BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 2 2005, pages 187–198
doi:10.1093/bioinformatics/bth499

Missing value estimation for DNA microarray gene expression data: local least squares imputation

Hyunsoo Kim[1], *Gene H. Golub*[2] *and Haesun Park*[1,3,*]

[1]Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 200 Union Street S.E., Minneapolis, MN 55455, USA, [2]Computer Science Department, Stanford University, Gates Building 2B #280, Stanford, CA 94305-9025, USA and [3]The National Science Foundation, 4201 Wilson Boulevard, Arlington, VA 22230, USA

Received on May 12, 2004; revised on August 6, 2004; accepted on August 18, 2004
Advance Access publication August 27, 2004

ABSTRACT

Motivation: Gene expression data often contain missing expression values. Effective missing value estimation methods are needed since many algorithms for gene expression data analysis require a complete matrix of gene array values. In this paper, imputation methods based on the least squares formulation are proposed to estimate missing values in the gene expression data, which exploit local similarity structures in the data as well as least squares optimization process.

Results: The proposed local least squares imputation method (LLSimpute) represents a target gene that has missing values as a linear combination of similar genes. The similar genes are chosen by k-nearest neighbors or k coherent genes that have large absolute values of Pearson correlation coefficients. Non-parametric missing values estimation method of LLSimpute are designed by introducing an automatic k-value estimator. In our experiments, the proposed LLSimpute method shows competitive results when compared with other imputation methods for missing value estimation on various datasets and percentages of missing values in the data.

Availability: The software is available at http://www.cs.umn.edu/˜hskim/tools.html

Contact: hpark@cs.umn.edu

1 INTRODUCTION

Microarray data analysis has been successfully applied in a number of studies over a broad range of biological disciplines including cancer classification by class discovery and prediction (Golub *et al.*, 1999), identification of the unknown effects of a specific therapy (Perou *et al.*, 2000), identification of genes relevant to a certain diagnosis or therapy (Cho *et al.*, 2003) and cancer prognosis (Shipp *et al.*, 2002; van't Veer *et al.*, 2002). Since multivariate supervised classification methods such as support vector machines (SVMs) (Vapnik, 1995), and multivariate statistical analysis methods such as principal component analysis (PCA), singular value decomposition (SVD) (Golub and van Loan, 1996; Alter *et al.*, 2000) and generalized SVD (GSVD) (Golub and van Loan, 1996; Alter *et al.*, 2003) cannot be applied to data with missing values, the missing value estimation is an important preprocessing step. Gene expression data sets often contain missing values due to various reasons, e.g. insufficient resolution, image corruption, dust or scratches on the slides or experimental error during the laboratory process. Since it is often very costly or time consuming to repeat the experiment, many algorithms have been developed to recover the missing values (Troyanskaya *et al.*, 2001; Oba *et al.*, 2003; Friedland *et al.*, 2003). Moreover, estimating unknown elements in the given matrix has many potential applications in the other fields. There are several approaches for estimating the missing values. Recently, for missing value estimation, the SVD-based method (SVDimpute) and weighted k-nearest neighbors imputation (KNNimpute) have been introduced (Troyanskaya *et al.*, 2001). It has been shown that KNNimpute performs better on non-time series data or noisy time series data, while SVDimpute works well on time series data with low noise levels. Overall, the weighted k-nearest neighbor based imputation provides a more robust method for missing value estimation than the SVD-based method (Troyanskaya *et al.*, 2001).

Throughout the paper, we will use $G \in \mathbb{R}^{m\times n}$ to denote a gene expression data matrix with m genes and n experiments, and assume $m \gg n$. In the matrix G, a row $\mathbf{g}_i^{\mathrm{T}} \in \mathbb{R}^{1\times n}$ represents expressions of the i-th gene in n experiments:

$$G = \begin{pmatrix} \mathbf{g}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{g}_m^{\mathrm{T}} \end{pmatrix} \in \mathbb{R}^{m\times n}.$$

[*]To whom correspondence should be addressed.

A missing value in the l-th location of the i-th gene is denoted as α, i.e.

$$G(i,l) = \mathbf{g}_i(l) = \alpha.$$

For simplicity of algorithm description, all missing value estimation algorithms mentioned in this paper are described first assuming there is a missing value in the first position of the first gene, i.e.

$$G(1,1) = \mathbf{g}_1(1) = \alpha,$$

then the general algorithms for our proposed missing value estimation methods for DNA microarray expression data are introduced.

The KNNimpute method (Troyanskaya *et al.*, 2001) finds k ($k < m$) other genes with expressions most similar to that of $\mathbf{g}_1$ and with the values in their first positions not missing. The missing value of $\mathbf{g}_1$ is estimated by the weighted average of values in the first positions of these k closest genes. For the weighted average, the contribution of each gene is weighted by the similarity of its expression to that of $\mathbf{g}_1$. In the SVDimpute method (Troyanskaya *et al.*, 2001), the SVD of the matrix G', which is obtained after all missing values of the G are substituted by zero or row averages, is computed. Then, using the t most significant eigengenes (Alter *et al.*, 2000) of G', where the specific value of t is either predetermined or determined based on datasets, a missing value α in $\mathbf{g}_1$ is estimated by regressing this gene against the t most significant eigengenes. Using the coefficients of the regression, the missing value is estimated as a linear combination of the values in the first position of t eigengenes. When determining these regression coefficients, the missing value $\mathbf{g}_1(1)$ of $\mathbf{g}_1$ and the first values of the t eigengenes are not used. The above procedure is repeated until the total change of the matrix becomes insignificant. The computational complexity of SVDimpute is $O(n^2mj)$, where j is the number of iterations performed before the threshold value is reached. SVDimpute is useful for time series data with low noise level. Recently, Bayesian PCA (BPCA), which simultaneously estimates a probabilistic model and latent variables within the framework of Bayesian inference, has been successfully applied to missing value estimation problems (Oba *et al.*, 2003). Also, a fixed rank approximation algorithm (FRAA) (Friedland *et al.*, 2003) using the SVD has been proposed. However, FRAA could not outperform KNNimpute even though it is more accurate than replacing missing values with 0's or with row means. More recently, Bø *et al.* (2004) has introduced a missing value estimation method based on the least squares principle, which utilizes correlations between both genes and arrays, referred to as LSimpute.

In this paper, we introduce novel least squares based imputation methods, where a target gene that has missing values is represented as a linear combination of similar genes. Rather than using all available genes in the data, since only similar genes based on a similarity measure are used, our method is referred to as local least squares imputation (LLSimpute). As similarity measures, both L_2-norm and Pearson correlation coefficients are investigated for comparison. We evaluate all proposed imputation methods on the five datasets and compare them with KNNimpute and an estimation method based on BPCA for various percentages of missing values.

2 METHODS

There are two steps in the local least squares imputation. The first step is to select k genes by the L_2-norm or by Pearson correlation coefficients. The second step is regression and estimation, regardless of how the k genes are selected. A heuristic k parameter selection method is described in the Results and discussion section.

2.1 Selecting genes

To recover a missing value α in the first location $\mathbf{g}_1(1)$ of $\mathbf{g}_1$ in $G \in \mathbb{R}^{m\times n}$, the k-nearest neighbor gene vectors for $\mathbf{g}_1$,

$$\mathbf{g}_{s_i}^{\mathrm{T}} \in \mathbb{R}^{1\times n}, \quad 1 \le i \le k,$$

are found for LLSimpute based on the L_2-norm (LLSimpute/L2). In this process of finding the similar genes, the first component of each gene is ignored following the fact that $\mathbf{g}_1(1)$ is missing.

The LLSimpute based on the Pearson correlation coefficient (Pearson, 1894), referred to as LLSimpute/PC, takes advantage of the coherent genes. When there is a missing value in the first location of $\mathbf{g}_1$, the Pearson correlation coefficient r_{1j} between two vectors $\mathbf{g}_1' = (g_{12}, \ldots, g_{1n})^{\mathrm{T}}$ and $\mathbf{g}_j' = (g_{j2}, \ldots, g_{jn})^{\mathrm{T}}$ is defined as

$$r_{1j} = \frac{1}{(n-1)} \sum_{k=2}^{n} \left(\frac{g_{1k} - \bar{g}_1}{\sigma_1}\right)\left(\frac{g_{jk} - \bar{g}_j}{\sigma_j}\right), \qquad (1)$$

where $\bar{g}_j$ is the average of values in $\mathbf{g}_j'$ and σ_j is the SD of these values. The components of $\mathbf{g}_1$ that correspond to missing values are not considered in computing the coefficients. We used the absolute values of the Pearson correlation coefficients since the highly correlated but opposite signed components of the genes, i.e $r \simeq -1.0$, are also helpful in estimating missing values. In LLSimpute/PC, missing values in the target genes are estimated by local least squares where highly correlated genes in the microarray data are selected based on the Pearson correlation coefficients. First, all Pearson correlation coefficients between $\mathbf{g}_1$ and the other genes are computed. Then, to recover a missing value in the first location of $\mathbf{g}_1$, $G(1,1) = \mathbf{g}_1(1) = \alpha$, the k genes with the largest Pearson correlation coefficients in magnitude,

$$\mathbf{g}_{s_i}^{\mathrm{T}} \in \mathbb{R}^{1\times n}, \quad 1 \le i \le k,$$

are found for LLSimpute/PC.

2.2 Gene-wise formulation of local least squares imputation

As imputation can be performed regardless of how the k-genes are selected, we present only the imputation based on L_2-norm for simplicity. Based on these k-neighboring gene vectors, the matrix $A \in \mathbb{R}^{k\times(n-1)}$ and the two vectors $\mathbf{b} \in \mathbb{R}^{k\times 1}$ and $\mathbf{w} \in \mathbb{R}^{(n-1)\times 1}$ are formed. The k rows of the matrix A consist of the k-nearest neighbor genes $\mathbf{g}_{s_i}^{\mathrm{T}} \in \mathbb{R}^{1\times n}$, $1 \le i \le k$, with their first values deleted, the elements of the vector $\mathbf{b}$ consists of the first components of the k vectors $\mathbf{g}_{s_i}^{\mathrm{T}}$, and the elements of the vector $\mathbf{w}$ are the $n-1$ elements of the gene vector $\mathbf{g}_1$ whose missing first item is deleted. After the matrix A, and the vectors $\mathbf{b}$ and $\mathbf{w}$ are formed, the least squares problem is formulated as

$$\min_{\mathbf{x}} \|A^{\mathrm{T}}\mathbf{x} - \mathbf{w}\|_2. \tag{2}$$

Then, the missing value α is estimated as a linear combination of first values of genes

$$\alpha = \mathbf{b}^{\mathrm{T}}\mathbf{x} = \mathbf{b}^{\mathrm{T}}(A^{\mathrm{T}})^{\dagger}\mathbf{w}, \tag{3}$$

where $(A^{\mathrm{T}})^{\dagger}$ is the pseudoinverse of A^{T}.

For example, assume that the target gene $\mathbf{g}_1$ has a missing value in the first position among the total of six experiments. If the missing value is to be estimated by the k similar genes, the matrix A, and vectors $\mathbf{b}$ and $\mathbf{w}$ are constructed as

$$\begin{pmatrix} \mathbf{g}_1^{\mathrm{T}} \\ \mathbf{g}_{s_1}^{\mathrm{T}} \\ \vdots \\ \mathbf{g}_{s_k}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \alpha & \mathbf{w}^{\mathrm{T}} \\ \mathbf{b} & A \end{pmatrix}$$

$$= \begin{pmatrix} \alpha & \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 & \mathbf{w}_4 & \mathbf{w}_5 \\ \mathbf{b}_1 & A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{b}_k & A_{k,1} & A_{k,2} & A_{k,3} & A_{k,4} & A_{k,5} \end{pmatrix},$$

where α is the missing value and $\mathbf{g}_{s_1}^{\mathrm{T}}, \ldots, \mathbf{g}_{s_k}^{\mathrm{T}}$ are genes similar to $\mathbf{g}_1^{\mathrm{T}}$. From the second to the last components of the neighbor genes, $\mathbf{a}_i^{\mathrm{T}}$, $1 \le i \le k$, form the i-th row vector of the matrix A. The vector $\mathbf{w}$ of the known elements of target gene $\mathbf{g}_1$ can be represented as a linear combination

$$\mathbf{w} \simeq \mathbf{x}_1\mathbf{a}_1 + \mathbf{x}_2\mathbf{a}_2 + \cdots + \mathbf{x}_k\mathbf{a}_k,$$

where $\mathbf{x}_i$ are the coefficients of the linear combination, found from the least squares formulation (2). Accordingly, the missing value α in $\mathbf{g}_1$ can be estimated by

$$\alpha = \mathbf{b}^{\mathrm{T}}\mathbf{x} = \mathbf{b}_1\mathbf{x}_1 + \mathbf{b}_2\mathbf{x}_2 + \cdots + \mathbf{b}_k\mathbf{x}_k.$$

Now, we deal with the case in which there are more than one missing values in a gene vector. In this case, to recover the total of q missing values in any locations of the gene $\mathbf{g}_1$, first, the k-nearest neighbor gene vectors for $\mathbf{g}_1$,

$$\mathbf{g}_{s_i}^{\mathrm{T}} \in \mathbb{R}^{1\times n}, \quad 1 \le i \le k,$$

are found. In this process of finding the similar genes, the q components of each gene at the q locations of missing values in $\mathbf{g}_1$ are ignored. Then, based on these k neighboring gene vectors, a matrix $A \in \mathbb{R}^{k\times(n-q)}$ a matrix $B \in \mathbb{R}^{k\times q}$ and a vector $\mathbf{w} \in \mathbb{R}^{(n-q)\times 1}$ are formed. The i-th row vector $\mathbf{a}_i^{\mathrm{T}}$ of the matrix A consists of the i-th nearest neighbor genes $\mathbf{g}_{s_i}^{\mathrm{T}} \in \mathbb{R}^{1\times n}$, $1 \le i \le k$, with its elements at the q missing locations of missing values of $\mathbf{g}_1$ excluded. Each column vector of the matrix B consists of the values of the j-th location of the missing values ($1 \le j \le q$) of the k vectors $\mathbf{g}_{s_i}^{\mathrm{T}}$. The elements of the vector $\mathbf{w}$ are the $n-q$ elements of the gene vector $\mathbf{g}$ whose missing items are deleted. After the matrices A and B and a vector $\mathbf{w}$ are formed, the least squares problem is formulated as

$$\min_{\mathbf{x}} \|A^{\mathrm{T}}\mathbf{x} - \mathbf{w}\|_2. \tag{4}$$

Then, the vector $\mathbf{u} = (\alpha_1, \alpha_2, \cdots, \alpha_q)^{\mathrm{T}}$ of q missing values can be estimated as

$$\mathbf{u} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_q \end{pmatrix} = B^{\mathrm{T}}\mathbf{x} = B^{\mathrm{T}}(A^{\mathrm{T}})^{\dagger}\mathbf{w}, \tag{5}$$

where $(A^{\mathrm{T}})^{\dagger}$ is the pseudoinverse of A^{T}.

For example, assume that the target gene $\mathbf{g}_1$ has two missing values in the 1st and the 6th positions among the total six experiments. If the missing value is to be estimated by the k similar genes, each element of the matrix A and B, and a vector $\mathbf{w}$ are constructed as

$$\begin{pmatrix} \mathbf{g}_1^{\mathrm{T}} \\ \mathbf{g}_{s_1}^{\mathrm{T}} \\ \vdots \\ \mathbf{g}_{s_k}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \alpha_1 & \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 & \mathbf{w}_4 & \alpha_2 \\ B_{1,1} & A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & B_{1,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ B_{k,1} & A_{k,1} & A_{k,2} & A_{k,3} & A_{k,4} & B_{k,2} \end{pmatrix},$$

where α_1 and α_2 are the missing values and $\mathbf{g}_{s_1}^{\mathrm{T}}, \ldots, \mathbf{g}_{s_k}^{\mathrm{T}}$ are the k genes that are most similar to $\mathbf{g}_1$. The known elements of $\mathbf{w}$ can be represented by

$$\mathbf{w} \simeq \mathbf{x}_1\mathbf{a}_1 + \mathbf{x}_2\mathbf{a}_2 + \cdots + \mathbf{x}_k\mathbf{a}_k,$$

where $\mathbf{x}_i$ are the coefficients of the linear combination, found from the least squares formulation (4). And, the missing values in $\mathbf{g}_1$ can be estimated by

$$\alpha_1 = B_{1,1}\mathbf{x}_1 + B_{2,1}\mathbf{x}_2 + \cdots + B_{k,1}\mathbf{x}_k,$$
$$\alpha_2 = B_{1,2}\mathbf{x}_1 + B_{2,2}\mathbf{x}_2 + \cdots + B_{k,2}\mathbf{x}_k,$$

where α_1 and α_2 are the first and second missing values in the target gene.

2.3 Experiment-wise formulation of local least squares imputation

In this subsection, we introduce another possible least squares formulation and illustrate relationship between these two least squares formulations. Based on the matrix A and the vector $\mathbf{b}$ presented in the previous subsection, the following least squares problem for missing value estimation can also be formulated:

$$\min_{\mathbf{y}} \|A\mathbf{y} - \mathbf{b}\|_2. \tag{6}$$

Then, the vector $\mathbf{b}$ of the known elements of the experiment that has the missing value α as its first component can be represented as a linear combination of other experiments. Accordingly, the missing value α can be estimated as a linear combination of values of the first components of the experiments by

$$\alpha = \mathbf{w}^{\mathrm{T}}\mathbf{y} = \mathbf{w}^{\mathrm{T}}A^{\dagger}\mathbf{b}, \tag{7}$$

where $A^{\dagger}$ is the pseudoinverse of A. The pseudoinverse $A^{\dagger}$ of A can be computed by

$$\begin{aligned} A^{\dagger} &= [V_1 V_2]\begin{bmatrix} \Sigma_1^{-1} & 0 \\ 0 & 0 \end{bmatrix}[U_1 U_2]^{\mathrm{T}} \\ &= V_1 \Sigma_1^{-1} U_1^{\mathrm{T}} \end{aligned}$$

from the SVD of A,

$$A = [U_1 U_2]\begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}[V_1 V_2]^{\mathrm{T}} = U_1 \Sigma_1 V_1^{\mathrm{T}},$$

where $U_1 \in \mathbb{R}^{k\times r}$, $\Sigma_1 \in \mathbb{R}^{r\times r}$, $V_1 \in \mathbb{R}^{n-1\times r}$ and $r = \mathrm{rank}(\Sigma_1) = \mathrm{rank}(A)$.

Although the gene-wise formulation and the experiment-wise formulation may seem to represent the missing value in two different ways, the solutions are in fact the same due to the following relations:

$$\mathbf{b}^{\mathrm{T}}\mathbf{x} = \mathbf{b}^{\mathrm{T}}(A^{\mathrm{T}})^{\dagger}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}A^{\dagger}\mathbf{b})^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\mathbf{y}. \tag{8}$$

Therefore, we choose an imputation formulation of Equation (2) which represents a gene by a linear combination of the other similar genes.

For the case in which there are more than one missing values in a gene vector, the formulation for multiple missing value estimation analogous to Equation (6) would be

$$\min_{Y} \|AY - B\|_F, \tag{9}$$

where F denotes the Frobenius norm. Then, the missing values $\mathbf{u} = (\alpha_1, \alpha_2, \cdots, \alpha_q)^{\mathrm{T}}$ can be estimated as a linear combination of values of $\mathbf{w}$, i.e.

$$\mathbf{u}^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}Y = \mathbf{w}^{\mathrm{T}}A^{\dagger}B, \tag{10}$$

where $A^{\dagger}$ is the pseudoinverse of A. This vector $\mathbf{u}$ is identical to that in Equation (5) which is obtained based on gene-wise formulation since

$$B^{\mathrm{T}}\mathbf{x} = B^{\mathrm{T}}(A^{\mathrm{T}})^{\dagger}\mathbf{w} = (\mathbf{w}^{\mathrm{T}}A^{\dagger}B)^{\mathrm{T}} = Y^{\mathrm{T}}\mathbf{w}. \tag{11}$$

Therefore, we chose an imputation formulation of Equation (4) which represents a gene by a linear combination of the similar genes.

For estimating each missing value, we need to build the matrices A and B and a vector $\mathbf{w}$, and solve the least squares problem of Equation (4). To take advantage of non-missing entries of neighbor genes which have missing values, each missing value is initially estimated by the gene-wise average. If number of missing entries is much smaller than the number of genes, the neighbor genes that contain missing value are excluded when building the least squares systems. In this case, the matrices A and B do not contain any estimated entries. This process is helpful in achieving more accurate estimation result since it circumvents possible errors generated from pre-estimation by the row-averages.

3 RESULTS AND DISCUSSION

3.1 Datasets

Five microarray datasets have been used in our experiments. The first dataset was obtained from α-factor block release that was studied for the identification of cell-cycle regulated genes in yeast *Saccharomyces cerevisiae* (Spellman *et al.*, 1998). We built a complete data matrix of 4304 genes and 18 experiments (SP.ALPHA) that does not have any missing value to assess missing value estimation methods. The second dataset of a complete matrix of 4304 genes and 14 experiments (SP.ELU) is based on an elutriation dataset (Spellman *et al.*, 1998). The 4304 genes originally had no missing values in the α-factor block release set and the elutriation dataset. The third dataset was from 784 cell-cycle-regulated genes, which were classified by Spellman *et al.* (1998) into five classes, for the same 14 experiments as the second data set. After removing all gene rows that have missing values, we built the third data set of 474 genes and 14 experiments (SP.CYCLE). The fourth dataset is from a study of response to environmental changes in yeast (Gasch *et al.*, 2001). It contains 6361 genes and 156 experiments that have time-series of specific treatments. A complete matrix of 2641 genes and 44 experiments was formed after removing experimental columns that have $>8\%$ missing values and then selecting gene rows that do not have any missing value (GA.ENV). The fifth dataset is the cDNA microarray data relevant to human colorectal cancer (CRC) (Takemasa *et al.*, 2001). This dataset contains 758 genes and 205 primary CRCs that include 127 non-metastatic primary CRCs, 54 metastatic primary CRCs to the liver and 24 metastatic primary CRCs to distant organs exclusive of the liver, and 12 normal colonic epithelia (TA.CRC) (Oba *et al.*, 2003). This is a challenging dataset with multiple experiments with no time course relationships. The SP.ALPHA, SP.ELU and TA.CRC are the same datasets that were used in the study of BPCA (Oba *et al.*, 2003). The SP.CYCLE dataset was designed to test how much an imputing method can take advantage of strongly correlated genes in estimating missing values.

Given an original expression data matrix G_o with m_o-genes × n-experiments from the Stanford Microarray Database (SMD) (Sherlock *et al.*, 2001), we prepared the initial full matrix $G_i \in \mathbb{R}^{m_i \times n}$ where m_i genes have no missing values ($m_i \leq m_o$). If there is no missing value in the original matrix, then the initial matrix G_i is set to G_o. Given an initial expression data matrix G_i, certain percentage of the data elements of G_i are randomly selected and regarded as missing values. The performance of the missing value estimation is evaluated by normalized root mean squared error (NRMSE):

$$\text{NRMSE} = \sqrt{\text{mean}[(\mathbf{y}_{\text{guess}} - \mathbf{y}_{\text{ans}})^2]} \;/\; \text{std}[\mathbf{y}_{\text{ans}}] \qquad (12)$$

where $\mathbf{y}_{\text{guess}}$ and $\mathbf{y}_{\text{ans}}$ are vectors whose elements are the estimated values and the known answer values, respectively, for all missing entries. The mean and the SD are calculated over missing entries in the entire matrix. In KNNimpute, a weighted average of the k-nearest neighbors is used as an estimate for each missing value in the target gene. The similarity between two genes is defined by the reciprocal of the Euclidian distance calculated for non-missing components. Then, the missing entry is estimated as an average weighted by the similarity values.

3.2 A method for selection of the model parameter *k*

The value for k in KNNimpute and the reduced rank in SVDimpute are important model parameters to choose for obtaining high performance. However, there is no theoretical result for determining these parameters optimally. Similarly, we need to determine the number of nearest neighbors for LLSimpute/L2 and the number of coherent genes for LLSimpute/PC. In our experiments, the following heuristic algorithm for estimating parameter k is used.

Assuming that there are q missing values in the target gene $\mathbf{g}$, all the gene vectors are sorted according to their similarity to $\mathbf{g}$,

$$\tilde{\mathbf{g}}_{s_i}^{\mathrm{T}} \in \mathbb{R}^{1\times n}, \quad 1 \leq i \leq m-1,$$

where the gene $\tilde{\mathbf{g}}_{s_1}$ is the most similar gene to $\mathbf{g}$. Then we estimate an artificial missing value in $\mathbf{g}$ using different k-values. The vector $\mathbf{w}$ is defined based on the $n-q$ elements of the target gene vector $\mathbf{g}$ whose missing items are deleted and the matrix B is formed by the values of j-th location of the missing values ($1 \leq j \leq q$) of k similar genes $\tilde{\mathbf{g}}_{s_i}$, $1 \leq i \leq k$. Now, assume that the first position of $\mathbf{w}$ has a missing value and it needs to be estimated by various numbers of similar genes. Let α_{true} denote $\mathbf{w}_1$ which is in fact known. For the least squares formulation, the rows of the matrix A consist of $\tilde{\mathbf{a}}_i(2:n-q)^{\mathrm{T}}$ for $1 \leq i \leq k$ and a vector $\mathbf{w}$ is set to be $\mathbf{w}(2:n-q)$.

For example, assume that the target gene $\mathbf{g}$ has two missing values in the 1st and 6th positions among the total six experiments. Considering $\mathbf{w}_1$ as a missing value which is in fact known, the matrix A and vectors $\mathbf{w} = (\mathbf{w}_2\ \mathbf{w}_3\ \mathbf{w}_4)^{\mathrm{T}}$ and $\mathbf{b} = (\mathbf{b}_1\ \ldots\ \mathbf{b}_k)^{\mathrm{T}}$ are constructed from

$$\begin{pmatrix} \mathbf{g}_1^{\mathrm{T}} \\ \mathbf{g}_{s_1}^{\mathrm{T}} \\ \vdots \\ \mathbf{g}_{s_k}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} \text{miss} & \alpha & \mathbf{w}_2 & \mathbf{w}_3 & \mathbf{w}_4 & \text{miss} \\ B_{1,1} & \mathbf{b}_1 & A_{1,2} & A_{1,3} & A_{1,4} & B_{1,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ B_{k,1} & \mathbf{b}_k & A_{k,2} & A_{k,3} & A_{k,4} & B_{k,2} \end{pmatrix}, \qquad (13)$$

where $\mathbf{g}_{s_1}^{\mathrm{T}}, \ldots, \mathbf{g}_{s_k}^{\mathrm{T}}$ are the k genes similar to $\mathbf{g}_1$. Then, $\mathbf{a}_i^{\mathrm{T}}$ is the i-th row vector of the matrix A, for $1 \leq i \leq k$, and the known elements of $\mathbf{w}$ can be represented by

$$\mathbf{w} \simeq \mathbf{x}_1\mathbf{a}_1 + \mathbf{x}_2\mathbf{a}_2 + \cdots + \mathbf{x}_k\mathbf{a}_k,$$

where $\mathbf{x}_i$ are the coefficients, found from the least squares formulation

$$\min_{\mathbf{x}} \|A^{\mathrm{T}}\mathbf{x} - \mathbf{w}\|_2.$$

Finally, the value $\mathbf{w}_1 = \alpha$ in $\mathbf{g}_1$ can be estimated by

$$\alpha = \mathbf{b}^{\mathrm{T}}\mathbf{x} = \mathbf{b}_1\mathbf{x}_1 + \mathbf{b}_2\mathbf{x}_2 + \cdots + \mathbf{b}_k\mathbf{x}_k,$$

and compared to the actual value $\mathbf{w}_1$. In the above, the k most similar genes are used for estimating entire missing values in the given matrix. Repeating these estimations using several k-values, a k-value that produces the best estimation ability for the artificial missing values can be found. If there are many missing entries per each row, the above process can be performed considering more than one non-missing positions as missing values in order to obtain more reliable k-value.

This procedure decides a number of similar genes that show good performance for estimating missing values using non-missing elements. The k-value depends on the characteristic of the given data matrix. The motivation of this procedures is that the k-value that shows the best performance using known elements of the matrix can be near an optimal k-value. The gene that has missing values is represented as a linear combination of the k similar genes in the least squares formulations. Hence, an optimal k-value is predicted by using the known values in the same gene that has missing values.

We introduce two extended missing value estimation methods, LLSk/L2 and LLSk/PC, which perform LLSimpute/L2 and LLSimpute/PC, respectively, after predicting k-value using the proposed model selection algorithm. Since LLSk/L2 and LLSk/PC automatically determine the only necessary parameter k, they can be classified as non-parametric missing value estimation methods like BPCA.

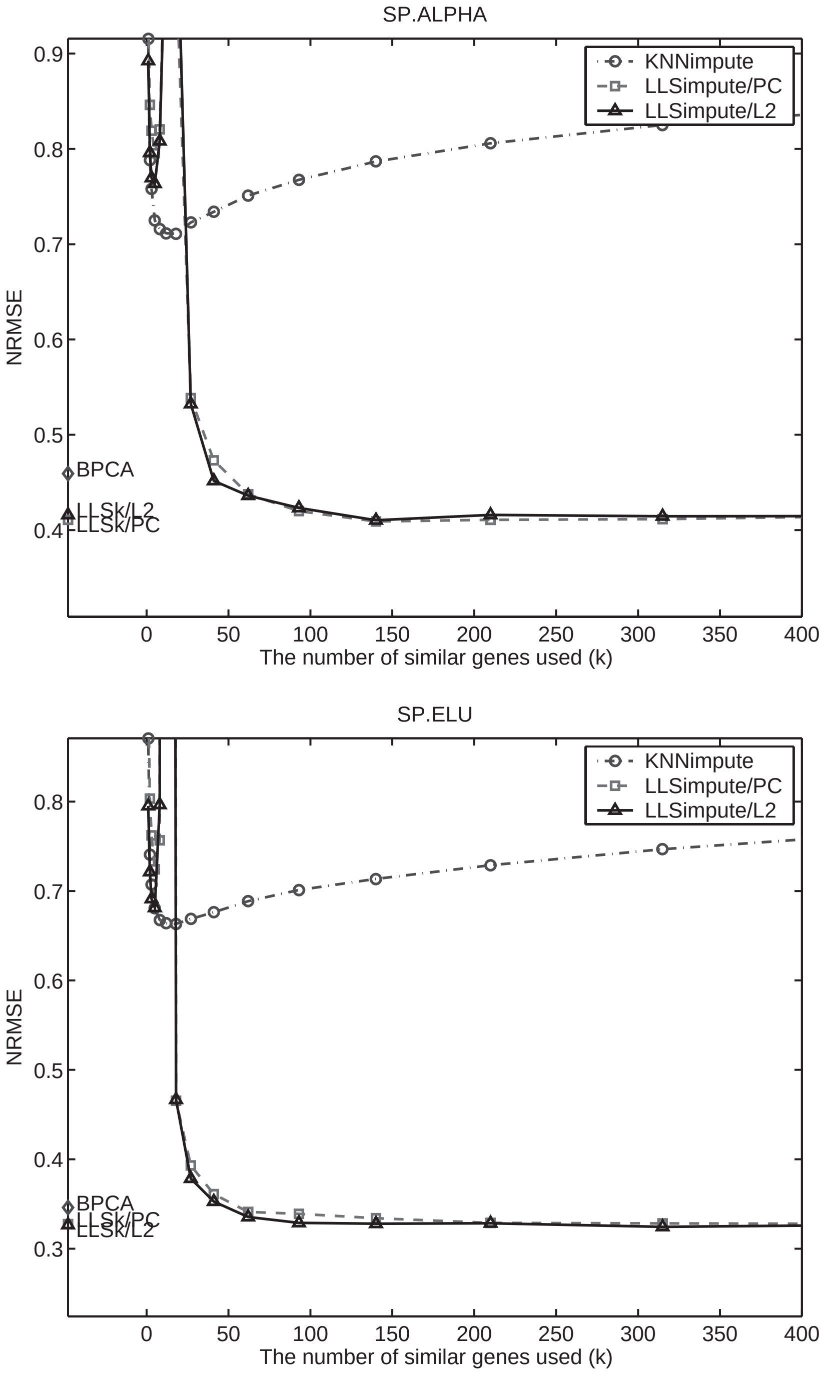

Fig. 1. Comparison of the NRMSEs of various methods and effect of the number of genes for estimating missing values on SP.ALPHA dataset of 4304 genes and 18 experiments and SP.ELU dataset of 4304 genes and 14 experiments. The 5% entries of each dataset were missing. The results of the methods that do not depend on the number of genes are shown on the y-axis.

3.3 Experimental results

In Figure 1, we compared NRMSE of Equation (12) of the missing value estimation methods discussed in this paper. The SP.ALPHA and SP.ELU sets are the same datasets used in the study of BPCA (Oba *et al.*, 2003) and we obtained the same NRMSE values for KNNimpute and BPCA as those presented by Oba *et al.* (2003). The missing value estimation based on BPCA showed good performance on the SP.ELU dataset. However, LLSimpute/L2 and LLSimpute/PC outperformed BPCA as well as KNNimpute when k is large.

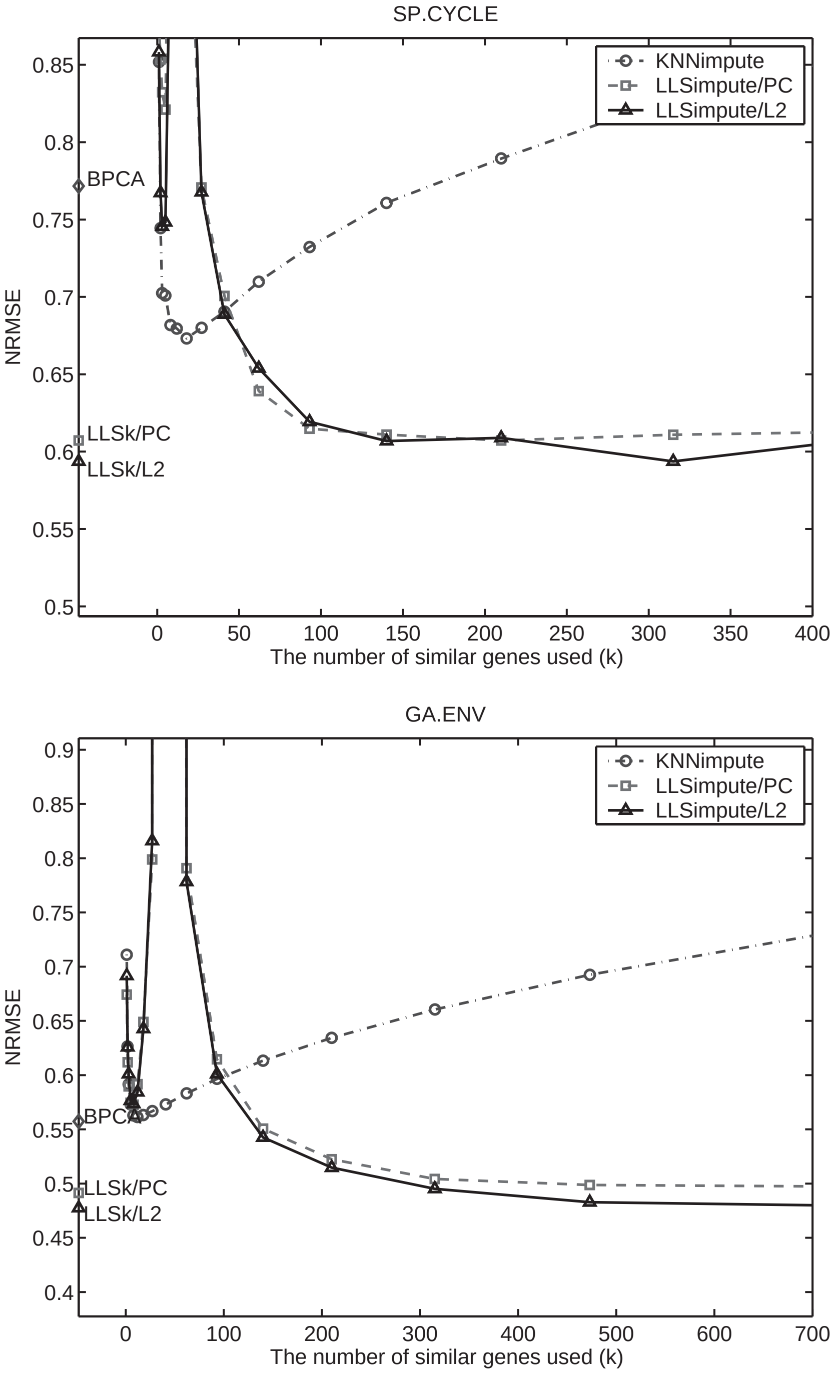

Fig. 2. Comparison of the NRMSEs of various methods and effect of the number of genes for estimating missing values on SP.CYCLE dataset of 474 genes and 14 experiments and GA.ENV dataset of 2641 genes and 44 experiments. The 5% entries of each dataset were missing. The results of the methods that do not depend on the number of genes are shown on the *y*-axis.

In Figure 2, overall, LLSimpute shows better performance as the number of genes increases for estimating missing values on the SP.CYCLE dataset. The NRMSE values of LLSimpute/L2 and BPCA were 0.594 and 0.771, respectively. The SP.CYCLE dataset has significant cluster structures. In this case, LLSimpute showed the best performance among all methods compared, while BPCA showed less accurate results than KNNimpute. The NRMSE values of LLSimpute and BPCA on the GA.ENV dataset were 0.534 and 0.603, respectively. From Figures 1 and 2, we confirmed that the *k*-value

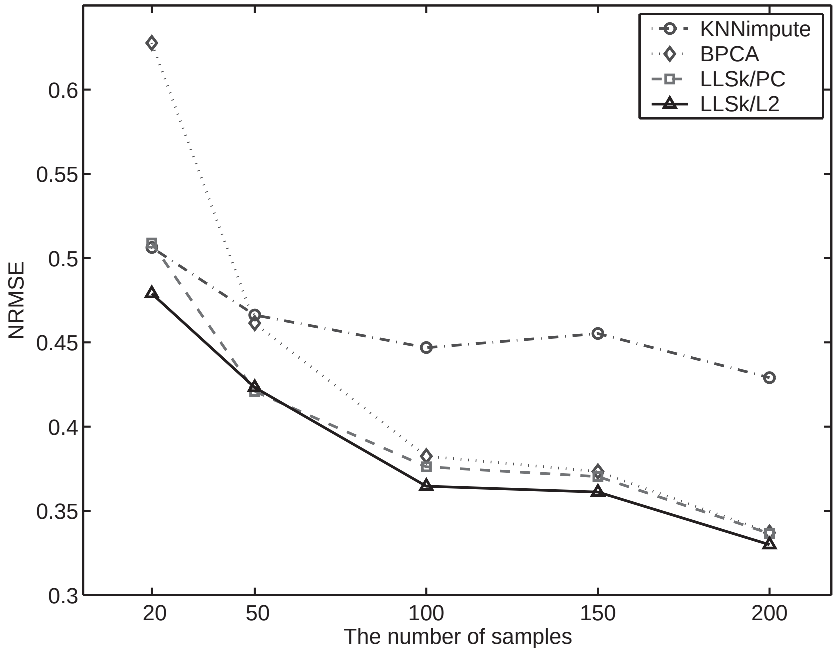

Fig. 3. Comparison of the NRMSEs against number of samples for four methods (KNNimpute, BPCA, LLSk/L2 and LLSk/PC) on TA.CRC dataset.

estimation algorithm is producing appropriate k-values. In Figures 1 and 2, as k increase, the NRMSE of LLSimpute/LS first drops, then rise (but the top portions of the plots are truncated), then drops again and stabilizes. The first drop and rise can be explained by Equation (2). Getting more information from a larger number of genes is responsible for the first drop. As k increase, the matrix A of Equation (2) is getting to be square, i.e. the number of rows of A is the same as the number of columns. Then, the effect of the genes less similar to the target gene become involved in the solution and the performance may become worse. As k keeps increasing, the square structure of A is getting changed to the least squares structure, i.e. the number of rows of A is larger than the number of columns. Then, the minimum norm solution of Equation (2) is the same as the least squares solution of Equation (6) [see Equation (8)]. The second big drop and stabilization can be more easily explained by Equation (6). It shows that considering a sufficient number of genes is helpful in the least squares problem.

To show the dependency of the performance with respect to the number of experiments, the smaller datasets were prepared by clipping a certain number of experiments from TA.CRC dataset. Figure 3 shows the missing value estimation ability for various experiment sizes: 20, 50, 100, 150 and 200. For these experiments, the 5% missing entries were randomly generated from the smaller datasets of TA.CRC. As the number of samples increased the information useful for the imputation increased. The results of KNNimpute were obtained by choosing a k-value that provided the best performance of KNNimpute in each test. The results for LLSk/L2 and LLSk/PC were obtained from k-value estimation algorithm. As reported in the study of BPCA (Oba *et al.*, 2003), the performance of KNNimpute did not improve much as the number of samples increased. When the number of samples was small, KNNimpute exhibited better performance than BPCA. The advantage of KNNimpute for smaller numbers of samples is in using local similarity. The advantage of BPCA for larger numbers of samples is its ability to capture useful information by a Bayesian optimization process. LLSimpute showed the best performance even when the number of sample was small since it uses the local similarity structures and optimization process by the least squares.

Figure 4 shows the results for various percentage (1, 5, 10, 15 and 20%) of missing entries on SP.CYCLE and GA.ENV datasets. For KNNimpute, the number of genes (k) was chosen to be the one that exhibited the best performance in each test. For various percentages of missing entries, LLSimpute showed the best performance consistently. In KNNimpute, the Euclidean distance seems to be an accurate norm since the log-transformation of the data reduces the effect of outliers on gene similarity determination (Troyanskaya *et al.*, 2001). In Figures 3 and 4, we observed the similar result that the Euclidean distance norm is slightly better than the Pearson correlation coefficient for computing gene similarity in the local least squares imputation methods.

To show how the methods respond to higher noise levels, six noisy datasets were prepared based on SP.CYCLE by adding random noise of various levels, with normal distribution.

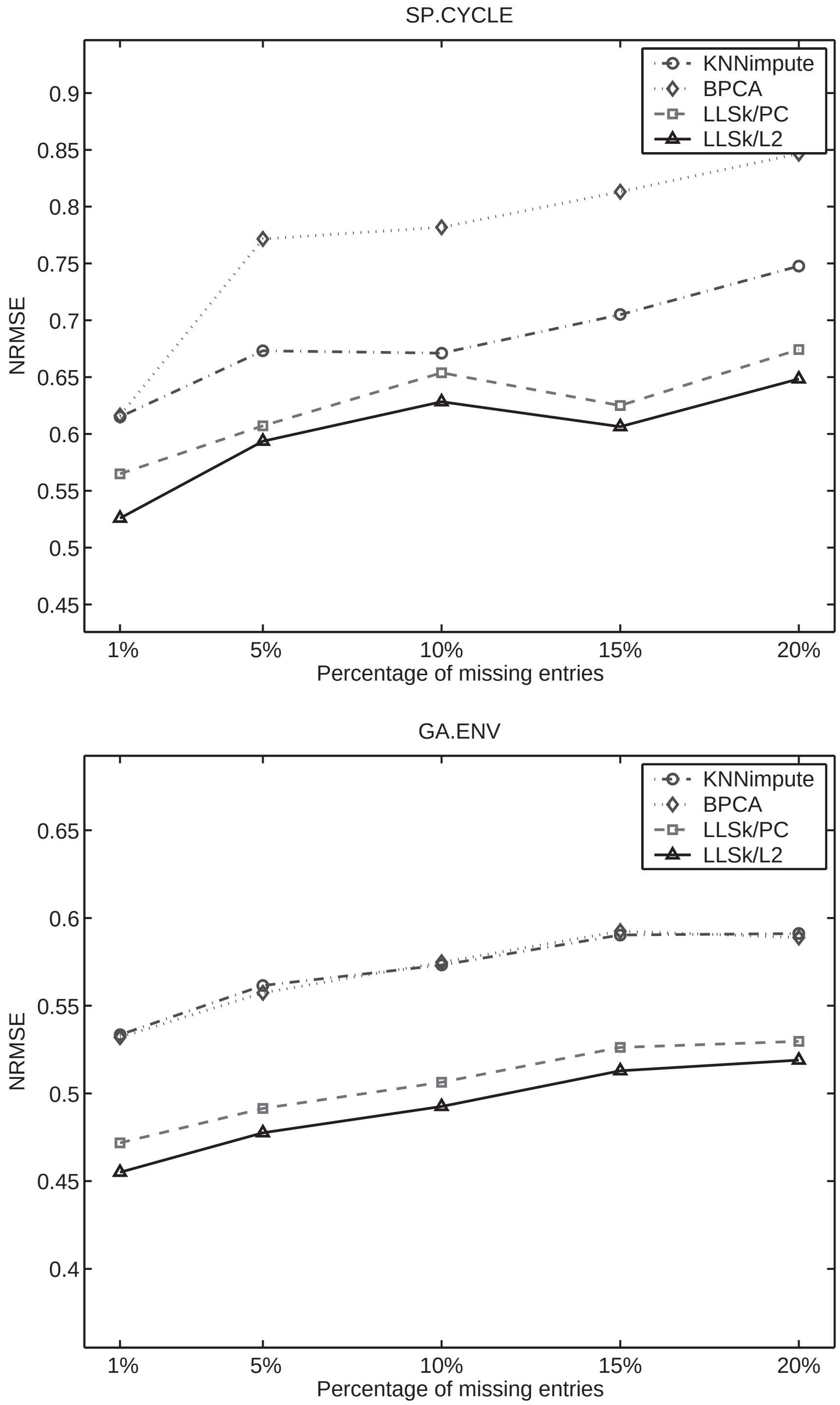

Fig. 4. Comparison of the NRMSEs against percentage of missing entries for four methods (KNNimpute, BPCA, LLSk/L2 and LLSk/PC) on SP.CYCLE and GA.ENV datasets.

After building matrices of random numbers with the normal distribution of mean $\mu = 0$ and various SD (σ = 0.01, 0.05, 0.1, 0.15, 0.2 and 0.25), each noise matrix was added to the data matrix of SP.CYCLE with the 5% of missing values in order to build the six noisy datasets. Figure 5 shows that the performance of BPCA varies relatively largely dependent on the noise level. The performance results of LLSk/L2 and LLSk/PC were less sensitive to the noise level.

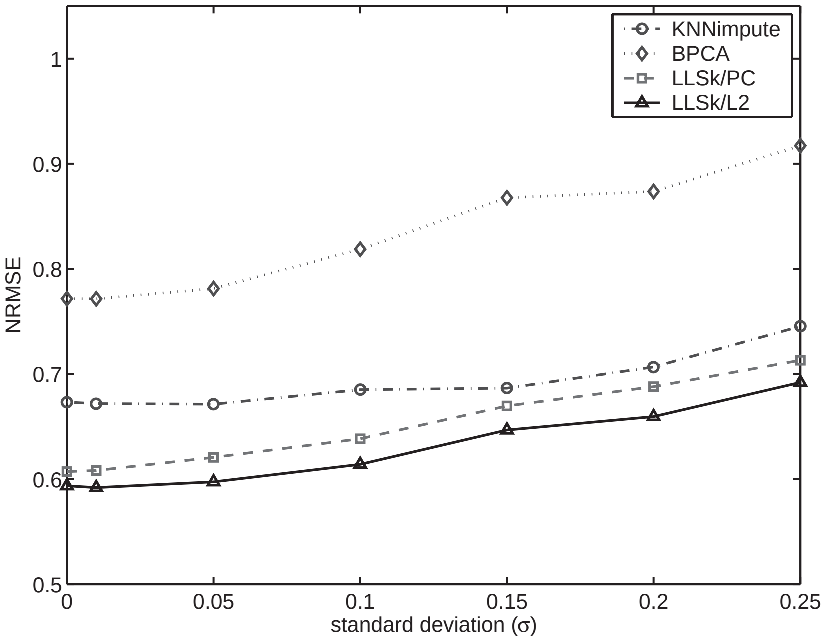

Fig. 5. Comparison of the NRMSEs with respect to noise levels. We added artificial noise with normal distribution of a mean $\mu = 0$ and various SD ($\sigma = 0.01$, 0.05, 0.1, 0.15, 0.2 and 0.25) to SP.CYCLE dataset.

KNNimpute showed robustness against the noise when the SD was less <0.15.

3.4 Comparison to other methods

In KNNimpute (Troyanskaya *et al.*, 2001), after obtaining the k genes $\mathbf{g}_{s_i}$ which are most similar to the target gene $\mathbf{g}$, it estimates the gene vector $\mathbf{g}^*$ by

$$\mathbf{g}^* = \frac{\omega_1 \mathbf{g}_{s_1} + \omega_2 \mathbf{g}_{s_2} + \cdots + \omega_k \mathbf{g}_{s_k}}{\omega_1 + \cdots + \omega_k},$$

where ω_i is the similarity between $\mathbf{g}$ and $\mathbf{g}_{s_i}$. In implementing of KNNimpute, we used

$$\omega_i = 1/\|\mathbf{w} - \mathbf{a}_i\|_2, \qquad (14)$$

where $\mathbf{w}$ and $\mathbf{a}_i$ are the sub-vectors of $\mathbf{g}$ and $\mathbf{g}_{s_i}$, respectively, where the missing components of $\mathbf{g}$ are deleted. If $\mathbf{w} = \mathbf{a}_i$, then a missing value in the target gene $\mathbf{g}$ is estimated in KNNimpute as the value in the corresponding location of the vector $\mathbf{g}_{s_i}$. In LLSimpute, the coefficients of the linear combination of non-missing part of the similar genes $\mathbf{a}_i$, $1 \leq i \leq k$, are optimized by the least squares solution instead of using the similarity measure of Equation (14).

It should be noted that LLSimpute and LSimpute (Bø *et al.*, 2004) use different approaches for imputation, even though both use least squares. According to Figures 1 and 5, the Pearson correlation based method is no better than the L_2-norm based method. The LSimpute (Bø *et al.*, 2004) uses only the Pearson correlation in selecting genes and arrays. Moreover, Bø *et al.* focused on testing only small k values ($k = 5, 10, 15, 20$ and 25) without k-value estimator in order to compare with KNNimpute by using the fact that Troyanskaya *et al.* (2001) reported the best results for k between 10 and 20. However, our experiments indicate that the optimal k-value can be larger than 25 in our least squares formulation.

The BPCA method (Oba *et al.*, 2003) consists of three components: (1) principal component (PC) regression, (2) Bayesian estimation and (3) iterations based on expectation-maximization (EM). In PC regression, the missing part $\mathbf{u}$ in a gene expression vector $\mathbf{g}$ is estimated from the observed part $\mathbf{w}$ by using the principal axis vectors. Let s denote the number of the principal axis vectors. Then, the known elements can be represented by

$$\mathbf{w} \simeq \varsigma_1 \mathbf{p}_1^{\text{obs}} + \varsigma_2 \mathbf{p}_2^{\text{obs}} + \cdots + \varsigma_s \mathbf{p}_s^{\text{obs}},$$

where ς_i are the coefficients of the linear combination and $\mathbf{p}_i^{\text{obs}}$ are the observed parts of principal axis vectors. After obtaining the coefficients by the least squares, the missing part is estimated as

$$\mathbf{u} = \varsigma_1 \mathbf{p}_1^{\text{miss}} + \varsigma_2 \mathbf{p}_2^{\text{miss}} + \cdots + \varsigma_s \mathbf{p}_s^{\text{miss}},$$

where $\mathbf{p}_i^{\text{miss}}$ are the missing parts of principal axis vectors. This is the conceptual process of BPCA even though it takes advantage of the sophisticated Bayesian estimation and EM-line repetitive algorithm. The SVDimpute and BPCA showed similar results when s is small, since they employ the same PC regression process, while BPCA showed better performance than SVDimpute when s is larger since BPCA automatically reduces the redundant principal axes (Oba *et al.*, 2003).

The major difference between BPCA and LLSimpute is that LLSimpute is an optimization process based on local similar structure while BPCA is an optimization method based on PCs. The BPCA achieves an improvement over SVDimpute by incorporating Bayesian optimization and LLSimpute achieves an improvement over KNNimpute by incorporating the least squares. In the study of Troyanskaya *et al.* (2001), it was shown that KNNimpute is more robust and accurate than SVDimpute. The SVDimpute has several weaknesses. The SVDimpute solution relies on entire genes and experiments in the dataset and does not consider local structure. In addition, for non-time series data, a clear expression pattern may not exist. For noisy data, expression patterns for smaller groups of genes may not be represented well by the dominant eigengenes. Based on these observations, it is possible to expect that LLSimpute can exhibit highly competitive performance, which is corroborated in our experiments.

4 CONCLUSION

We have successfully developed local least squares imputation methods for the missing value estimation of DNA microarray gene expression data. Once the genes similar to the target gene with missing values are identified based on Euclidean distance or Pearson correlation coefficient, missing values can be estimated by representing the target gene as a linear combination of the similar genes or by representing the target experiment that has missing values as a linear combination of related experiments. Non-parametric missing values estimation methods of LLSk/L2 and LLSk/PC are designed by introducing automatic k-value estimator. The proposed missing value estimation methods can be applied to various biological and chemical experiment data.

Even though BPCA showed better performances than KNNimpute for all datasets tested in the study of BPCA (Oba *et al.*, 2003), when genes have dominant local similarity structures, BPCA may be less accurate than KNNimpute (Oba *et al.*, 2003). However, our local least squares imputation methods take advantage of the local similarity structures in addition to the optimization process by the least squares, which is one of the most important advance of LLSimpute. Although we cannot guarantee that LLSimpute will always show better performance than BPCA and KNNimpute, our experiments suggest that the LLSimpute is a robust and accurate missing value estimation method.

ACKNOWLEDGEMENTS

The authors would like to thank the University of Minnesota Supercomputing Institute (MSI) for providing the computing facilities. We also thank Dr Shigeyuki Oba for providing datasets and helpful discussions. The work of H.P. has been performed while at the NSF and was partly supported by IR/D from the National Science Foundation (NSF). This material is based upon the work supported in part by the National Science Foundation Grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Alter,O., Brown,P.O. and Botstein,D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl Acad. Sci. USA*, **97**, 10101–10106.

Alter,O., Brown,P.O. and Botstein,D. (2003) Generalized singular value decomposition for comparative analysis of genome-scale expression datasets of two different organisms. *Proc. Natl Acad. Sci. USA*, **100**, 3351–3356.

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

Cho,J.H., Lee,D., Park,J.H. and Lee,I.B. (2003) New gene selection method for classification of cancer subtypes considering within-class variation. *FEBS Lett.*, **551**, 3–7.

Friedland,S., Niknejad,A. and Chihara,L. (2003). A simultaneous reconstruction of missing data in DNA microarrays. *Institute for Mathematics and its Applications Preprint Series*, No. 1948.

Gasch,A.P., Huang,M., Metzner,S., Botstein,D., Elledge,S.J. and Brown,P.O. (2001) Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol. Biol. Cell*, **12**, 2987–3003.

Golub,G.H. and van Loan,C.F. (1996) *Matrix Computations*; 3rd edn. Johns Hopkins University Press, Baltimore, CA.

Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,J.R., Caligiuri,M.A., Bloomfield,C.D. and Lander,E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–537.

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

Pearson,K. (1894) Contributions to the mathematical theory of evolution. *Phil. Trans. R. Soc. London*, **185**, 71–110.

Perou,C.M., Sorlie,T., Eisen,M.B., van de Rijn,M., Jeffrey,S.S., Rees,C.A., Pollack,J.R., Ross,D.T., Johnsen,H., Akslen,L.A. *et al.* (2000) Molecular portraits of human breast tumors. *Nature*, **406**, 747–752.

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

Shipp,M.A., Ross,K.N., Tamayo,P., Weng,A.P., Kutok,J.L., Aguiar,R.C., Gaasenbeek,M., Angelo,M., Reich,M., Pinkus,G.S. *et al.* (2002) Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.*, **8**, 68–74.

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

Takemasa,I., Higuchi,H., Yamamoto,H., Sekimoto,M., Tomita,N., Nakamori,S., Matoba,R., Monden,M. and Matsubara,K. (2001) Construction of preferential cDNA microarray specialized for human colorectal carcinoma: molecular sketch of colorectal cancer. *Biochem. Biophys. Res. Commun.*, **285**, 1244–1249.

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

van't Veer,L.J., Dai,H., van de Vijver,M.J., He,Y.D., Hart,A.A., Mao,M., Peterse,H.L., van der Kooy,K., Marton,M.J., Witteveen,A.T. *et al*. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Vapnik,V. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.