and it can also predict entire columns of data at unobserved time points.

We propose an imputation method GMCimpute, based on Gaussian mixture clustering and model averaging. The microarray data are assumed being generated by a Gaussian mixture of some number of components. A multitude of estimates are computed. For a missing entry, an estimate is made from each of the components in the mixture; the estimate by the mixture is a linear combination of the component-wise estimates, weighted by the probabilities that the gene belongs to the components. The final estimate by GMCimpute is the average of the estimates by several mixtures. We use two data sets in simulations: the yeast cell cycle data (Eisen *et al*., 1998), and the yeast environmental stress data (Gasch *et al*., 2000). GMCimpute is more accurate with statistical significance than KNNimpute, SVDimpute, ROWimpute, COLimpute and ZEROimpute, in terms of both evaluation metrics.

2 METHODS AND DATA

2.1 Gaussian mixture clustering

Yeung *et al*. (2001a) and Ghosh and Chinnaiyan (2002) have considered Gaussian mixture clustering of microarray data, but they did not apply the method to missing value estimations. In a Gaussian mixture, each component is modeled by a multivariate normal distribution. The parameters of component k comprise the mean vector μ_k and the covariance matrix Σ_k, and the probability density function is

$$f_k(A_i|\mu_k, \Sigma_k) = \frac{\exp\left\{-\frac{1}{2}(A_i - \mu_k^{\mathrm{T}})\Sigma_k^{-1}(A_i^{\mathrm{T}} - \mu_k)\right\}}{|2\pi\,\Sigma_k|^{1/2}}.$$

Let K be the number of components in the mixture, and let τ_ks be mixing proportions: $0 < \tau_k < 1$, $\sum_k \tau_k = 1$. Then the likelihood of the mixture is

$$\mathcal{L}(\mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K|A) = \prod_{i=1}^{m}\sum_{k=1}^{K}\tau_k f_k(A_i|\mu_k, \Sigma_k).$$

Σ_k determines the geometric properties of component k, C_k. Banfield and Raftery (1993) proposed a general framework for parameterization of Σ_k, and Celeux and Govaert (1995) discussed 14 parameterizations. The parameterization restricts the components to having some common properties, such as spherical or elliptical shapes, and equal or unequal volumes. We use the unconstrained model of Σ_k, to be described below.

Given the value of K, there are two steps in Gaussian mixture clustering. The first step initializes the mixture by partitioning the A_is into K subsets. We use the classic k-means clustering with the Euclidean distance to obtain the initial partition. The k-means clustering itself requires the initial K means, and we use the technique of Bradley and Fayyad (1998) to compute them. Let $\{C_1, \ldots, C_K\}$ be the partition. The second step uses the iterative Classification Expectation–Maximization algorithm (CEM Banfield and Raftery, 1993) to maximize the likelihood of the mixture. There are three steps in CEM. In the Maximization step, μ_k, Σ_k and τ_k, $k = 1, \ldots, K$, are estimated from the partition; specifically,

$$\mu_k = \frac{\sum_{A_i \in C_k} A_i^{\mathrm{T}}}{|C_k|},$$

$$\Sigma_k = \frac{1}{|C_k|}\sum_{A_i \in C_k}(A_i^{\mathrm{T}} - \mu_k)(A_i - \mu_k^{\mathrm{T}}),$$

$$\tau_k = \frac{|C_k|}{m}.$$

In the Expectation step, the probabilities $t_k(A_i)$ that A_i is generated by component k, $i = 1, \ldots, m$, $k = 1, \ldots, K$, are computed; specifically

$$t_k(A_i) = \frac{\tau_k f_k(A_i|\mu_k, \Sigma_k)}{\sum_{l=1}^{K}\tau_l f_l(A_i|\mu_l, \Sigma_l)}.$$

In the Classification step, the partition $\{C_1, \ldots, C_K\}$ is updated; A_i is assigned to C_k if $t_k(A_i)$ is the maximum among $t_1(A_i), \ldots, t_K(A_i)$. CEM repeats the three steps till the partition $\{C_1, \ldots, C_K\}$ converges.

2.2 GMCimpute

In GMCimpute, data are modeled by Gaussian mixtures, and missing entries are estimated by the Expectation–Maximization algorithm (EM; Dempster *et al*., 1977). Assume missing entries are permanently highlighted, so even after GMCimpute inserts values, it can still update the estimates. Figure 1 is the algorithm. It uses `K_estimate` to estimate the missing entries by $1, \ldots, S$-component mixtures; the value of S is empirically determined. Each missing entry then has S estimates; the final estimate is the average of them. Let B be the complete rows of A. `K_estimate` has two parts. The first part initializes the missing entries by first obtaining the Gaussian mixture clustering of B, then estimating the missing entries by `EM_estimate`. Let A' be the matrix with the initial estimates. The second part consists of a loop that repeatedly computes the Gaussian mixture clustering of A', and updates the estimates. After each pass through the loop, we use the parameters $\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K, \tau_1, \ldots, \tau_K$ to classify the rows of A'. A'_i is assigned to cluster k if $t_k(A'_i)$ is the maximum among $t_1(A'_i), \ldots, t_K(A'_i)$. The loop is terminated when the cluster memberships of two consecutive passes are identical. The `EM_estimate` procedure uses the EM algorithm to estimate the missing entries row by row. To simplify notation, we write R (in addition to A_i) as a row of the matrix. Since there are K components, each missing entry

```
EM_estimate(μ_1,...,μ_K,Σ_1,...,Σ_K,τ_1,...,τ_K,A')
{  for each row R of A' with missing values
   {  for i = 1,...,K
      {  Use EM and N(μ_i,Σ_i) to estimate the
            missing values in R.
         R_i ← R with missing values
               replaced by estimates.
      }
      R' ← WeightedAverage(R_1,...,R_K).
      Replace R in A' by R'.
   }
   return A'.
}

K_estimate(K,A)
{  /* first part:  initialization */
   B ← rows of A without missing values.
   μ_1,...,μ_K,Σ_1,...,Σ_K,τ_1,...,τ_K ←
      Gaussian mixture clustering of B.
   A' ← EM_estimate(μ_1,...,μ_K,Σ_1,...,Σ_K,
         τ_1,...,τ_K,A).
   /* second part:  iteration */
   repeat
   {  μ_1,...,μ_K,Σ_1,...,Σ_K,τ_1,...,τ_K ←
         Gaussian mixture clustering of A'.
      A' ← EM_estimate(μ_1,...,μ_K,Σ_1,...,Σ_K,
            τ_1,...,τ_K,A').
   }  until convergence
}

GMCimpute(S,A)
{  for K = 1,2,...,S
   {  A_K ← K_estimate(K,A).
   }
   return (A_1 + A_2 + ... + A_S)/S.
}
```

Fig. 1. The GMCimpute algorithm.

has K estimates: $R_1, \ldots, R_K$. The weighted average R' of R_ks is defined by:

$$R' = \frac{\sum_{i=1}^{K} R_i \tau_i f_i(R_i|\mu_i, \Sigma_i)}{\sum_{i=1}^{K} \tau_i f_i(R_i|\mu_i, \Sigma_i)}.$$

`EM_estimate` and `K_estimate` may not converge in all cases, but they did converge in our simulation runs. Note that, in the initialization part of `K_estimate`, Gaussian mixture clustering is applied to the subset of complete rows. In subsequent iterations, all rows are used. Thus non-missing information of incomplete rows is incorporated from the second iteration and on. If we were to use all rows in the initialization, we would need to apply some other imputation method prior to GMCimpute. As we found out by simulations, all other methods are not as accurate. Thus using other methods prior to GMCimpute will introduce their biases to imputation.

2.3 KNNimpute and SVDimpute

KNNimpute and SVDimpute were studied in the context of microarray data imputation by Troyanskaya *et al.* (2001). There are n columns in A. Let t be the number of missing entries in a row R, $1 \leq t < n$; assume the missing entries are in columns $1, \ldots, t$. Let B be the complete rows of A. Both KNNimpute and SVDimpute require a parameter K, which is determined empirically. KNNimpute finds K rows, $R_1, \ldots, R_K$, in B, that have the shortest Euclidean distances to R in the $(n-t)$-dimensional space (columns $t+1, \ldots, n$). Let d_k be the Euclidean distance from R_k to R, and let us write $R^{(j)}$ for the j-th element of R. Then the missing entries of R are estimated by: for $j = 1, \ldots, t$,

$$R^{(j)} = \frac{\sum_{k=1}^{K} R_k^{(j)}/d_k}{\sum_{k=1}^{K} 1/d_k}.$$

In Singular Value Decomposition (SVD; Watkins, 1991), the $m \times n$ matrix A, $m > n$, is expressed as the product of three matrices: $A = U\Sigma V^{\mathrm{T}}$, where the $m \times m$ matrix U and the $n \times n$ matrix V are orthogonal matrices, and Σ (not related to the covariance matrices of multivariate normal distributions) is an $m \times n$ matrix that contains all zeros except for the diagonal $\Sigma_{i,i}, i = 1, \ldots, n$. These diagonal elements are rank-ordered ($\Sigma_{1,1} \geq \cdots \geq \Sigma_{n,n} \geq 0$) square roots of the eigenvalues of AA^{T}. Holter *et al.* (2000) showed that the product of the first two or three columns of $U\Sigma$ and the corresponding rows of V^{T} can capture the fundamental patterns in cell cycle data.

Let $R_1, \ldots, R_K$ be the first K rows of V^{T}, and let R be a row of A with the first t entries missing. The estimation procedure of SVDimpute performs a linear regression of the last $n-t$ columns of R against the last $n-t$ columns of $R_1, \ldots, R_K$. Let c_k be the regression coefficients. Then the missing entries of R are estimated by: for $j = 1, \ldots, t$,

$$R^{(j)} = \sum_{k=1}^{K} c_k R_k^{(j)}.$$

SVDimpute first performs SVD on B, then it uses the estimation procedure on each incomplete row of A. Let A' be the imputed matrix. Then SVDimpute repeatedly performs SVD on A', then updates A' by the estimation procedure, until the root mean squared error (defined in the next section) between two consecutive A's falls below 0.01. Note that Troyanskaya *et al.* (2001) used ROWimpute to compute the first A', whereas the SVDimpute described here uses SVD on B to initialize the iterations.

2.4 Data, simulation and evaluation

The method of creating missing entries is: each entry in the complete matrix is randomly and independently marked as missing with a probability p. For each of the two data sets to be described next, we use four missing probabilities to render different proportions of missing entries.

The yeast cell cycle data[1] (Eisen *et al*., 1998) has 6221 genes (rows) and 80 experiments (columns). The columns are correlated; in fact, some columns are replicated experiments. In the original data, each column has at least 182, and up to 765 missing entries. If a missing entry arises randomly and independently with probability p, then the expected number of genes with s missing entries is

$$E_M = 6221 \begin{pmatrix} 80 \\ s \end{pmatrix} p^s (1-p)^{80-s}.$$

3222 genes have no missing entry; solving for p when $E_M = 3222$ and $s = 0$, we get $p \approx 0.0082$. Similarly, 1583 genes have one missing entry, $p \approx 0.0265$; 478 genes have two, $p \approx 0.0063$; 178 genes have three, $p \approx 0.0088$. We use the complete 3222×80 matrix, and p of 0.003, 0.005, 0.007 and 0.009 in the simulations.

The yeast environmental stress data (Gasch *et al*., 2000) in Stanford Microarray Database (Sherlock *et al*., 2001) contains 6361 rows and 156 columns. There are over a dozen stress treatments to yeast cells. After each treatment, the time-series expression data are collected. In contrast to the correlated columns in the cell cycle data, for the stress data we aim to study a subset of the 156 columns that are uncorrelated representatives of gene expression under different conditions. For some treatments, there are a transient response and a stationary response in gene expression. As an example, Table 1 shows the two cliques of early and late time points of amino acid starvation that have large Pearson correlation coefficients within each clique. In such a case, we choose the time point in the clique that has the fewest missing entries as the representative, thus denying imputation methods the information embedded in correlated columns. Fifteen columns[2] are chosen, and the Pearson correlation coefficients among them are all less than 0.6. In the 6361×15 original matrix, 5068 genes have no missing entry, $p \approx 0.0150$; 806 genes have one missing entry, $p \approx 0.0097$; 185 genes have two, $p \approx 0.0188$; 63 genes have three, $p \approx 0.0318$. We use the complete 5068×15 matrix, and p of 0.01, 0.02, 0.03, 0.04 in the simulations.

The simulation method is: take a complete matrix; independently mark the entries as missing with probability p; apply the imputation methods to obtain the imputed matrices; compare the imputed matrices to the original one; compare the clustering of imputed data to that of the original data. This procedure is performed 100 times for each missing probability. One evaluation metric is the RMSE: the root mean squared difference between the original values and the imputed values of the missing entries, divided by the root mean squared original values of the missing entries. The other evaluation metric is the number of mis-clustered genes between the k-means clusterings of the original matrix and the imputed one. The value of K in k-means is determined by the sub-sampling algorithm in Ben-Hur *et al*. (2002) and the statistic B of Fowlkes and Mallows (1983), although Ben-Hur *et al*. used hierarchical clustering and we use k-means.

Table 1. Pearson correlation coefficients of expression data among five time points of yeast under amino acid starvation (correlation coefficients greater than 0.6 are boldfaced)

Time	0.5 h	1 h	2 h	4 h	6 h
0.5 h	**1.000**	**0.647**	0.353	0.342	0.413
1 h	**0.647**	**1.000**	0.575	0.408	0.445
2 h	0.353	0.575	**1.000**	0.497	0.435
4 h	0.342	0.408	0.497	**1.000**	**0.694**
6 h	0.413	0.445	0.435	**0.694**	**1.000**

3 RESULTS

The cell cycle data are represented by a 3222×80 matrix. For missing probability p equal to 0.003, 0.005, 0.007, 0.009, the expected numbers of incomplete rows are 688, 1064, 1385, 1659. The stress data are represented by a 5068×15 matrix. For p equal to 0.01, 0.02, 0.03, 0.04, the expected numbers of incomplete rows are 709, 1325, 1859, 2321. An incomplete row may have more than one missing entry. As an example, for the cell cycle data with p equal to 0.009, the expected numbers of rows with 1, 2, 3, 4 missing entries are 1136, 407, 96, 26.

KNNimpute requires the value of K, the number of nearest neighbors used in imputation. Figure 2 contains the plots of average RMSEs of 100 randomized runs. The values of K are set at 8 and 16 for cell cycle and stress data, respectively. SVDimpute requires the value of K, the number of vectors in V used in imputation. Figure 3 contains the plots of average RMSE. The values of K are set at 12 and 2 for cell cycle and stress data, respectively. GMCimpute requires the value of S: $1, \ldots, S$-component mixtures are used in imputation. Figure 4 contains the plots of average RMSE. For cell cycle data, the values of S are set at 5, 3, 1, and 1 for missing probabilities 0.003, 0.005, 0.007, 0.009. For stress data, the value of S is set at 7 for all missing probabilities.

The simulations compare six imputation methods by two evaluation metrics. The means and standard deviations of the

[1] http://rana.lbl.gov/EisenData.htm

[2] Constant 0.32 mM H_2O_2 (80 min) redo; 1 mM menadione (50 min) redo; DTT (30 min); DTT (120 min); 1.5 mM diamide (10 min); 1M sorbitol (15 min); hypo-osmotic shock (15 min); amino acid starvation (1 h); amino acid starvation (6 h); nitrogen depletion (30 min); nitrogen depletion (12 h); YPD 25°C (4 h); YP fructose versus reference pool; 21°C growth; and DBY msn2msn4 0.32 mM H_2O_2 (20 min).

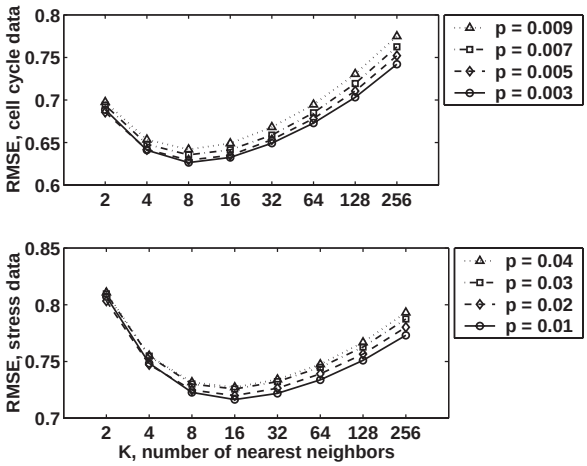

Fig. 2. Values of RMSE by KNNimpute; top: yeast cell cycle data; bottom: yeast environmental stress data.

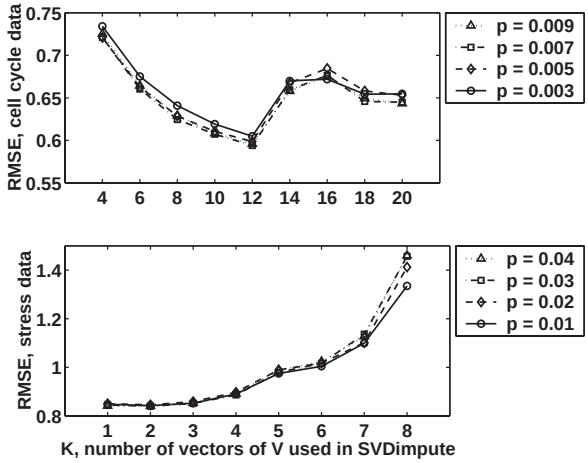

Fig. 3. Values of RMSE by SVDimpute; top: yeast cell cycle data; bottom: yeast environmental stress data.

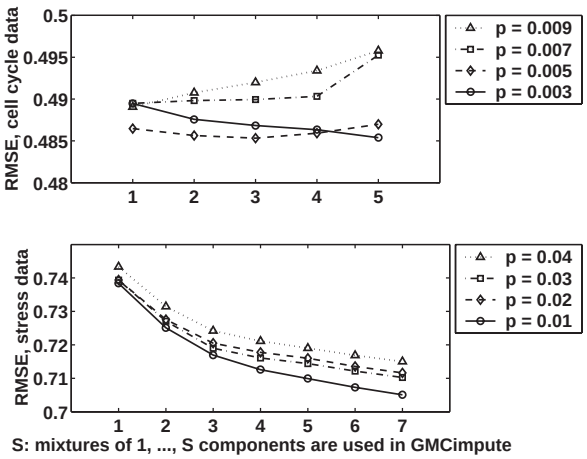

Fig. 4. Values of RMSE by GMCimpute; top: yeast cell cycle data; bottom: yeast environmental stress data.

Table 2. Comparison of RMSE of six imputation methods (the entries are mean/std of 100 randomized runs)

p	0.003	0.005	0.007	0.009
Cell cycle data				
gmc	0.48/0.03	0.48/0.02	0.48/0.02	0.49/0.02
knn	0.62/0.03	0.63/0.02	0.63/0.02	0.64/0.02
svd	0.59/0.04	0.59/0.03	0.59/0.02	0.60/0.02
col	0.96/0.01	0.96/0.01	0.96/0.01	0.96/0.01
row	0.97/0.01	0.97/0.01	0.97/0.01	0.97/0.01
0	1.00/0.00	1.00/0.00	1.00/0.00	1.00/0.00
p	0.01	0.02	0.03	0.04
Stress data				
gmc	0.70/0.03	0.71/0.02	0.71/0.02	0.72/0.02
knn	0.72/0.03	0.72/0.02	0.73/0.02	0.73/0.01
svd	0.84/0.04	0.84/0.03	0.84/0.02	0.85/0.02
col	0.96/0.02	0.96/0.02	0.96/0.01	0.96/0.01
row	1.00/0.00	1.00/0.00	1.00/0.00	1.00/0.00
0	1.00/0.00	1.00/0.00	1.00/0.00	1.00/0.00

Table 3. Comparison of numbers of mis-clustered genes of six imputation methods (the entries are mean/std of 100 randomized runs)

p	0.003	0.005	0.007	0.009
Cell cycle data				
gmc	3.8/3.4	5.0/4.7	5.7/4.6	7.5/5.5
knn	4.4/3.5	5.8/3.9	8.3/5.4	10.0/5.6
svd	4.3/3.9	5.6/4.0	7.9/4.8	8.9/5.8
col	7.9/5.8	9.8/5.2	13.7/6.5	17.2/7.3
row	7.4/5.1	10.8/6.1	14.6/6.5	18.1/8.1
0	8.0/5.6	10.5/5.4	15.8/7.5	18.4/8.4
p	0.01	0.02	0.03	0.04
Stress data				
gmc	44/14	75/17	97/17	124/19
knn	46/14	82/21	100/19	132/27
svd	49/14	85/27	111/20	142/29
col	60/17	95/15	128/19	163/21
row	59/22	93/19	126/21	160/25
0	57/12	93/18	125/18	162/50

first metric, RMSE, are listed in Table 2. The second metric requires the number of clusters in the data. We find there are three and four clusters in cell cycle and stress data, respectively, by sub-sampling (sub-sampled statistics not shown). The means and standard deviations of the second metric, the number of mis-clustered genes, are listed in Table 3.

GMCimpute, KNNimpute and SVDimpute are clearly superior to the other imputation methods. It seems that GMCimpute is the best among the three methods for both data sets, and SVDimpute is better than KNNimpute on cell cycle data, while KNNimpute is better than SVDimpute on stress data. All these observations have P values less than 0.05 by the paired t-tests. In fact, most of the P values are much less than 0.05.

4 DISCUSSION

Troyanskaya *et al*. (2001) was the first study of microarray data imputation. They used a different definition of RMSE than this work. Both studies use the same numerator: the root mean squared difference between the true values and the imputed values of the missing entries, but differ in the denominator. They used the mean true values of the complete matrix, while we use the root mean squared true values of the missing entries as the denominator. Thus the RMSEs in these two studies are not directly comparable. The advantage of our definition is that the RMSE of ZEROimpute is always one, making it easy to compare imputation difficulty across data sets.

The stress data are more difficult for imputation than the cell cycle data. The difference in difficulty is evident in Figures 2–4 and Tables 2 and 3. There are at least two reasons for the difference in difficulty. First, the cell cycle data consist of correlated columns, while the stress data, by our choice, have all uncorrelated columns. Second, the cell cycle data have more columns than the stress data (80 versus 15). Therefore, in practice, as many correlated columns as possible should be used in imputation.

SVD is commonly used in dimension reduction, but it requires a complete matrix. One way to obtain one is to remove incomplete rows. With the cell cycle data, half of the original rows would be removed. Given the smaller RMSE of GMCimpute than SVDimpute, it is worthy of consideration employing GMCimpute to fill in missing entries so as to work with a larger matrix in SVD analysis. The original studies that put microarray data in the public domain usually include cluster analyses, but almost all of them do not explicitly employ imputation. Depending on the similarity measure used (such as Pearson correlation coefficient) and details of implementations, the implicit operations done for missing entries often correspond to ROWimpute, COLimpute or ZEROimpute. The findings of the present work indicate that published *k*-means clustering results can be improved by applying GMCimpute prior to clustering. Note that the goal of imputation is not to improve clustering, but to provide unbiased estimates that would prevent biased clustering.

KNNimpute uses local information in imputation, with less than 50 genes involved in imputing one gene. SVDimpute uses global patterns that come from all genes on the arrays. GMCimpute uses information in the intermediate structures, i.e. the clusters, that consist of hundreds of genes per cluster. Gaussian mixture clustering is but one clustering method. With some augmentation, it is likely that other clustering or classification methods can be used for imputation too.

ACKNOWLEDGEMENTS

We would like to thank the two anonymous referees whose comments helped improve the presentation. M.O. is partially supported by the US EPA funded Center for Exposure and Risk Modeling (CERM) at EOHSI (EPAR-827033). The authors would also like to acknowledge support from the NIH-NLM for an Integrated Advanced Information Management Systems (IAIMS, grant no. 2 G08 LM06230-03AI); the New Jersey Commission on Higher Education for a High-Technology Workforce Excellence (grant no. 801020-09); and additional support was provided by NIEHS (grant no. ES0522). M.O. and W.J.W. wish to thank Dr Don Delker, Dr David Dix, Dr Edward Karoly, Dr Ann Richard, Dr John Rockett and Dr Douglas Wolf at EPA-NHEERL for stimulating discussions.

REFERENCES

Banfield,J.D. and Raftery,A.E. (1993) Model-based Gaussian and non-Gaussian clustering. *Biometrics*, **49**, 803–821.

Bar-Joseph,Z., Gerber,G., Gifford,D.K., Jaakkola,T.S. and Simon,I. (2002) A new approach to analyzing gene expression time series data, Sixth Annual International Conference on Research in Computational Molecular Biology, Washington, DC.

Ben-Hur,A., Elisseeff,A. and Guyon,I. (2002) A stability based method for discovering structure in clustered data. *Pac. Symp. Biocomput.*, 6–17.

Bradley,P. and Fayyad,U. (1998) Refining initial points for *k*-means clustering, 15th International Conference on Machine Learning, Madison, WI.

Celeux,G. and Govaert,G. (1992) A classification EM algorithm for clustering and two stochastic versions. *Comput. Statist. Data Anal.*, **14**, 315–332.

Celeux,G. and Govaert,G. (1995) Gaussian parsimonious clustering models. *Pattern Recognition*, **28**, 781–793.

Dempster,A.P., Laird,N.M. and Rubin,D.B. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Stat. Soc. B*, **39**, 1–38.

Eisen,M., Spellman,P., Brown,P. and Botstein,D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci., USA*, **95**, 14863–14868.

Fowlkes,E.B. and Mallows,C.L. (1983) A method for comparing two hierarchical clusterings. *J. Am. Stat. Assoc.*, **78**, 553–569.

Gasch,A., Spellman,P., Kao,C., Carmel-Harel,O., Eisen,M., Storz,G., Botstein,D. and Brown,P. (2000) Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell.*, **11**, 4241–4257.

Ghosh,D. and Chinnaiyan,A. (2002) Mixture modelling of gene expression data from microarray experiments. *Bioinformatics*, **18**, 275–286.

Hartigan,J.A. (1975) *Clustering Algorithms*. Wiley, New York.

Holter,N., Mitra,M., Maritan,A., Cieplak,M., Banavar,J. and Fedoroff,N. (2000) Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proc. Natl Acad. Sci., USA*, **97**, 8409–8414.

Kass,R.E. and Raftery,A.E. (1995) Bayes factors. *J. Am. Stat. Assoc.*, **90**, 773–795.

Kaufman,L. and Rousseeuw,P. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.

Levine,E. and Domany,E. (2001) Resampling method for unsupervised estimation of cluster validity. *Neural Comput.*, **13**, 2573–2593.

Schwarz,G. (1978) Estimating the dimension of a model. *Ann. Stat.*, **6**, 461–464.

Sherlock,G., Hernandez-Boussard,T., Kasarskis,A., Binkley,G., Matese,J., Dwight,S., Kaloper,M., Weng,S., Jin,H., Ball,C. *et al*. (2001) The Stanford microarray database. *Nucleic Acids Res.*, **29**, 152–155.

Tibshirani,R., Walther,G. and Hastie,T. (2001) Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. B*, **63**, 411–423.

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

Watkins,D.S. (1991) *Fundamentals of Matrix Computations*. Wiley, New York.

Yeung,K., Fraley,C., Murua,A., Raftery,A. and Ruzzo,W. (2001a) Model-based clustering and data transformations for gene expression data. *Bioinformatics*, **17**, 977–987.

Yeung,K., Haynor,D. and Ruzzo,W. (2001b) Validating clustering for gene expression data. *Bioinformatics*, **17**, 309–318.