Imputation of Missing Gene Expressions from Microarray Dataset:A Review

Article *in* International Journal of Emerging Trends & Technology in Computer Science · April 2017

DOI: 10.14445/22312803/IJCTT-V46P104

Rakhi Wajgi
Yeshwantrao Chavan College of Engineering
22 PUBLICATIONS **103** CITATIONS
SEE PROFILE

Manali Kshirsagar
Yeshwantrao Chavan College of Engineering
20 PUBLICATIONS **56** CITATIONS
SEE PROFILE

Imputation of Missing Gene Expressions from Microarray Dataset: A Review

Chanda Panse(Wajgi)#1, Manali Kshirsagar#2, Dipak Wajgi*3

#Department of Computer Technology,YCCE, Nagpur, India 441110

*Department of Computer Engineering, SVPCE, Nagpur, India 441108

Abstract:

DNA microarray technology captures gene expressions of thousands of genes simultaneously. But while recording these gene expressions through software after scanning, missing values get generated in the database due to various artifacts. It could be due to variety of reasons including hybridization failures, artefacts on the microarray, insufficient resolution, noisy image or corrupted image. It may also occur systematically as a result of the spotting process. This hinders performance of downstream analysis. There are certain solutions proposed in the literature to deal with this problem but due to their limitations imputation of missing values is preferred as the best solutions. This paper presented a review of existing methods used for imputation of missing values along with their advantages and limitations.

Keywords: *Microarray, Hybridization, Imputation*

I. INTRODUCTION

In today's world, computational biology is emerging as a major area of research since vast amount of biological data getting accumulated day by day. To understand functions of genes is a major challenge. To know the behavior of genes, they are observed at different instances through microarray technology. In microarray technique large numbers of tiny spots are attached on thin glass slide known as probes. There may be tens or thousands of spots on the array each containing millions of DNA molecules. For gene expressions, each of these molecules should identify single mRNA through transcription. Target RNA is reversed transcribed with complementary DNA (cDNA) labeled with fluorescent dyes and hybridized with cDNA on same glass slide. Hybridization occurs only when both the strands complement each other. Intensity of fluorescent is treated as amount of RNA expressed in the given sample. The overall process of microarray preparation is shown in Fig. 1.

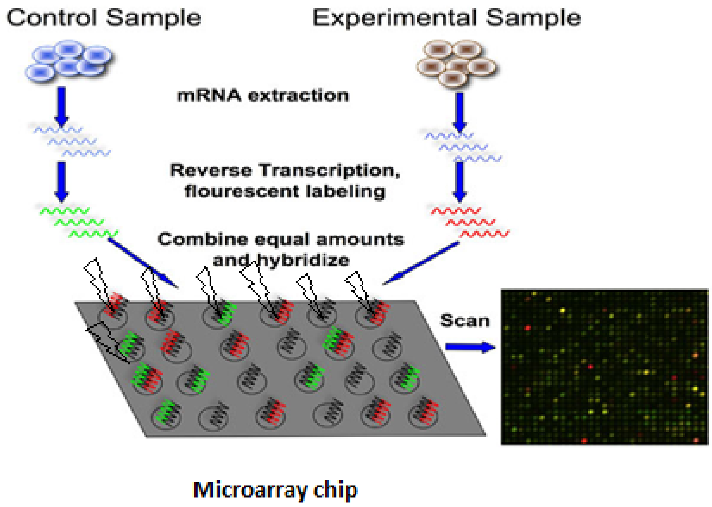

Control Sample
Experimental Sample
mRNA extraction
Reverse Transcription, flourescent labeling
Combine equal amounts and hybridize
Scan
Microarray chip

Fig. 1 Microarray preparations using two channels

After scanning microarray chip through microarray scanner, image containing color dots is generated. This is shown in Fig. 2.

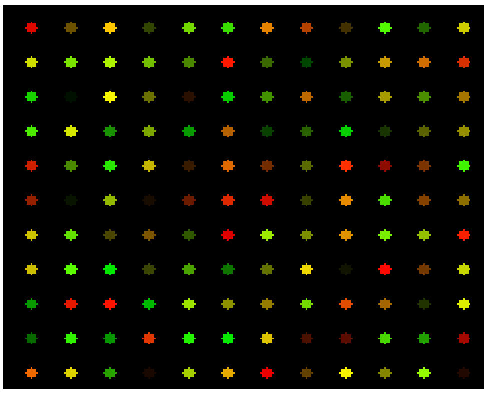


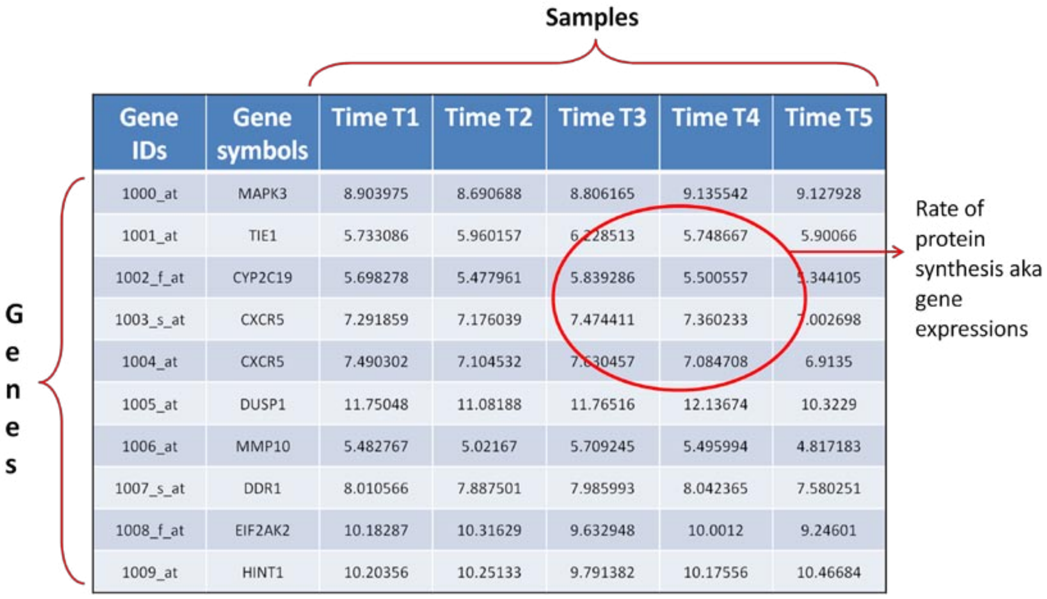
Fig. 2 Image generated from microarray scanner

By applying various image processing techniques mentioned in [1] on resultant microarray image, data values corresponding to gene expressions are generated through software like Acuity PTO built in with scanner which is ratio of red to green intensities. Red dot indicates DNA probe expressed from experimental sample while green indicates DNA probe expressed from fixed sample and yellow probe means equal intensities expressed from both samples.

One microarray chip measures expressions of thousands of genes. But while creating microarray chip noise gets added in the samples which results in error generation. To minimize the percentage of error, samples are replicated i.e. samples are measured for few more times to reduce the effect of noise. In this paper non-replicated datasets are considered. Many times due to polluted glass slides, some expressions are not recorded which results in generation of missing values that hinder the performance of downstream analysis. To avoid this, preprocessing of microarray is needed. The reviews of various preprocessing techniques are discussed in the next section.

II. PREPROCESSING OF MICROARRAY

Microarray gene expression dataset consists of matrix of size NxM containing gene expressions of N genes for M samples. Sample microarray matrix is shown in Fig. 3. Analysis of gene expressions is important in the view of building of gene regulatory network, tumor classification, genome analysis, and drug designing.


Samples
Genes
Rate of protein synthesis aka gene expressions

Gene IDs	Gene symbols	Time T1	Time T2	Time T3	Time T4	Time T5
1000_at	MAPK3	8.903975	8.690688	8.806165	9.135542	9.127928
1001_at	TIE1	5.733086	5.960157	6.228513	5.748667	5.90066
1002_f_at	CYP2C19	5.698278	5.477961	5.839286	5.500557	5.344105
1003_s_at	CXCR5	7.291859	7.176039	7.474411	7.360233	7.002698
1004_at	CXCR5	7.490302	7.104532	7.630457	7.084708	6.9135
1005_at	DUSP1	11.75048	11.08188	11.76516	12.13674	10.3229
1006_at	MMP10	5.482767	5.02167	5.709245	5.495994	4.817183
1007_s_at	DDR1	8.010566	7.887501	7.985993	8.042365	7.580251
1008_f_at	EIF2AK2	10.18287	10.31629	9.632948	10.0012	9.24601
1009_at	HINT1	10.20356	10.25133	9.791382	10.17556	10.46684

Fig. 3 Sample Microarray Matrix

There are different platforms and methods used for preparation of microarray. To maintain uniformity in data generated on different platforms using different types of chips, preprocessing is done in the form of normalization. Normalization removes all the distortions in gene expressions and prepares them for fair comparison. It helps in uniform interpretation of changes in gene expressions.

Gene Expression Omnibus is the huge repository of microarray data where most of the data is available in log2 normalized form. In order to minimize the impact of artifacts (i.e. dye biasing) Quantile normalization is performed on log2 data. Fig. 4 and Fig. 5 show Box plots for yeast-alpha dataset before and after quantile normalization respectively. Box plot shows how data is distributed by considering five factors: minimum, first quartile, median, third quartile and maximum. It also shows outliers data using plus symbol with red color. Outliers are those data points which have values greater or less than maximum and minimum respectively are shown above and below the top and bottom whiskers. For time series data box plot helps in identifying effects of noise on gene expressions. In Fig. 5 centers of all boxes are aligned which is an indication of noise reduction.

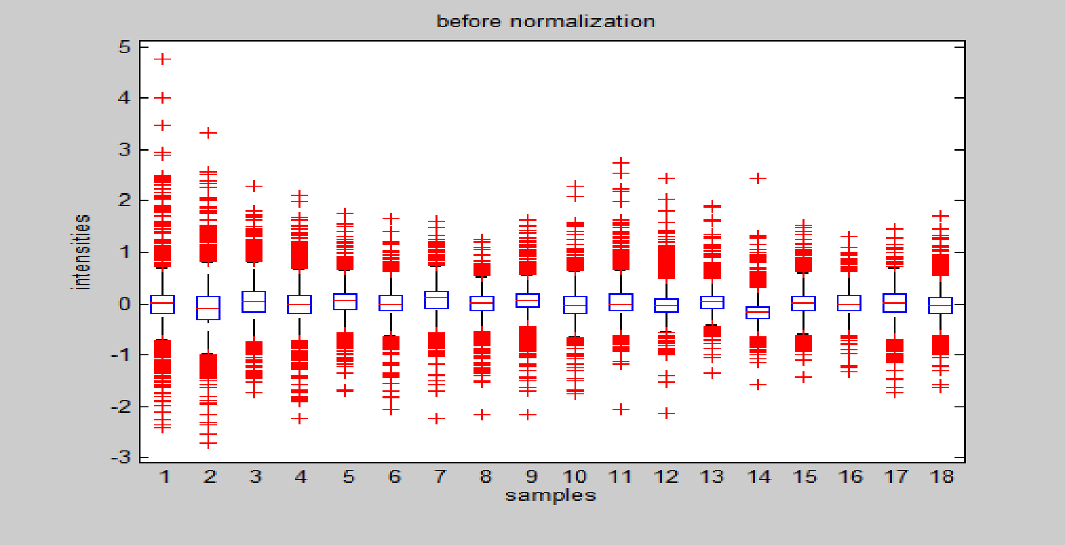


Fig. 4 Sample Boxplot before normalization

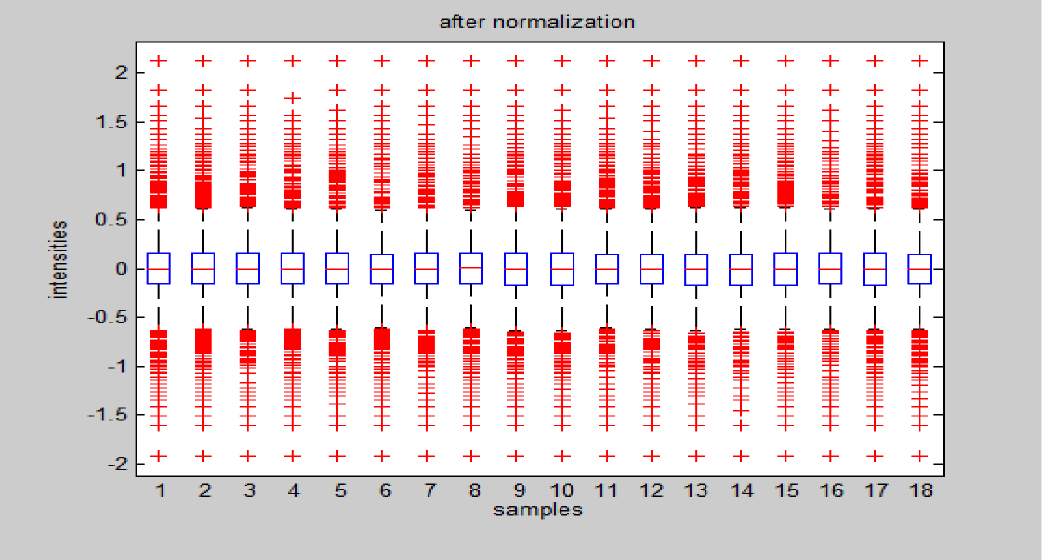
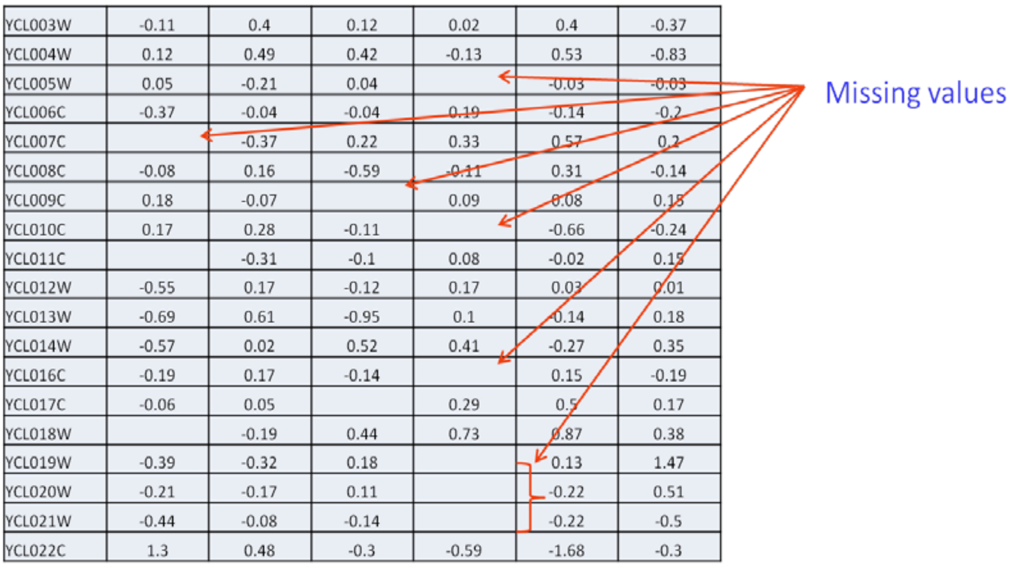


Fig. 5 Sample Boxplot after normalization

Though microarray technology is widely used it is still error prone. Microarray data can contain up to 10% missing values and in some datasets, up to 90% of genes have one or more missing values [2][3][4]. Presence of missing values create hurdles in down-stream analysis such as identification of differentially expressing genes [5][6], reconstruction Gene Regulatory Network [7][8].

Missing values occur due to a variety of reasons including hybridization failures, artefacts on the microarray, insufficient resolution, noisy image or corrupted image, or they may occur systematically as a result of the spotting process [9].

A spot that has negative background will have corrected intensity value normally declared as missing value. In addition, suspicious values are also often flagged as missing. Sample yeast cell cycle dataset containing 7% of missing values is shown in Fig. 6. Table I shows various datasets with percentage of missing values in them. Yeast cell cycle dataset contains least percentage of missing values as it is lengthy process and samples recorded are less.



YCL003W	-0.11	0.4	0.12	0.02	0.4	-0.37
YCL004W	0.12	0.49	0.42	-0.13	0.53	-0.83
YCL005W	0.05	-0.21	0.04		-0.03	-0.03
YCL006C	-0.37	-0.04	-0.04	0.19	-0.14	-0.2
YCL007C		-0.37	0.22	0.33	0.57	0.2
YCL008C	-0.08	0.16	-0.59	-0.11	0.31	-0.14
YCL009C	0.18	-0.07		0.09	0.08	0.15
YCL010C	0.17	0.28	-0.11		-0.66	-0.24
YCL011C		-0.31	-0.1	0.08	-0.02	0.15
YCL012W	-0.55	0.17	-0.12	0.17	0.03	0.01
YCL013W	-0.69	0.61	-0.95	0.1	-0.14	0.18
YCL014W	-0.57	0.02	0.52	0.41	-0.27	0.35
YCL016C	-0.19	0.17	-0.14		0.15	-0.19
YCL017C	-0.06	0.05		0.29	0.5	0.17
YCL018W		-0.19	0.44	0.73	0.87	0.38
YCL019W	-0.39	-0.32	0.18		0.13	1.47
YCL020W	-0.21	-0.17	0.11		-0.22	0.51
YCL021W	-0.44	-0.08	-0.14		-0.22	-0.5
YCL022C	1.3	0.48	-0.3	-0.59	-1.68	-0.3

Fig. 6 Sample yeast cell cycle dataset containing 7% of missing values

The Gasch dataset [10] consists of 173 experiments which examine gene expressions to predict behaviour of yeast during various stress conditions.

TABLE I SAMPLE DATASETS WITH MISSING VALUES

Sr.No.	Dataset	Reference	Dataset size	% of Missing values
1.	Yeast (CDC28)	[23]	6178 x 17	80%
2.	Yeast(α-factor)	[23]	6178x18	46%
3.	CDC15_28	[23]	6178x41	90%
4.	Gasch dataset	[10]	6152x173	91%
5.	Yeast Cell Cycle	[23]	6178x76	7%

To deal with the missing values, there are following solutions proposed in the literature [3].

(1) Repeat the experiments which is quiet costly and time consuming

(2) Neglect rows containing missing values. But this will hamper downstream analysis due to loss of important data

(3) Replace missing values with zeros which does not create any impact on data

(4) Replace missing values with row averaging. If missing values are more in one row then all will be replaced by same expressions which will not reflect variation in gene expression. This does not consider nature of co-expressing genes

(5) Perform method of imputation for estimation of missing values which is flexible, reliable and cheaper solution.

III. REVIEW OF IMPUTATION METHODS

In this section review of methods used for missing value imputation is presented. Imputation techniques discovered after 2010, have considered the knowledge and information related to the dataset for estimation of missing values. We classified different existing methods of imputations into two categories: Distance Based Imputation Techniques (DBIT) and non-Distance Based Imputation Techniques (NDBIT) which are reviewed in detailed as follows.

Imputation methods classified under the category DBIT includes: K-Nearest Neighbor Impute (KNN) , Sequential KNN Impute, GO Based KNN Impute and Non-Distance Based Imputation NDBIT includes : Row Averaging (RAVG), Zero imputation (ZERO), Singular Value Decomposition based imputation, Local Least Square Imputation, Bayesian Principal Component Analysis (BPCA), Entropy-based selection (EBS), Self-training selection (STS), Support Vector Regression imputation, Iterated Local Least Squares (iLLS), Fixed Rank Approximation Method, Gaussian Mixture Clustering Method, Collateral Missing Value Imputation.

A. DBIT Methods

Algorithms in this category perform missing value imputation based on different distance calculation measures between similarly expressing genes. It is assumed that lower distance indicates more similarity between genes and suitable choice as neighbor to be considered for imputation.

In KNN-based impute method [9], k neighbors of target genes containing missing values are selected by finding correlation coefficient. Highly correlated k neighbors are selected for imputation. If gene G is target gene containing missing expression at position i^{th} row j^{th} column of expression matrix, then expression at that position is denoted as E_{ij} where i is the row value and j is the column value. Missing value in the target gene is imputed as average of j^{th} component of k neighbors. Euclidean distance is the best suitable distance measure used in KNN impute. This method is used when strong correlation exists between genes in the data. In weighted KNN impute method, significance of i^{th} neighbor is decided from its weight factor which is calculated as

$$W_i = \frac{1/D_i}{\sum_{i=1}^{k} D_i} \qquad (1)$$

Where D_i is the Euclidean distance between target gene and i^{th} neighbor.

Several variants of basic KNN impute algorithms are proposed in [11][12]. For evaluating the accuracy of KNN impute some percentage of gene expression are deleted (between 1% to 20%) randomly from existing datasets and Root Mean Squared Error (RMSE) is calculated between original value and missing value which is defined as

$$RMS_{error} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (R_i - I_i)^2} \qquad (2)$$

Where R_i is the original value and I_i is imputed value and N is number of missing expressions. This method achieves the best result when RMS_{error} is minimum but it is suitable for samples less than 4 columns.

Sequential KNN Impute described in [11], is implemented by doing small variation in basic KNN impute. The complete expressions are divided into two sets: the reference set which is a complete set without missing values and another target set containing genes with missing values. Initially imputation is applied on the gene with lowest number of missing values using KNN impute and once all missing values of this genes are calculated, it is moved to reference set for further imputation of missing values for genes in target set. It has been observed that performance of SKNN was higher than KNN imputing method for the data with high missing rates and large number of experiments but again deciding value of k is a major challenge.

In [13], GO based KNN impute method is presented. Gene ontology (GO) is a structured network of defined terms which describes gene product attributes. The goal of the gene ontology is to produce a dynamic, controlled vocabulary that can be applied to all eukaryotes even as knowledge of gene and protein roles in cells is accumulating and changing. This method uses similarities in gene semantics from gene ontology annotations to improve the performance of KNN impute method. Using the gene ontology downloaded from GO web site, ontology tree is created and is used for calculating semantic dissimilarity between two genes g1 and g2. However this method is not suitable for larger number of missing values.

B. Non Distance Based Imputation Techniques

This category of imputation method does not involve use of distance measure formula for selecting target genes whose expressions are used for missing value imputation.

The most common and simplest method of missing value imputation is Row Averaging (RAVG) [14]. In this method, missing value in row i is estimated by taking average of all remaining available values in the same row. This newly estimated value is further used for finding missing values in the same row if more than one missing value exists in same row. In Zero Imputation [14] method, missing values are replaced with zeroes. This method is not optimal since it does not consider correlation structure of data.

Singular Value Decomposition based imputation (SVD) method presented in [9]. SVD is based on a theorem from linear algebra which says that a rectangular matrix *A* can be broken down into the product of three matrices - an orthogonal matrix *U*, a diagonal matrix *S*, and the transpose of an orthogonal matrix *V*. The theorem is usually presented as

$$A_{mn} = U_{mn}\, S_{mn}\, V_{nn}^{T} \qquad (3)$$

Where U is orthonormal eigenvectors calculated from AA^T and V is orthonormal eigenvectors calculated from A^TA and S is diagonal matrix containing eigenvalues from U or V arranged in descending order. Let A be the complete matrix of size mxn where m is number of genes and n is number of samples and m > n. Columns of missing values for row r are identified as s_1…..,$s_{t..}$ Let B contains complete rows of A. After applying SVD impute on A, linear regression is calculated against n-t non missing entries of first k rows of V^T and n-t missing values in row R. Missing expressions in row r are estimated by

$$R^j = \sum_{k=1}^{K} R_k^{(j)} \qquad j = 1,2,\ldots \qquad (4)$$

This procedure is repeated until error between imputed and original source matrix Y is minimum. Accuracy achieved using SVD impute is more than row averaging. However performance of this method deteriorates with increasing missing values and it is more sensitive to values of k and generally used for small sample dataset.

In Least Square Imputation (LSI) which is based on regression, to impute missing value of gene g, highly correlated k genes are chosen [15]. The target gene and reference k genes are assumed to be related by linear regression model $y = \alpha + \beta x + \varepsilon$. initially pair-wise linear regression is taken between target gene and k reference gene. After k estimates all are linearly combined to get missing value. LSI is nearly same as KNN impute except it uses least square regression instead of Euclidean distance. It provides flexibility in selecting k reference genes for missing value imputation. This method is generally used for large data with highly correlated genes.
There are three more extensions to LSimpute which are Local Least square (LLS)[16] impute and Sequential LLS impute [17] and iterative LLS impute(iLLS). Local Least Square Imputation (LLS) achieves missing value estimation from k reference genes using multiple regression models

$$y = X^T \alpha + \varepsilon \qquad (5)$$

It is at par with KNN impute method. In sequential LLS, imputation starts with gene with lowest missing values. Gene for whom missing values are calculated took part in further missing value estimation of other genes provided they have less number of missing values below certain threshold. Because of reusability of genes, Sequential LLS imputation is stronger than LLS impute. In iLLS impute, every target gene have different reference genes based on correlation between reference genes and each target gene. Number of reference genes vary based upon distance threshold which is proportional to distance between target genes and reference genes. iLLS use imputed value in previous iteration for further imputation by again selecting reference genes. iLLS is better than basic LSimpute but it is again applicable for dataset containing small set of missing values.

A new method of imputation called Bayesian Principal Component Analysis (BPCA) is presented in [8]. It uses probabilistic Bayesian theorem for missing value imputation It mainly consists of three phases: i) Principal Component Regression, ii) Bayesian Estimation and iii) Exceptation-Maximazation problem. The given D dimensional gene expression matrix Y is represented as k principal axis vectors ω_l (K < D)

$$y = \sum_{l=1}^{K} x_l \omega_l + \in \qquad (6)$$

The linear coefficients x_l are called factor scores and $\in$ denoted residual error. Eigen values and Eigen vectors of covariance matrix S of Y is calculated as follows from which principal axis vector is estimated.

$$S = \frac{1}{N} \sum_{i=1}^{N} (y_i - \mu)(y_i - \mu)^T \qquad (7)$$

Where μ is the mean vector of Y: μ defined as (1/N) $\sum_{i=1}^{N} y_i$. T denotes transpose of a matrix. If $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ and $u_1, u_{2,} \ldots . u_D$ denote eigenvalues and eigenvectors of covariance matrix S then l-th principal axis component is calculated as $\omega_l = \sqrt{\lambda_l u_l}$. From this l-th factor score of expression vector y is calculated by $x_l = \left(\frac{\omega_l}{\lambda_l}\right)^T y$. Using factor score x, missing part of expression matrix is calculated as $y^{miss} = W^{miss}\, x$ where $W = (W^{obs}, W^{miss})$ is a matrix whose columns consist of vectors of observed parts and missing parts. It is quiet time consuming and computationally intensive. It is mostly used for prior missing value estimation for many algorithms because it uses complete information in given dataset.

Based on survey in [18] and [19], a hybrid method of BPCA-iLLS is presented in [20] which take advantages of both methods of exploiting local similarity and lower complexity. Initial dataset is generated by BPCA followed by iLLS used for actual imputation. Time taken by this hybrid method gets doubled than individual BPCA and iLLS.
In Entropy-based selection scheme (EBS) presented in [18], a linear model is proposed which evaluates performance of any imputation method from entropy level. In EBS method of missing value imputation it is observed that performance of different imputation methods is related to complexity of data which is judged by entropy of Eigen values of covariance matrix of expression matrix Y. Complexity of data is calculated using entropy measure defined as

$$e(Y) = \frac{\sum_{i=1}^{\mathscr{k}} \wp_i \, log \, \wp_i}{log \, \mathscr{k}} \qquad (8)$$

Where

$$\wp_i = \sqrt{\lambda_i} \Big/ \sum_{l=1}^{\mathscr{k}} \sqrt{\lambda_l} \qquad (9)$$

Low entropy measure indicates less complexity of data which is known from few distinctly calculated larger Eigen values and large entropy indicates more complex data which cannot be reduced to low dimensional space. Following linear model is fitted on the data set so that performance of any imputation method is evaluated from entropy level.

$$LRMSE\left(\widehat{D}_{j;M_i}^{(\mathscr{k})}, D_j\right) = \alpha_0 + \alpha_i + \beta_i\, e\left(D_j\right) + \gamma_j + \varepsilon_{ijk} \qquad (10)$$

Where $e\left(D_j\right)$ is the entropy level of D_j, α_i and β_i are the intercept and regression slope for imputation method Mi. γ_j is a fixed effect representing the intrinsic imputation difficulty of data set D_j, and ε_{ijk} are random noises and LRMSE is log-transformed root mean squared Error. The linear model (1) is fitted using set $D_{(j)}$ and define the EBS scheme for each data set with intentional missing values $D_j^{(\mathscr{k})}$ as:

$$S_{EBS}\left(D_j^{\mathscr{k}}\right) = M_{EBS}\left(D_j^{(\mathscr{k})}, L\left(D_{(j)}\right)\right) \qquad (11)$$

This method helps in selecting appropriate algorithm for missing value imputation. However it is too much computationally intensive.

In Self-Training Selection (STS) scheme [18], different methods are used for simulation of missing values and rank of each imputation method is calculated in terms of LRMSE of each simulation. The method with smallest rank sum statistics is selected. Rank sum statistic is calculated as follows:

$$R\left(M_i, D_j^{\mathscr{k}}\right) = \sum_{l=1}^{n} Rank_{M_i}\left(LRMSE\left(D_{j;M_i}^{(\mathscr{k})(\ell)}, D_j^{(\mathscr{k})}\right)\right) \qquad (12)$$

And STS scheme is then defined as

$$S_{STS}\left(D_j^{(\mathscr{k})}\right) = \arg min_{M_i}\ R\left(M_i, D_j^{(\mathscr{k})}\right) \qquad (13)$$

In [21], author generalizes SVR which is mainly used for machine learning, for imputation of missing data. Initially gene expressions are trained in such a way that they should not have deviation more than $\mathcal{E}$. The SVR generate such a function which assures the deviation between original and predicted expressions to be $\mathcal{E}$. SVR uses kernel function to transform samples from input space into higher dimension space. Then SVR searches for the global optimal solution to the corresponding problem using the quadratic programming by finding the corresponding support vectors. However performance of SVR depends on Kernel type and optimization techniques used for finding missing value.

The Fixed Rank Approximation Algorithm (FRAA) discussed in [22] is based on optimization algorithm. As estimation of one missing entry influences estimation of other missing entry, all the missing expressions are calculated simultaneously. The new expression matrix Y' is obtained from original matrix Y in such a way that rank of Y' should not exceed d where d is number of singular values of Y. For optimizing the objective function so as to get Y', FRAA is used.

Thus after reviewing above methods it is observed that

- DBI Methods like KNN,SKNN and GO-based KNN
  a. are suited for datasets containing small number of columns
  b. for less number of K (neighbors), performance declines because of more importance given to dominant expressions pattern
  c. performance deteriorates with increasing missing value
- NDBI Methods like SVD, Row-averaging, LLSI, BPCA,EBS, STS, SVR, iLLS
  a. Need complete dataset prior to imputation and their performance depends on method used for temporary estimation
  b. Generally used for prior missing value estimation
  c. Computationally intensive

References

1. Y. Yang, M. Buckley, and S. Dudoit, “Comparison of methods for image Analysis in cDNA microarray data”, *Technical Report No.584*, Department of Statistics, UC Berkeley University,2000
2. M. Ouyang, W. Welsh, P. Georgopoulos, “Gaussian mixture clustering and imputation of microarray data”, *Journal of Bioinformatics*, 20(6):913-923, 2004
3. C. Chiu, S. Chan, C. Wang, and W. Wu, “Missing value imputation for microarray data: a comprehensive comparison study and a web tool”, *BMC System Biology*, 7(6):1-13,2013
4. A. De Brevern, and H. Malpertuy “A: Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering” *BMC Bioinformatics*, 5(114):4272-4279, 2004
5. I. Scheel, M. Aldrin, I. Glad, R. Sorum, H. Lyng, and A. Frigessi, “The influence of missing value imputation on detection of differentially expressed genes from microarray data”, *Bioinformatics*, 21(23):4272-4279, 2005.
6. M. Sehgal, I. Gondal, L. Dooley, and R. Coppel, “How to improve postgenomic knowledge discovery using imputation”, *EURASIP Journal on Bioinformatics and Systems Biology*, 2009(717136):1-14,2009
7. Y. Zhang, J. Xuan, B. Reyes, R. Clarke, and H. Ressom, “Reverse engineering module networks by PSO-RNN hybrid modelling”, *BMC Genomics*, 10(1):1-15, 2009
8. S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, and S. Ishii, “A Bayesian missing value estimation method for gene expression profile data,” *Bioinformatics*, 19(16):2088–2096, 2003

9. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, T. Tibshirani, D. Botstein and R. Altman, “Missing value estimation methods for DNA microarrays”, *International Journal of Bioinformatics*, 17(6):520-525, 2001.
10. A. Gasch, P. Spellman, C. Kao, O. Harel, M. Eisen, G. Storz, D. Botstein and P. Brown, “Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes”, *Journal of Molecular Biology of the cell*, 11(12):4241-4257, 2000
11. K. Kim, B. Kim, and G. Yi, “Reuse of imputed data in microarray analysis increases imputation efficiency”, *International Journal of BMC Bioinformatics*, 5(160):1-9,2004
12. L. Bras, and J. Menezes, “*Improving cluster-based missing value estimation of DNA microarray data*”, International Journal of Biomolecular Engineering, 24(1): 273-282,2007
13. M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, and J. Eppig, “Gene ontology: tool for the unification of biology”, *Journal of Nature Genetics*, 25(1): 25- 29, 2000.
14. A. Alizadeh, M. Eisen, R. Davis, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, and J. Powell, Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Journal of Nature*, 403(6769):503-11, 2000
15. T. Bo, B. Dysvik and I. Jonassen, “LSimpute: accurate estimation of missing values in microarray data with least squares methods”, *Nucleic Acids Res*. e34, 32(3):1-8,2004
16. K. Hyunsoo, G. Golub and P. Haesun, “*Missing value estimation for DNA microarray gene expression data: local least squares imputation*”, *Journal of Bioinformatics*,21(2):187-198,2005
17. X. Zhang, X. Song, and H. Wang, “Sequential local least squares imputation estimating missing value of microarray data”, *International Journal of Computers in Biology and Medicine*, 38(1):1112-1120,2008
18. G. Brock, J Shaffer, R. Blakesley, M. Lotz and G. Tseng, “Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes”, *BMC Bioinformatics*, 9(12):1-12,2008
19. L. Bras and J. Menezes, “Dealing with gene expression missing data,” *IEEE Transaction on Systems Biology*, 153(3):105–119, 2006.
20. F. Shi, D. Zhang, J. Chen and H. Karimi, “Missing Value Estimation for Microarray Data by Bayesian Principal Component Analysis and Iterative Local Least Squares”, *Mathematical Problems in Engineering*, 2013(162938):1-5, 2013
21. X. Wang, A. Li, Z. Jiang, and H. Feng, “Missing value estimation for DNA microarray gene expression data by support vector regression imputation and orthogonal coding scheme”, *BMC Bioinformatics*,7(32), 2006
22. S. Friedland, A. Niknejad, and L. Chihara, “*A simultaneous reconstruction of missing data in DNA microarrays”*, *Journal of Linear Algebra and its applications*, 416(1):8-28,2006
23. P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, P. Brown, D. Botstein and B. Futcher, “Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization,” *Molecular Biology of the Cell*, 9(12):3273-3297, 1998