BMC Bioinformatics

BioMed Central

Research

Robust imputation method for missing values in microarray data

Dankyu Yoon[1], Eun-Kyung Lee[2] and Taesung Park*[2]

Address: [1]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, Korea and [2]Department of Statistics, College of Natural Science, Seoul National University, San 56-1, Shin Lim-Dong, Kwanak-ku, Seoul, 151-742, Korea

Email: Dankyu Yoon - avanti@chol.com; Eun-Kyung Lee - lee.eunk@gmail.com; Taesung Park* - tspark@stats.snu.ac.kr

* Corresponding author

from Probabilistic Modeling and Machine Learning in Structural and Systems Biology
Tuusula, Finland. 17–18 June 2006

Published: 3 May 2007

BMC Bioinformatics 2007, **8**(Suppl 2):S6 doi:10.1186/1471-2105-8-S2-S6

This article is available from: http://www.biomedcentral.com/1471-2105/8/S2/S6

Abstract

Background: When analyzing microarray gene expression data, missing values are often encountered. Most multivariate statistical methods proposed for microarray data analysis cannot be applied when the data have missing values. Numerous imputation algorithms have been proposed to estimate the missing values. In this study, we develop a robust least squares estimation with principal components (RLSP) method by extending the local least square imputation (LLSimpute) method. The basic idea of our method is to employ quantile regression to estimate the missing values, using the estimated principal components of a selected set of similar genes.

Results: Using the normalized root mean squares error, the performance of the proposed method was evaluated and compared with other previously proposed imputation methods. The proposed RLSP method clearly outperformed the weighted *k*-nearest neighbors imputation (kNNimpute) method and LLSimpute method, and showed competitive results with Bayesian principal component analysis (BPCA) method.

Conclusion: Adapting the principal components of the selected genes and employing the quantile regression model improved the robustness and accuracy of missing value imputation. Thus, the proposed RLSP method is, according to our empirical studies, more robust and accurate than the widely used kNNimpute and LLSimpute methods.

Background

Microarray experiment technique has been successfully applied to a variety of biological studies including cancer classification, discovery of the unknown gene function, and identification of effects of a specific therapy. When analyzing microarray data, we often face missing values due to various factors such as scratches on the slide, spotting problems, dusts, experimental errors, and so on. In practice, every experiment contains missing entries and sometimes more than 90% of the genes in the microarray experiment are affected [1]. Moreover, most of the classic multivariate analysis methods for microarray data cannot be used when the data have missing values. Therefore, we need to treat missing values appropriately.

An easy way to handle missing data is to repeat the whole experiment. However, often it is not a realistic option secondary to economic limitations and/or scarcity of available biological material [2]. Accordingly, many missing value estimation methods have been developed. The weighted *k*-nearest neighbors imputation method (kNNimpute) selects genes with expression profiles similar to the gene of interest to impute missing values [3]. The singular value decomposition method (SVDimpute) employs a singular value decomposition to obtain to a set of mutually orthogonal patterns that can be linearly combined to approximate the expression of all genes in the data set [3]. In a comparative study presented by Troyanskaya et al. [3], kNNimpute is more robust and accurate than SVDimpute.

Least squares imputation (LSimpute) is a regression-based method using the correlation between both genes and arrays [2]. LSimpute showed best performance when data have a strong local correlation structure. Local least squares imputation (LLSimpute) is an extension of LSimpute method which selects *k* similar genes by L_2-norm or Pearson correlation and applies multiple regression to impute missing values [4].

Bayesian principal component analysis (BPCA) uses a Bayesian estimation algorithm to predict missing values [5]. BPCA suggests using the number of samples minus 1 as the number of principal axes. Since BPCA uses an EM-like repetitive algorithm to estimate missing values, it needs intensive computations to impute missing values. Gaussian mixture imputation (GMCimpute) estimates missing values using Gaussian mixture and model averaging [1]. Collateral missing value imputation (CMVE) [6] predicts missing values based on a multiple covariance-based imputation matrices and performs imputation using least square regression and linear programming methods.

Recently, several imputation methods using a priori information to impute missing values have been proposed such as a set theoretic framework approach based on projection onto convex sets (POCS) [7] and an approach based on the functional similarities of gene ontology [8]. While most traditional missing imputation methods treated spots as binary value such as missing or present, weighted nearest neighbours method (WeNNI) adopted a continuous spot quality weight for the missing value estimation [9].

Among these methods, kNNimpute, LSimpute and LLSimpute are most commonly used because they are easy to apply with less computational burdens. Note that LSimpute and LLSimpute are regression based methods and kNNimpute can also be regarded as a regression based method for the simple intercept model. In this paper, we focus on these regression based methods and present their improvements.

kNNimpute and LLSimpute both use the *k* selected genes to estimate missing values. Kim *et al.* [4] showed LLSimpute performed well for a large value of *k*, say over 200. However, it is inefficient to use such a large number of genes to estimate one missing value from a practical point of view. Furthermore, there is no guarantee that the selected *k* is sufficiently large enough for LLSimpute to perform well. Surprisingly, the performance of LLSimpute becomes very poor when *k* is close to the number of samples.

On the other hand, kNNimpute performs well with relatively small values of *k*. For example, kNNimpute suggests using 10 or 15 similar genes. However, kNNimpute performs poorly when *k* is too small or too large. Its performance depends on the sample size and the correlations between genes. Therefore, kNNimpute is negatively affected by a badly chosen *k*.

To overcome the limitations of these regression based imputation methods, we propose the robust least square estimation with principal components (RLSP) method. RLSP is an improved version of LLSimpute. We use the estimated principal components of the selected genes and apply quantile regression to estimate missing values with the estimated principal components. Note that the most imputation methods are not robust to outliers. RLSP performs well even when *k* is small. Moreover RLSP shows similar performance with LLSimpute when *k* is large. Therefore, RLSP is more robust to the choice of *k*.

The normalized root mean squared error (NRMSE) is used to evaluate the differences in performances between the proposed RLSP method and the other imputation methods for various missing rates [4]. The RLSP method clearly outperforms the LLSimpute method.

Methods

A whole gene expression profile is represented by a $G \times N$ matrix, $\mathbf{Y}$, where the rows correspond to the genes, the columns correspond to the experiments (samples), and the entry $Y_{i,j}$ is the expression level of gene i in experiment j. For simplicity, we assume that the target gene vector $\mathbf{g}^*$ has a missing value at the first sample, denoted by α. For the k selected similar genes, let $\mathbf{g}_{s_j}$ be a $N \times 1$ vector consisting of the jth selected genes with its first element w_{s_j}, where s_j denote the index for representing k selected genes

for $j = 1,...,k$. The selected similar genes have complete values without missing observations. Then,

$$\begin{pmatrix} \mathbf{g}^{*T} \\ \mathbf{g}_{\mathbf{s}_1}^T \\ \mathbf{g}_{\mathbf{s}_2}^T \\ \vdots \\ \mathbf{g}_{\mathbf{s}_k}^T \end{pmatrix} = \begin{pmatrix} \alpha & \gamma_1 & \gamma_2 & \cdots & \gamma_{N-1} \\ w_{s_1} & x_{s_1,1} & x_{s_1,2} & \cdots & x_{s_1,N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w_{s_k} & x_{s_k,1} & x_{s_k,2} & \cdots & x_{s_k,N-1} \end{pmatrix} = \begin{pmatrix} \alpha & \mathbf{y}^T \\ \mathbf{w} & \mathbf{X} \end{pmatrix} \quad (1)$$

where $\mathbf{w} = [w_{s_1}, w_{s_2}, \cdots, w_{s_k}]^T$, and $\mathbf{y} = [\gamma_1, \gamma_2, ..., \gamma_{N-1}]^T$ is a subvector of $\mathbf{g}^*$ excluding the missing value α.

LLSimpute selects the k most similar genes using L_2-norm or Pearson correlation and applies multiple regression to impute missing values with a linear combination of the k selected genes. LLSimpute applies multiple regression in two ways.

The first model is

$$\mathbf{y} = \mathbf{X}^T\beta + \varepsilon$$

$$\hat{\alpha} = \mathbf{w}^T\hat{\beta} = \mathbf{w}^T(\mathbf{X}\mathbf{X}^T)^{-}\mathbf{X}\mathbf{y} \quad (2)$$

and the second model is

$$\mathbf{w} = \mathbf{X}\beta^* + \varepsilon^*$$

$$\hat{\alpha}^* = \mathbf{y}^T\hat{\beta}^* = \mathbf{y}^T(\mathbf{X}^T\mathbf{X})^{-}\mathbf{X}^T\mathbf{w} \quad (3)$$

where $(\mathbf{X}\mathbf{X}^T)^-$ is the generalized inverse of $(\mathbf{X}\mathbf{X}^T)$. If N is larger than k, $(\mathbf{X}\mathbf{X}^T)^-$ in the first model is easier to calculate $(\mathbf{X}^T\mathbf{X})^-$ than in the second model and vice versa.

In case of multiple missing values in a gene, all missing components of each gene are excluded to find similar genes. Then, the vectors $\mathbf{w}$ and $\mathbf{y}^T$, and matrix $\mathbf{X}$ are formed in a similar way as in the case of one missing entry, only with different dimensions.

LLSimpute showed a good performance for a relatively large value of k. However, if a value of k is close to the number of samples, LLSimpute performed poorly compared to other imputation methods. It is probably due to the multi-collinearity problem that LLSimpute performs poorly when k is small. The patterns of gene expression are highly correlated leading to the poor performance of multiple regression.

To overcome this limitation, we perform a regression with the principal components rather than the original data. Our technique utilizes the selection of two models in terms of k and applies the principal components analysis to the k selected genes. Also we consider the robustness to reduce the effects of the outliers by fitting robust regression.

The RLSP method consists of three parts: (1) selection of k similar genes, (2) principal component analysis with the k selected genes, and (3) robust regression analysis using these principal components. We describe these processes step by step.

STEP 1 : Selection of k *similar genes*

To impute a missing value α, the k similar genes are used for RLSP, where k is a pre-determined number. In LLSimpute, L_2-norm or Pearson correlation coefficient is used to select k similar genes. However, it is well known that L_2-norm and Pearson correlation coefficients are sensitive to outliers. In RLSP, we use L_1-norm as a distance measure to select the k similar genes for imputing the missing values of the gene g^*,

$$d(\mathbf{g}^*, \mathbf{g}_i) = \sum_{j=1}^{N-1} \left| x_{i,j} - \gamma_j \right|.$$

STEP 2 : Principal component

After selecting k similar genes, we perform the principal component analysis. We define two types of variance-covariance matrix. The first one is a $k \times k$ matrix

$$V = \sum_{i=1}^{N-1}\sum_{j=1}^{N-1}(\mathbf{x}_i - \bar{\mathbf{x}}_\cdot)(\mathbf{x}_j - \bar{\mathbf{x}}_\cdot)^T \quad (4)$$

where $\mathbf{x}_i = [x_{s_1,i}, x_{s_2,i}, \cdots, x_{s_k,i}]^T$, $\bar{\mathbf{x}}_\cdot = [\bar{x}_{s_1,\cdot}, \bar{x}_{s_2,\cdot}, \cdots, \bar{x}_{s_k,\cdot}]^T$, and $\bar{x}_{s_l,\cdot} = \frac{1}{N-1}\sum_{i=1}^{N-1} x_{s_l,i}$. The second type is a $(N\text{-}1) \times (N\text{-}1)$ matrix

$$V^* = \sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{x}_i^* - \bar{\mathbf{x}}_\cdot^*)(\mathbf{x}_j^* - \bar{\mathbf{x}}_\cdot^*)^T \quad (5)$$

where

$\mathbf{x}_i^* = [x_{s_i,1}, x_{s_i,2}, \cdots, x_{s_i,N-1}]^T$, $\bar{\mathbf{x}}_\cdot^* = [\bar{x}_{\cdot,1}, \bar{x}_{\cdot,2}, \cdots, \bar{x}_{\cdot,N-1}]^T$, and $\bar{x}_{\cdot,l} = \frac{1}{k}\sum_{i=1}^{k} x_{s_i,l}$.

When k is larger than N, the size of V matrix becomes too large to handle and it is not computationally efficient to derive the principal components. Therefore, we use V^* instead of V and use a different type of regression. V corresponds to the first model and V^* corresponds to the second model in LLS impute, respectively. Kim et al. [4] showed that the solutions based on V and V* are in fact

the same. Let $\mathbf{PC}_x = \{PC_1, PC_2,...,PC_k\}$ be the principal components using V and $\mathbf{PC}_x^* = \{PC_1^*, PC_2^*, \cdots, PC_{N-1}^*\}$ be the principal components using V^*. Then, these principal components $\mathbf{PC}_x$ and $\mathbf{PC}_x^*$ are used for the first type of regression (equation (2)) and the second type of regression model (equation (3)), respectively.

STEP 3 : Robust regression

We use $PC_1, PC_2,...,PC_p$ or $PC_1^*, PC_2^*, \cdots, PC_p^*$ as new exploratory variables and fit the regression model in a robust manner, where p is the predetermined number of the principal components. The corresponding regression models are

$$\mathbf{y} = \sum_{i=1}^{p} PC_i \beta_i + \varepsilon \qquad (6)$$

and

$$\mathbf{w} = \sum_{i=1}^{p} PC_i^* \beta_i^* + \varepsilon^* \qquad (7)$$

In our method, we use a quantile regression to fit the regression model in a robust manner. Robust regression usually provides an alternative analysis to least square regression when fundamental assumptions such as normality or variance homogeneity are violated. The quantile regression using the 50th percentile estimates the model parameters by minimizing the sum of absolute values of the residuals [10]. It is estimated by minimizing $\sum_{j=1}^{N-1} \left| y_j - \sum_{i=1}^{p} PC_{ij} \beta_i \right|$ or $\sum_{j=1}^{k} \left| w_{s_j} - \sum_{i=1}^{p} PC_{ij}^* \beta_i^* \right|$. This way our analysis method can reject outliers and maintains robustness.

If $p = k$ and the regression model is estimated by the least squares method, RLSP is the same as LLSimpute. When k is small, we recommend using $p = 1$ and fit the regression model $y_i = \beta_1 PC_{1i} + \varepsilon_i$ using the sum of least absolute deviations. The imputed value of α is defined by $\hat{\alpha} = \hat{\beta}_1 PC_w$ where PC_w is the projected data of $\mathbf{w}$ onto the direction of PC_1. For k much larger than N, we recommend using p close to the number of the sample size.

Results and discussion

Datasets

Four data sets are used for the comparative study: three Spellman data sets (ALPHA, ELU, and ALPHA+ELU, [5,11]), and Gasch data set [12]. These data sets were also used in the comparative study of LLSimpute [4]. ALPHA dataset was obtained from α-factor block release studied for the identification of cell-cycle regulated genes in yeast *Saccharomyces cerevisiae* [13]. ELU dataset was elutriation dataset in the same study. After removing all the genes with missing values in the ALPHA and ELU datasets, we obtained complete data matrices that contain 4,304 genes and 18 experiments and 4,304 genes and 14 experiments, respectively. ALPHA+ELU dataset was used for the examination of the additional sample effects as studied Oba et al. [5]. Gasch dataset was obtained from the study of genomic expression responses to DNA damage [14]. After removing all genes with missing values, a complete data matrix with 2641 genes and 44 experiments was prepared for this study. For the simulation study, 1% and 5% missing observations were randomly generated in these data sets.

We use the normalized root mean squares error (NRMSE) to evaluate the performances of the missing value imputation approaches, computed by

$$\text{NRMSE} = \sqrt{mean[(\gamma_{\text{guess}} - \gamma_{\text{answer}})^2]} \Big/ sd(\gamma_{\text{answer}}),$$

where γ_{guess} is the imputed value and γ_{answer} the true value.

Experimental results

Figures 1 and 2 show the plots of NRMSE vs. k for the ELU Spellman data set, when the missing rates are 1% and 5%, respectively. We compare our RLSP with kNNimpute, LLSimpute and BPCA. For a large k, both LLSimpute and RLSP show highly competitive results and perform best compared to kNNimpute and BPCA. However, for a smaller k, RLSP performed much better than LLSimpute. LLSimpute shows a high peak when k is close to the number of samples. Because highly correlated k genes are usually selected in LLSimpute, the poor performance of LLSimpute is probably due to multi-collinearity of the selected k genes.

Figures 3 and 4 show the smallest NRMSE values for four data sets. Figure 3 represents the results of the case of 1% missing rate. LLSimpute and RLSP showed the similar results and the best performances in ALPHA and ELU data sets. For ALPHA+ELU and Gasch data, BPCA showed a little bit better performance than RLSP and LLSimpute, but it is competitive to LLSimpute and RLSP. Figure 4 shows the results of the case of the 5% missing rate. RLSP, LLSimpute and BPCA show competitive performances for ALPHA and ELU data sets. For large datasets such as

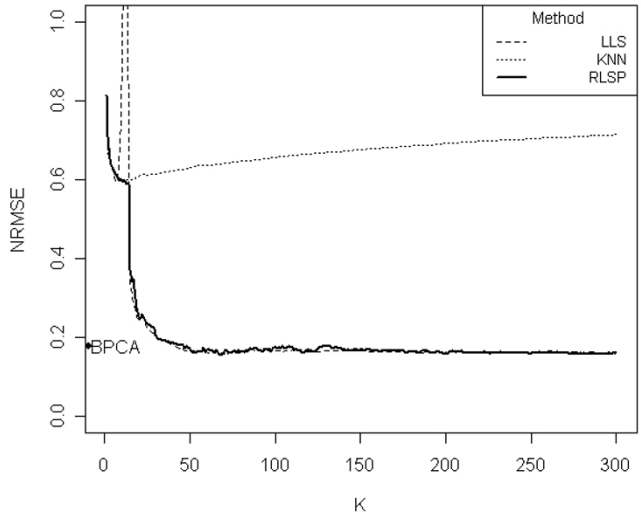

Figure 1
Comparison of the NRMSEs of various methods. Comparison of the NRMSEs of LLSimpute, kNN, BPCA, and RLSP imputation methods on ELU data set with the 1% missing rate. BPCA results are shown on the y-axis. The x-axis represents the value of *k* (selected similar genes). When *k* is close to sample size (*N*), LLSimpute has a high peak. Thus we truncated it in the graph. Overall, RLSP method showed improved performance compared to other methods.

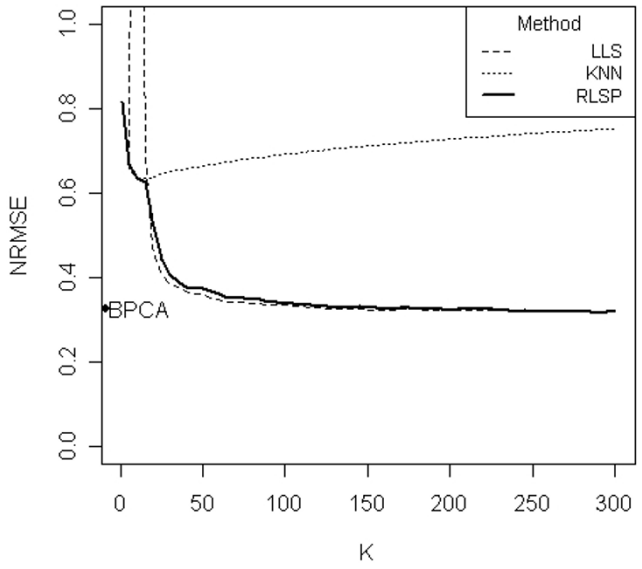

Figure 2
Comparison of the NRMSEs of various methods. Comparison of the NRMSEs of LLSimpute, kNN, BPCA, and RLSP imputation methods on ELU data set with the 5% missing rate. BPCA results are shown on the y-axis. The x-axis represents the value of *k* (selected similar genes). When *k* is close to sample size (*N*), LLSimpute has a high peak. Thus we truncated it in the graph. RLSP method demonstrated a better result than other methods.

ALPHA+ELU and Gasch data sets, BPCA showed a little bit better performance. kNNimpute showed the worst performance in all data sets.

We presume that the differences in the performance of missing value imputation methods are highly dependent on the data set as well as the value of *k*. When a moderate value of *k* is selected, say when *k* is close to the number of samples, the proposed RLSP method outperforms the LLSimpute method on all data sets (data not shown). As the missing rate increases, NRMSEs increase rapidly for all methods and the performances of all four methods become worse.

Conclusion

The proposed RLSP method was motivated by a similar idea to that of the LLSimpute method. Both methods use the information from the selected *k* genes to estimate missing observations. LLSimpute uses the least squares method using the selected *k* genes. On the other hand, RLSP uses the principal components of the selected genes instead of the original *k* genes, and employs the quantile regression model for a robust analysis. The use of the principal components leads to a large difference between the two methods. The performance of LLSimpute is poor when *k* is small (near the number of samples). Assuming that multi-collinearity is probably the main cause for the poor performance of LLSimpute, RLSP addresses this problem using the principal components and then applying the robust regression approach to reduce the effect of outliers. In summary, RLSP showed more stable performance than LLSimpute for all data sets in our comparative studies.

The performance of RLSP may depend on the value of *p*, the number of principal components. By varying the value of *p*, we examined its effect on the parameter estimators. The result showed that the performance of RLSP is optimal when the value of *p* is close to the number of the sample size (data not shown). A similar result was obtained from a previous study of the BPCA method [5]. However, since the imputation procedure is executed for each missing value of a gene, the optimal value of *p* may differ from gene to gene. Selecting an optimal value for each missing value requires intensive computation. Thus, we recommend using the number of sample size as the value of *p* for practical application, although we expect that an appropriate choice of *p* would improve the performance of the RLSP method.

In terms of computational efficiency, although RLSP and BPCA showed competitive results, BPCA required a higher

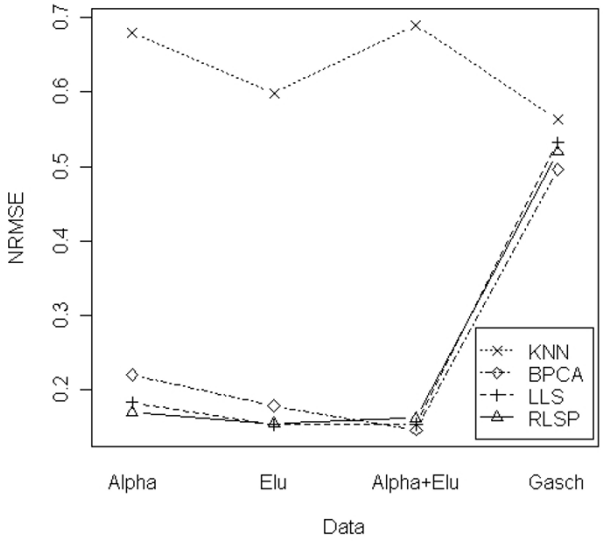

Figure 3
Comparison of the NRMSEs for 4 different data sets. Comparison of the NRMSEs for 4 different data sets (ALPHA, ELU, ALPHA+ELU, and Gasch) with 1% missing rate.

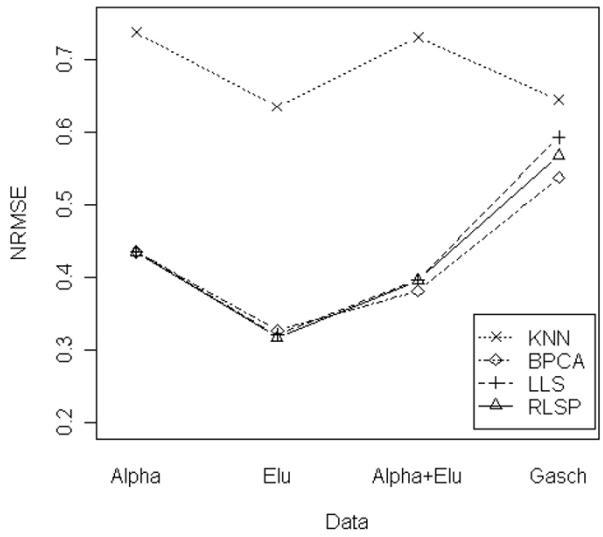

Figure 4
Comparison of the NRMSEs for 4 different data sets. Comparison of the NRMSEs for 4 different data sets (ALPHA, ELU, ALPHA+ELU, and Gasch) with 5% missing rate.

computational demand due to the EM-like repetitive algorithm. In addition, RLSP seemed less computationally intensive than CMVE. However, a further study based on the same platform would be desirable for the systematic comparison.

The presented method consists of three separate steps, where the first step applied L1 metric to select similar genes, the second step performs PCA on the selected set, which is a L2 method, and finally in the third step L1 metric is applied again to perform robust regression. Among the several combinations of metrics, the proposed combination provided the minimum NRMSE and provided the most computationally efficient result.

The main motivation of the robust regression was to reduce the effect of outliers in estimation of missing observations. Our empirical studies demonstrated that the effect of outliers were not large enough to cause huge differences between robust regression and ordinary regression. Among the several robust regression methods including Tukey's bi-weight M-estimator, the proposed quantile regression using the 50th percentile provided the best result. However, a further study on selecting the better robust method is desirable.

Finally, most missing imputation methods for microarray data assume the simple missing data mechanism to be the so called 'missing completely at random' [15]. However, this mechanism may assume too much to be expected to hold in real applications. Therefore, more complicated methods are required for handling other possible missing data mechanisms. By incorporating the missing data mechanism or missing patterns in the microarray data, we could improve the performance of the missing imputation method.

Authors' contributions

All authors contributed to the development of RLSP method and the comparative studies with previously proposed imputation methods.

Acknowledgements

The authors would like to thank the editors and two anonymous referees whose comments were extremely helpful. The authors also would like to thank Dr. Sehgal for many constructive discussions on the CMVE program. The work was supported by the National Research Laboratory Program of Korea Science and Engineering Foundation (M105000000126) and the Brain Korea 21 Project of the Ministry of Education.

This article has been published as part of *BMC Bioinformatics* Volume 8, Supplement 2, 2007: Probabilistic Modeling and Machine Learning in Structural and Systems Biology. The full contents of the supplement are available online at http://www.biomedcentral.com/1471-2105/8?issue=S2.

References

1. Ouyang M, Welsh WJ, Georgopoulos P: **Gaussian mixture clustering and imputation of microarray data.** *Bioinformatics* 2004, **20(6):**917-923.
2. Bo TH, Dysvik B, Jonassen I: **LSimpute: accurate estimation of missing values in microarray data with least squares methods.** *Nucleic Acids Res* 2004, **32(3):**e34.
3. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R: **Missing value estimation methods for DNA microarrys.** *Bioinformatics* 2001, **17(6):**520-525.
4. Kim H, Golub GH, Park H: **Missing Value Estimation for DNA microarray gene expression data: local least squares imputation.** *Bioinformatics* 2005, **21(2):**187-198.
5. Oba S, Sato M, Takemasa I, Monden M, Matsubara K, Ishii S: **A Bayesian missing value estimation method for gene expression profile data.** *Bioinformatics* 2003, **19(16):**2088-2096.
6. Sehgal MS, Gondal I, Dooley LS: **Collateral missing value imputation: a new robust missing value estimation algorithm for microarray data.** *Bioinformatics* 2005, **21(10):**2417-2423.
7. Gan X, Liew AW, Yan H: **Microarray missing data imputation based on a set theoretic framework and biological knowledge.** *Nucleic Acids Res* 2006, **34(5):**1608-1619.
8. Tuikkala J, Elo L, Nevalainen OS, Aittokallio T: **Improving missing value estimation in microarray data with gene ontology.** *Bioinformatics* 2006, **22(5):**566-572.
9. Johansson P, Hakkinen J: **Improving missing value imputation of microarray data by using spot quality weights.** *BMC Bioinformatics* 2006, **7():**306.
10. Koenker R, Hallock K: **Quantile Regression.** *Journal of Economic Perspectives* 2001, **15:**143-156.
11. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B: **Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization.** *Mol Biol Cell* 1998, **9:**3273-3297.
12. Gasch AP, Huang M, Metzner S, Bostein D, Elledge SJ, Brown PO: **Genomic expression responses to DNA damaging agents and the regulatory role of the yeast ATR homolog Mec1p.** *Mol Biol Cell* 2001, **12:**2987-3003.
13. **Yeast Cell Cycle Analysis Project** [http://cellcycle-www.stanford.edu]
14. **The web supplement to Gasch et al** [http://www-genome.stanford.edu/Mec1]
15. Little RJA, Rubin DB: **Statistical analysis with missing data.** 2nd edition. *Wiley, Hoboken, New Jersey*; 2002.