A full analysis of imputation procedures for Affymetrix gene expression datasets

Sreevidya Sadananda Sadasiva Rao[2], Lori Shepherd[2] , Andrew Bruno[3] , Song Liu[2], Jeffrey C. Miecznikowski*[1,2]

[1] Department of Biostatistics; University at Buffalo; Phone (716) 881.8953; Fax (716) 829.2200; Buffalo, NY 14214 USA;
[2] Department of Biostatistics; Roswell Park Cancer Institute; Buffalo, NY 14263 USA
[3]Center for Computational Research,University at Buffalo,NYS Center of Excellence in Bioinformatics and Life Sciences,Buffalo, New York 14203.

Email: Sreevidya Sadananda Sadasiva Rao - sreevidy@buffalo.edu; Lori Shepherd - las65@buffalo.edu; Andrew Bruno - aebruno2@buffalo.edu; Song Liu - Song.Liu@roswellpark.org; Jeffrey C. Miecznikowski - jcm38@buffalo.edu;

*Corresponding author

Abstract

Background: Microarray technology makes possible the monitoring of gene expression on a genomewide scale and has been widely applied to detect gene activity changes in many areas of biomedical research. However, due to the complexities of the microarray process, the expression data of individual genes may be missing due to flaws in the array and background noise. The microarray datasets on well-characterized RNA samples from MicroArray Quality Control (MAQC) project has enabled the assessment of the precision and comparability of microarrays, as well as the strengths and weakness of various microarray analysis methods. However, to date few studies have reported the performance of missing value imputation schemes on the MAQC datasets. In this study, we use the Affymetrix data sets generated by the MAQC projects to evaluate various imputation procedures in single color microarray platform.

Results Using the MAQC data, we evaluated several imputation procedures (BPCA, KNN, LLS, LSA, NIPALS, SVD, Row average), comparing them using five error measures (RMSE, LRMSE, NRMSE, RAE, RAEL2). We randomly deleted 5% and 10% of the data and imputed the missing values using these imputation tests. We performed a 1000 simulations and averaged the results. The results for both 5% and 10% deletion are similar. Among all the imputation methods, we observe that LLS with $k = 4$ has the lowest value across all the error measures. KNN with $k = 1$ has the highest value of all the imputation methods for all the error measures.

Conclusions: Based on our study we conclude that, for imputing missing values in Affymetrix microarray datasets, using the MAS 5.0 pre-processing scheme, local least squares method with $k = 4$ has the best overall performance and k nearest neighbour method with $k = 1$ has and worst overall performance. These results hold true for both 5% and 10% missing values. These conclusions are based on technical datasets and without any downstream analyses.

Background

In microarray experiments randomly missing values may occur due to scratches on the chip, spotting errors, dust, or hybridization errors. Other non random missing values may be biological in nature, for example, probes with low intensity values or intensity values that may exceed a readable threshold. These missing values will create incomplete gene expression matrices where the rows refer to genes and the columns refer to samples. These incomplete expression matrices will make it difficult for researchers to perform downstream analyses such as differential expression inference,clustering or dimension reduction methods like principal components analysis or multidimensional scaling. Hence, it is critical to understand the nature of the missing values and to choose an accurate method to impute the missing values.

There have been several methods put forth to impute missing data in microarray experiments. In one of the early papers related to microarrays, [1] examines several methods of imputing missing data and ultimately suggests a k-nearest neighbors approach. Researchers also explored applying previously developed schemes for microarrays such as the nonlinear interative partial least squares (NIPALS) as discussed in [2]. A Bayesian approach for missing data in gene expression microarrays is provided in [3]. Other approaches such as [4] suggest using least squares methods to estimate the missing values in microarray data, while [5] suggests using a local least squares imputation. A Gaussian mixture method for imputing missing data is proposed in [6].

While many of these approaches can be generally applied to different platforms of gene expression arrays, we will focus on applying these methods to Affymetrix gene expression arrays , one of the most popular arrays in scientific research. In Affymetrix gene expression arrays, before studying missing data imputation schemes, we need to remove any missing values. A detection call algorithm is used to filter and remove missing expression values based on absent/present calls [7]. Subsequently, a pre-processing scheme is employed. There are numerous tasks to perform in pre-processing Affymetrix arrays, including background adjustment, normalization, and summarization. A nice overview of the methods available for preprocessing is provided in [8]. The detection call employs MAS 5.0 [9] to obtain expression values, hence it is reasonable to use MAS 5.0 as our pre-processing method.

Naturally, when proposing a new imputation scheme for Affymetrix expression arrays, it is necessary to compare the new method against existing methods. Several excellent papers have compared missing data procedures on high throughput data platforms such as in two-dimensional gel electrophoresis as in [10] or gene expression arrays [11–13].

For our analysis, we choose to focus on the microarray quality control (MAQC) datasets, where the datasets have been specifically designed to address the strengths and weaknesses of various microarray analysis methods.

The MAQC datasets were designed by the U.S. Food and Drug Administration to provide quality control (QC) tools to the microarray community to avoid procedural failures. The project aimed to develop guidelines for microarray data analysis by providing the public with large reference datasets along with readily accessible reference RNA samples. Another purpose of this project was to establish QC metrics and thresholds for objectively assessing the performance achievable by various microarray platforms. These datasets were designed to evaluate the advantages and disadvantages of various data analysis methods. The initial results from the MAQC project were published in [14] and later in [15] and [16]. Specifically, the MAQC experimental design for Affymetrix gene expression HG-U133 Plus 2.0 GeneChips includes six different test sites, four samples per site, five replicates per site, for a total of 120 arrays (see Methods). This rich data set provides an ideal setting for evaluating imputation methods on Affymetrix expression arrays. While this dataset has been mined to determine inter-intra platform reproducibility of measurements, to our knowledge, no one has studied imputation methods on this dataset.

The MAQC dataset hybridizes two RNA sample types - Universal Human Reference RNA (UHRR) from Stratagene and a Human Brain Reference RNA (HBRR) from Ambion. These two reference samples and varying mixtures of these samples constitute the four different samples included in the MAQC dataset. By using various mixtures of the samples, this data set is designed to study technical variation present in this technology. By technical variation, we are referring to the variability between preparations and labelling of sample, variability between hybridizations of the same sample to different arrays, and variability between the signal on replicate features of the same array. Meanwhile biological variability refers to variability between individuals in population and is independent of the microarray process itself. This key feature will allow us to examine the accuracy of the imputation procedures without the confounding feature of biological variability. These technical datasets are commonly used to evaluate different analysis methods specific to Affymetrix microarrays like methods for identifying differentially expressed genes [17–19].

In summary, our analysis examines cutting edge imputation schemes on an Affymetrix technical dataset devoid of biological variation. The Methods section discusses the MAQC dataset and the proposed imputation schemes. Meanwhile, the Results section describes the results from applying the imputation methods for addressing missingness in the MAQC datasets. Finally, we conclude our manuscript with a Discussion and Conclusion.

Methods

Data Sets

The MAQC dataset is fully described in [14]. The MAQC dataset hybridizes two RNA samples, a Universal Human Reference RNA (UHRR) from Stratagene and a Human Brain Reference RNA (HBRR) from Ambion. From these two samples, four pools are created, the two reference RNA samples as well as two mixtures of the original samples: Sample A, 100% UHRR; Sample B, 100% HBRR; Sample C, 75% UHRR:25% HBRR; and Sample D, 25% UHRR:75% HBRR. Both Sample A and Sample B are commercially available and biologically different where we expect a large number of differentially expressed genes between Sample A and Sample B.

There are six different test sites where each test site assayed all four samples with five replicates per sample. Thus, for each test site there is a total of 30 arrays and a total of 120 arrays over all the six sites. The data is examined for all four samples individually within each site and across the six sites.The data from each site for each sample is preprocessed separately giving us “site datasets” or separate datasets for each site.

Missing Values and Detection Call Algorithm

Using MAS 5.0, a detection call algorithm is used to flag the missing values [9]. The detection call determines if the transcript of a gene is present or absent in the sample. For every gene, the microarray chip has probes that perfectly match to a segment of the gene sequence (PM probes) and probes that contain a single mismatched nucleotide in the center of the perfect match probe (MM probes). The difference in the intensity of the perfect and mismatch probes is used to make detection calls.

The detection call algorithm is summarized in [7]. First discrimination scores are calculated for each probe set from the raw intensity data. Each probe set has 11 to 20 probe pairs. For each probe pair, the ratio of the sum and difference of the PM and MM probes gives the discrimination score for that probe pair. This score is calculated for all the probe pairs in a probe set. The null hypothesis is that the median discrimination score of a probe set is equal to τ and the alternate hypothesis is that the median discrimination score is greater than τ, where τ is defined as a small non negative number which can be changed by the user to adjust the specificity and sensitivity. One sided Wilcoxon rank sum test is used to obtain p-values for the hypotheses test of each probe set. Two significance levels α_1 and α_2 are cutoffs of p-values for detection calls. A present call is made for a gene with p-value $< \alpha_1$,an absent call for a gene with p-value $\geq \alpha_2$ and a marginally detected call for a gene for $\alpha_1 \leq$ p-value $< \alpha_2$. Depending on the detection call, we can determine if the target transcript is present or absent in the sample.

Percent present

Using the detection call algorithm, each probe set is identified as absent/present/marginal.The SimpleAffy package has methods for quality control metrics on Affymetrix arrays [20]. One metric is percent present calls. This method calculates the percentage of present probes in each array. Using this method we calculate the percent present calls for all 120 arrays separately and then average the percentages over the 5 replicates for each sample and each site. The details are summarized in Table 1.

We have used the “mas5calls” function detailed in [21] from the affy package [9] to determine the detection calls. This method employs the MAS 5.0 algorithm to obtain the expression values. The default value for τ is 0.015 , for $\alpha 1$ is 0.04 and $\alpha 2$ is 0.06. Using this method we get a P/M/A value (present/marginal/absent call)for each probe in all the 120 arrays. For every sample, probes were filtered based on the present calls where probes that were detected present in all five replicates of a given sample were retained for further analyses. Probes that were detected as absent in one or more replicates of a sample were removed.

Preprocessing algorithms

The full matrix has only probes which were identified as present in all 5 replicates for each sample. We now preprocess each full matrix using MAS 5.0 to obtain expression values for further analyses. The datasets were preprocessed using MAS 5.0 available in the Bioconductor suite of libraries [22]. The MAS 5.0 preprocessing was implemented using the R language “affy” library [9].

Preprocessing algorithms for Affymetrix gene expression microarrays are necessary in order to account for the systematic variation present in array technology and to summarize the signal for each gene which is measured via a series of multiple short probes. As discussed in [8], preprocessing schemes can be organized into three steps, a background adjustment step, a normalization step and a summarization step. In short, the MAS 5.0 preprocessing algorithm is outlined in the Statistical Algorithms Description Document [21] and used in the MAS 5.0 software [21]. The steps in MAS 5.0 involve 1) a weighted nearest neighbor approach is used to estimate and remove the background signal, 2) a normalization that scales all arrays to a baseline array, and 3) a summarization step using the ideal mismatch.

The result of preprocessing is the complete expression matrix or expression dataset containing no missing values. To compare imputation methods, we randomly remove a percentage of the datapoints from the complete expression matrix and compare the results between the complete dataset and the dataset(s) with simulated missing genes. We examine randomly removing 5% and 10% of the data points from the complete expression matrix. We repeat the procedure 1000 times for each percentage of missing data.

Missing Value imputation methods

Similar to the work presented in [13] for the MAQC data, we examine the following missing data analysis methods :

1. row average (Row),

2. k nearest neighbors using euclidean distance or Pearson correlation, with k=1, and 5, where k is the number of neighbors used in the imputation(KNN),

3. singular value decomposition (SVD) [1],

4. Least squares adaptive (LSA) [4],

5. Local Least squares (LLS),choosing k=1,3, and 4; k is the cluster size which is the number of similar genes used for regression [5],

6. Bayesian PCA (BPCA) Bayesian principal components [3],

7. noniterative partial least squares (NIPALS) [2].

Note, the row average (ROW) and k nearest neighbor (KNN) imputation were done using the R computing language with the *impute* package [23] while LSA was implemented using the java language code [24]. In the ROW method, the average of the values that are present for that particular gene are used to replace the missing data points. The KNN algorithm classifies objects based on closest ("nearest") genes. In this algorithm we find the k nearest neighbors using a suitable distance metric, and then we impute the missing elements by averaging those (non-missing) elements of its neighbors. In the KNN method, there are different types of distance metrics (Pearson correlation, Euclidean, Mahalonobis, and Chebyshev distance) that can be employed. We chose the Euclidean distance metric as it has been reported to be more accurate [25]. We used different values for $k = 1, 5$ for the KNN method.

The least squares method (LSM) estimates missing values utilizing correlations between probes and arrays. There are several variants of the LSM described fully in [4], where each variation is related to different methods of estimating the correlation within the dataset. LSA uses an adaptive procedure for weighting the gene method and array method estimates. The LSA method was implemented using the LSimpute.jar java script available at http://www.ii.uib.no/~trondb/imputation/.

The LLS method is a neighbouring based approach,that selects neighbours based on Pearson correlation. Multiple regression is performed using k nearest neighbours as described in [5]. It is implemented using the R package "pcaMethods" [26]. The method restricts k to be less than number of replicates/columns. In our case, with 5 replicates, we chose k equal to 1,3, or 4. Global based methods SVD [1] and BPCA [3] were implemented using the R package pcaMethods [26]. The NIPALS method is summarized in [2] and is implemented using the R package "pcaMethods" [26]. Similar to KNN, in order to implement the NIPALS algorithm, it is necessary for the user to specify the number of principal components. To evaluate the different methods of imputation, gene expresssion values were randomly deleted across groups from the complete dataset, and the summary measures described below were calculated and compared across the methods.

Quantitative Error Evaluation

After the preprocessing of the MAQC datasets, we obtain a gene expression matrix, where the rows correspond to genes, and the colums correspond to samples. Similar to [13], we denote this complete gene expression matrix as $CD = (y_{gs})_{G \times S}$ where y_{gs} is the expression intensity of gene g on sample s. To simulate the missing data, we randomly remove 5%, or 10% of the datapoints in matrix CD. Given a missing value imputation scheme, the missing value for gene g, sample s is imputed as $\hat{y}_{gs}$ and the imputed dataset is denoted as ID.

To compare the imputed dataset ID with the complete dataset CD, we employed the following summary statistics:

1. Root mean squared error (RMSE),

$$RMSE = \sqrt{\frac{1}{\text{\# of missing}} \sum_{y_{gs} \text{ missing}} (\hat{y}_{gs} - y_{gs})^2}, \tag{1}$$

2. Normalized Root mean squared error (NRMSE) [1],

$$NRMSE = \frac{\sqrt{\frac{1}{\text{\# of missing}} \sum_{y_{gs} \text{ missing}} (\hat{y}_{gs} - y_{gs})^2}}{\frac{1}{G*S} \sum_g \sum_s y_{gs}}, \tag{2}$$

3. Relative estimation error (RAE) [25],

$$RAE = \frac{1}{\text{\# of missing}} \sum_{y_{gs} \text{ missing}} \frac{|\hat{y}_{gs} - y_{gs}|}{\phi(y_{gs})}, \tag{3}$$

   where,

$$\phi(y_{gs}) = \begin{cases} |y_{gs}|, & \text{if } |y_{gs}| > \epsilon \\ \epsilon, & \text{if } |y_{gs}| < \epsilon. \end{cases} \tag{4}$$

4. logged RMSE (LRMSE) [11],

$$LRMSE = \sqrt{\frac{1}{\text{\# of missing}} \sum_{x_{gs} \text{ missing}} (\hat{x}_{gs} - x_{gs})^2}, \tag{5}$$

   where $\hat{x}_{gs} = \log(\hat{y}_{gs})$,

5. RAE-L2 [13],

$$RAE - L2 = \sqrt{\frac{1}{\text{\# of missing}} \sum_{(y_{gs} \text{ missing})} \frac{(\hat{y}_{gs} - y_{gs})^2}{y_{gs}}}. \tag{6}$$

To understand the variability in the imputation procedures, we repeat each missing data simulation 1000 times.

Results

We are summarizing our findings in terms of detection call algorithm results and imputation results.

Detection Call Algorithm Results

Using the detection call algorithm, we obtain the percentage of present calls and the present probes for all the samples.

Examining Table 1, the percentage of present calls are comparable among the different samples and sites. The percent present values are between 51% and 58.5% and hence we can compare between different

	Site 1	Site 2	Site 3	Site 4	Site 5	Site 6	Average
Sample A	53.30	52.95	54.83	56.09	55.03	52.46	54.11
Sample B	52.10	51.16	53.65	54.17	52.98	51.82	52.65
Sample C	55.12	54.51	57.92	56.38	57.27	55.19	56.06
Sample D	55.65	54.23	55.53	58.58	56.64	54.97	55.93
Average	54.04	53.21	55.48	56.30	55.48	53.61	54.69

Table 1: Average Percent present calls for each sample in each site

samples.The percent present calls are also represented graphically, per site and per sample in Figure 1. From the figure we see that Site 4 has the highest present calls and Site 2 has the lowest present calls. In terms of samples, Sample B has the lowest present calls.

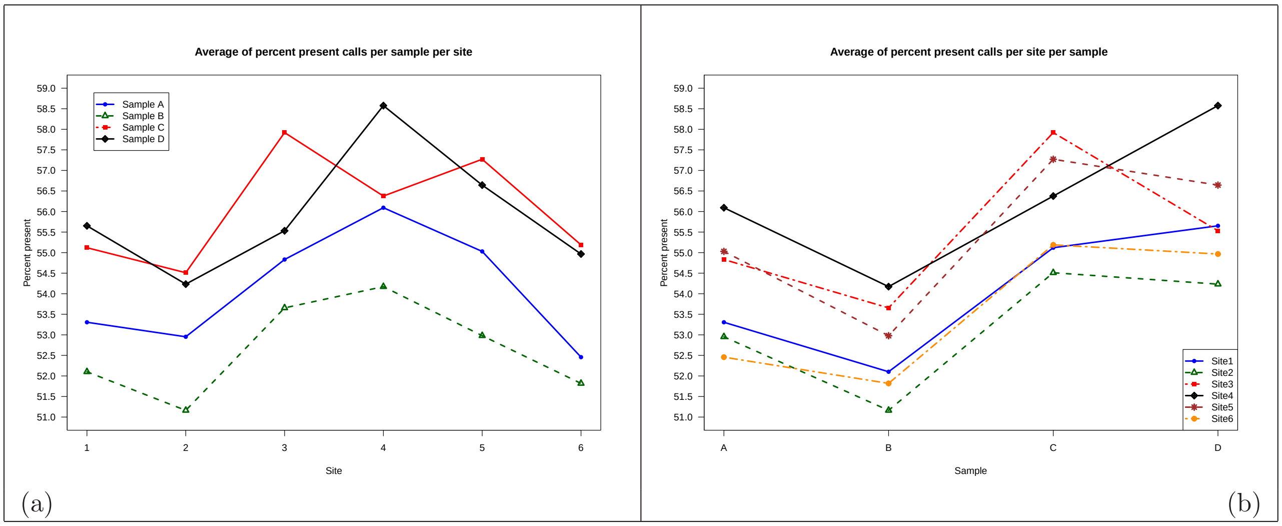

Figure 1: The percent present calls are calculated for all the 5 replicates for each sample and site combination. We average percent present calls over the 5 replicates to give a single value per sample per site. The average percent present calls is plotted in the graphs. Figure (a) represents the average of percent present calls for each site across the 4 samples. The y axis has the average percent present values and the x axis has the different test sites. The four curves represent the 4 samples. Figure (b) represents the average of percent present calls for each sample across the 6 test sites. The y axis has the average percent present values and the x axis has 4 samples. The six curves represent the 6 test sites.

The Affymetrix HG-U133 chip has 54675 probes. After filtering the absent calls, the number of present probes ranges from 22,900 (Sample B, Site 2) to 27,021 (Sample C, Site 3). Table 2 gives the number of probes identified as present for each sample and each site. These data represent a full matrix with no missing values. The full matrix for each setting represents the starting point for subsequent analyses.

	Sample A	Sample B	Sample C	Sample D
Site 1	24184	23557	25163	25318
Site 2	24202	22900	24416	24631
Site 3	25471	24788	27021	25232
Site 4	25587	24454	26092	25877
Site 5	25281	24335	26352	26152
Site 6	24552	24096	25887	26004

Table 2: Number of probes present in all 5 replicates for each site and sample combination

Imputation Results

We rank the imputation methods (IM) based on its average performance across the different error measures, across all sample types and all sites. The ranking procedure is carried out separately for 5% and 10% deletion. We carry out 1000 simulations for each site and sample combination, a total of 24 times (6 Sites and 4 Samples). Each simulation in turn consists of applying the 5 error measures on each of the 10 imputation methods. Averaging over the 1000 simulations we get a single value for each imputation method/error measure combination for every site/sample combination. For example, for the metric RMSE there are 10 unique values; one for each imputation method at every site/sample combination. Then we rank the 10 IM based on each error measure separately at each site/sample combination. For example, based on RMSE values, the IM are ranked from the lowest to highest, the IM with lowest RMSE value is 1 and highest is 10. The IM each have a rank value at each site for each one of the 4 samples. For every imputation method, the RMSE based rank values are averaged across the six sites for each sample, thus we obtain 4 average rank values, one for each sample. Finally we average these 4 rank values to obtain a single number, which reflects the RMSE based rank of the given imputation method across all sites and all sample types for a given deletion percentage. Similar to RMSE, the IM are ranked based on LRMSE, NRMSE, RAE and RAEL2 values. Thus each imputation method has 5 different average rank values based on the 5 error measures. The IM ranked lowest and highest for each error metric for 5% deletion are shown in Tables 5 to 9 and the IM for 10% deletion are shown in Tables 10 to 14. Based on this ranking, the results are identical for both 5% and 10% deletion. RMSE and NRMSE metrics suggest that LSA imputation method has the best performance. With LRMSE and RAEL2 metrics, ROW is the best performing imputation method. The imputation method LLS with $k = 4$ has the best performance when we use the RAE metric. KNN with $k = 1$ is the worst performing imputation scheme with all the five error measures across all samples and all sites.

Taking our ranking a step further, for each imputation method, we take the average of the 5 rank values across the five error measures. This gives a global ranking to every imputation method which reflects its overall performance across different error measures, across all sites and samples for a given deletion percentage. The final results are displayed in Tables 3 and 4 for 5% and 10% deletion, respectively. From the tables we observe that KNN with $k = 1$ has the highest value for any given error measure, thus it is the worst performing imputation method. LLS with $k = 4$ has the overall best performance across the different error measures. These results hold true for both 5% and 10% deletion.

error metric	RMSE	LRMSE	NRMSE	RAE	RAEL2	Average
bpca	2.38	5.5	1.08	5.13	5.63	3.94
knn1	**9.79**	**9.88**	**9.75**	**9.83**	**9.79**	**9.81**
knn5	7.83	9.08	7.83	8.33	8.42	8.3
lls1	5.17	4.29	3.04	4.67	4.37	4.31
lls3	3.83	2.17	1.21	2.12	2.17	2.3
lls4	**2.79**	**2.29**	**1.08**	**1.96**	**2**	**2.02**
lsa	1	4.87	1	4.33	4.71	3.18
nipals	7.25	7.33	7.17	7.33	7.33	7.28
row	6	1.5	5.58	2	1.58	3.33
svd	8.96	8.29	8.87	8.29	8.33	8.55

Table 3: Summary table for 5% deletion - each imputation is ranked based on its average performance across the different error measures, across all samples and all sites. Rows correspond to Imputation methods and columns correspond to error measures. Each imputation method is ranked based on its average performance across all samples and all sites. From the table we observe that RMSE and NRMSE measures, suggest LSA imputation method has the best performance. With LRMSE and RAEL2 measures, ROW is the best imputation method. LLS with $k = 4$ (lls4) has the best performance when we use the RAE measure. KNN with $k = 1$ (knnl) has the highest rank value for any given error measure, thus it is the worst performing imputation method. LLS with $k = 4$ (lls4) has the overall best performance across the different error measures.

Figures 2 to 11 represent the performance of different imputation methods as measured by the 5 error measures for all the sample and site combinations. Results from 5% deletion and 10% deletion show a similar pattern but the actual values vary. The imputed values and variance of 10% missing data are larger than 5% missing data. The imputation methods exhibit different outcomes for some samples and sites but perform similarly at others. Site 4 has the highest values for most of the imputation tests for all the samples. Results of the imputation schemes, in terms of the best and worst performance, is similar for samples A, B and C but it is different for sample D.

error	RMSE	LRMSE	NRMSE	RAE	RAEL2	Average
metric						
bpca	2.33	5.71	1.13	5.46	5.67	4.06
knn1	**9.79**	**10**	**9.79**	**10**	**9.88**	**9.89**
knn5	7.83	8.87	8.5	8.75	8.79	8.55
lls1	5.21	4.25	3.62	4.5	4.33	4.38
lls3	3.75	2.25	1.33	2.17	2.17	2.33
lls4	**2.92**	**2.92**	**1.21**	**1.96**	**2.08**	**2.22**
lsa	1	4.92	1	4.5	4.71	3.22
nipals	7.25	6.96	7.08	7.08	7.04	7.08
row	5.96	1.5	5.46	2	1.46	3.27
svd	8.96	8.08	8.5	8.17	8.29	8.40

Table 4: Summary table for 10% deletion - each imputation is ranked based on its average performance across the different error measures, across all samples and all sites. Rows correspond to Imputation methods and columns correspond to error measures. Each imputation is ranked based on its average rank performance across all samples and all sites. From the table we observe that RMSE and NRMSE measures, suggest LSA imputation method has the best performance. With LRMSE and RAEL2 measures, ROW is the best imputation method. LLS with $k = 4$ (lls4) has the best performance when we use the RAE measure. KNN with $k = 1$ (knnl) has the highest rank value for any given error measure, thus it is the worst performing imputation method. LLS with $k = 4$ (lls4) has the overall best performance across the different error measures.

Discussion

The MAQC project allows researchers to study a variety of microarray aspects including comparisons of one color and two color arrays [27], reproducibility [14, 15, 28], removal of batch effects [29], and determining differentially expressed genes [30]. From this diverse research, it is clear that the MAQC projects represent a fertile testing ground for microarray inspired algorithms and methods. However, to date, we are not aware of any work examining imputation methods on the MAQC datasets.

Here we study imputation methods while using only the MAS 5.0 as preprocessing method. However there are other pre-processing schemes such as RMA [31–33], GCRMA [34], that are routinely used and these methods may influence the imputation scheme performance differently. We highlight several works that extensively study pre-processing schemes for Affymetrix datasets including [17, 18, 35, 36].We study the various cutting edge imputation schemes on the MAQC datasets and compare them with the results from similar missing data imputation manuscripts such as [10–13]. While our summary measures are important to compare the imputation schemes, it is not clear how the different imputation procedures will affect downstream biological analysis and interpretation. In the work of [13], they introduce biological measures to compare imputation procedures. Their biological measures are designed to study the clustering and classification schemes commonly applied to gene expression microarrays.

We recognize that the MAQC datasets are not without their criticisms. For example, the issue of choosing an overall optimal preprocessing scheme is still somewhat of an open question [37]. Another serious criticism is provided in [38] with a reply in [39]. In that discussion one of the main concerns involves technical versus biological variation. This important issue has arisen when studying other "technical" microarray datasets [35]. Considering both aspects of this question, if we use datasets containing biological and technical variation, that is, datasets designed to answer biological questions then there are biases due to the intent of the original datasets (e.g. biological variation of the species, sample preparation, procurement of RNA, hybridization affinities).

Conclusions

In our work, we use the MAQC datasets with the MAS 5.0 pre-processing scheme to compare missing data imputation schemes. The best and worst performing imputation schemes remain the same for both 5% and 10% deletion. We observe that k nearest neighbour method with $k = 1$ has worst performance among the imputation schemes across all error measures for both deletion percentages. Local least squares (LLS) method with $k = 4$ gives the best performance for imputing missing values across all error measures for both 5% and 10% deletion. These conclusions are based on technical datasets and without any downstream analyses.

Missing values in microarray experiments is a common problem with effects on downstream analysis. Many variables such as the biological variability of the data set, experimental conditions of the study, percentage of missing values and type of downstream analysis performed need to be considered when chosing an imputation method. We used a small subset of these conditions to examine these imputation methods. There is further scope for additional study under different settings or combination of settings.

Author's contributions

JCM and SL designed the study. SSSR performed the statistical analysis. SSSR and JCM wrote the manuscript. LS and AB assisted with the data analysis, and assisted in writing the manuscript. All authors read and approved the final manuscript.

References

1. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman R: **Missing value estimation methods for DNA microarrays**. *Bioinformatics* 2001, **17**(6):520.
2. Wold H: **Path models with latent variables: the NIPALS approach**. *Quantitative sociology: International perspectives on mathematical and statistical modeling* 1975, :307–357.
3. 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.
4. Bø T, Dysvik B, Jonassen I: **LSimpute: accurate estimation of missing values in microarray data with least squares methods**. *Nucleic Acids Research* 2004, **32**(3):e34.
5. Kim H, Golub G, Park H: **Missing value estimation for DNA microarray gene expression data: local least squares imputation**. *Bioinformatics* 2005, **21**(2):187.
6. Ouyang M, Welsh W, Georgopoulos P: **Gaussian mixture clustering and imputation of microarray data**. *Bioinformatics* 2004, **20**(6):917.
7. Mei R, Di X, Ryder T, Hubbell E, Dee S, Webster T, Harrington C, Ho M, Baid J, Smeekens S, et al.: **Analysis of high density expression microarrays with signed-rank call algorithms**. *Bioinformatics* 2002, **18**(12):1593–1599.
8. Gentleman R, Carey V, Huber W, Irizarry R, Dudoit S: **Bioinformatics and Computational Biology Solutions Using R and Bioconductor (Statistics for Biology and Health)** 2005.
9. Gautier L, Cope L, Bolstad B, Irizarry R: **affy.analysis of Affymetrix GeneChip data at the probe level**. *Bioinformatics* 2004, **20**(3):307–315.
10. Miecznikowski J, Damodaran S, Sellers K, Coling D, Salvi R, Rabin R: **A comparison of imputation procedures and statistical tests for the analysis of two-dimensional electrophoresis data**. *Proteome Science* 2011, **9**:14.
11. Brock G, Shaffer J, Blakesley R, Lotz M, Tseng G: **Which missing value imputation method to use in expression profiles: a comparative study and two selection schemes**. *BMC bioinformatics* 2008, **9**:12.
12. Celton M, Malpertuy A, Lelandais G, De Brevern A: **Comparative analysis of missing value imputation methods to improve clustering and interpretation of microarray experiments**. *BMC genomics* 2010, **11**:15.
13. Oh S, Kang D, Brock G, Tseng G: **Biological impact of missing-value imputation on downstream analyses of gene expression profiles**. *Bioinformatics* 2011, **27**:78.
14. Shi L, Reid L, Jones W, Shippy R, Warrington J, Baker S, Collins P, De Longueville F, Kawasaki E, Lee K, et al.: **The MicroArray Quality Control (MAQC) project shows inter-and intraplatform reproducibility of gene expression measurements**. *Nature biotechnology* 2006, **24**(9):1151–1161.
15. Chen J, Hsueh H, Delongchamp R, Lin C, Tsai C: **Reproducibility of microarray data: a further analysis of microarray quality control(MAQC) data**. *BMC bioinformatics* 2007, **8**:412.
16. Shi L, Jones W, Jensen R, Harris S, Perkins R, Goodsaid F, Guo L, Croner L, Boysen C, Fang H, et al.: **The balance of reproducibility, sensitivity, and specificity of lists of differentially expressed genes in microarray studies**. *BMC bioinformatics* 2008, **9**(Suppl 9):S10.
17. Choe S, Boutros M, Michelson A, Church G, Halfon M: **Preferred analysis methods for Affymetrix GeneChips revealed by a wholly defined control dataset**. *Genome biology* 2005, **6**(2):R16.
18. Zhu Q, Miecznikowski J, Halfon M: **Preferred analysis methods for Affymetrix GeneChips. II. An expanded, balanced, wholly-defined spike-in dataset**. *BMC bioinformatics* 2010, **11**:285.
19. Zhu Q, Miecznikowski J, Halfon M: **A wholly defined Agilent microarray spike-in dataset**. *Bioinformatics* 2011, **27**(9):1284.
20. Wilson C, Miller C: **Simpleaffy: a BioConductor package for Affymetrix Quality Control and data analysis**. *Bioinformatics* 2005, **21**(18):3683–3685.
21. Affymetrix I: **Statistical algorithms description document**. *Technical paper* 2002.

22. Gentleman R, Carey V, Bates D, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, et al.: **Bioconductor: open software development for computational biology and bioinformatics**. *Genome biology* 2004, **5**(10):R80.

23. Hastie T, Tibshirani R, Narasimhan B, bert Chu G: *impute: impute: Imputation for microarray data* 1999. [R package version 1.10.0].

24. Bø T, Dysvik B, Jonassen I: **LSimpute: Accurate estimation of missing values in microarray data with least squares methods**. *http://www.ii.uib.no/~trondb/imputation/* 2005.

25. Nguyen D, Wang N, Carroll R: **Evaluation of missing value estimation for microarray data**. *Journal of Data Science* 2004, **2**(4):347–370.

26. Stacklies W, Redestig H, to Kevin Wright for improvements to nipalsPca T: *pcaMethods: A collection of PCA methods.* 2007. [R package version 1.18.0].

27. Patterson T, Lobenhofer E, Fulmer-Smentek S, Collins P, Chu T, Bao W, Fang H, Kawasaki E, Hager J, Tikhonova I, et al.: **Performance comparison of one-color and two-color platforms within the MicroArray Quality Control (MAQC) project**. *Nature biotechnology* 2006, **24**(9):1140–1150.

28. Wen Z, Wang C, Shi Q, Huang Y, Su Z, Hong H, Tong W, Shi L: **Evaluation of gene expression data generated from expired Affymetrix GeneChip® microarrays using MAQC reference RNA samples**. *BMC bioinformatics* 2010, **11**(Suppl 6):S10.

29. Luo J, Schumacher M, Scherer A, Sanoudou D, Megherbi D, Davison T, Shi T, Tong W, Shi L, Hong H, et al.: **A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data**. *The pharmacogenomics journal* 2010, **10**(4):278–291.

30. Kadota K, Shimizu K: **Evaluating methods for ranking differentially expressed genes applied to MicroArray Quality Control data**. *BMC bioinformatics* 2011, **12**:227.

31. Bolstad B, Irizarry R, Åstrand M, Speed T: **A comparison of normalization methods for high density oligonucleotide array data based on variance and bias**. *Bioinformatics* 2003, **19**(2):185–193.

32. Irizarry R, Bolstad B, Collin F, Cope L, Hobbs T B a nd Speed: **Summaries of Affymetrix GeneChip probe level data**. *Nucleic acids research* 2003, **31**(4):e15.

33. Irizarry R, Hobbs B, Collin F, Beazer-Barclay Y, Antonellis K, Scherf U, Speed T: **Exploration, normalization, and summaries of high density oligonucleotide array probe level data**. *Biostatistics* 2003, **4**(2):249–264.

34. Wu Z, Irizarry R, Gentleman R, Martinez-Murillo F, Spencer F: **A model-based background adjustment for oligonucleotide expression arrays**. *Journal of the American Statistical Association* 2004, **99**(468):909–917.

35. Dabney A, Storey J: **A reanalysis of a published Affymetrix GeneChip control dataset**. *Genome Biol* 2006, **7**(3):401.

36. Gaile D, Miecznikowski J: **Putative null distributions corresponding to tests of differential expression in the Golden Spike dataset are intensity dependent**. *BMC genomics* 2007, **8**:105.

37. Perkel J: **Six things you won't find in the MAQC**. *The Scientist* 2007, **20**(11):68.

38. Liang P: **MAQC papers over the cracks**. *Nature biotechnology* 2007, **25**:27–28.

39. Shi L, Jones W, Jensen R, Wolfinger R, Kawasaki E, Herman D, Guo L, Goodsaid F, Tong W: **Reply to MAQC papers over the cracks**. *Nature Biotechnology* 2007, **25**:28–29.

Tables

	Imputation : LSA, error metric : RMSE					**Imputation : KNN1, error metric : RMSE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	1		10	10	10	9	
Site 3	1	1	1	1		10	10	10	9	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	1		10	10	10	9	
Site 6	1	1	1	1		10	10	10	9	
Avg	1	1	1	1	1	10	10	10	9.17	9.79

Table 5: Imputation test Ranking based on RMSE error metric with 5% deletion - After 5% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the RMSE measure. By performing 1000 simulations and averaging, we obtain a single RMSE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the RMSE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the RMSE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and RMSE measure. With RMSE, LSA has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

	Imputation : ROW, error metric : LRMSE					**Imputation : KNN1, error metric : LRMSE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	5		10	10	10	10	
Site 3	1	1	3	3		10	10	10	10	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	3		10	10	10	9	
Site 6	1	1	1	3		10	10	10	9	
Avg	1	1	1.33	2.67	1.50	10	10	10	9.5	9.88

Table 6: Imputation test Ranking based on LRMSE error metric with 5% deletion - After 5% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the LRMSE measure. By performing 1000 simulations and averaging, we obtain a single LRMSE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the LRMSE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the LRMSE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and LRMSE measure. ROW has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively

	Imputation : LSA, error metric : NRMSE					**Imputation : KNN1, error metric : NRMSE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	1		9	10	10	9	
Site 3	1	1	1	1		10	10	10	9	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	1		10	10	10	9	
Site 6	1	1	1	1		10	10	10	9	
Avg	1	1	1	1	1	9.83	10	10	9.17	9.75

Table 7: Imputation test Ranking based on NRMSE error metric with 5% deletion - After 5% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the NRMSE measure. By performing 1000 simulations and averaging, we obtain a single NRMSE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the NRMSE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the NRMSE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and NRMSE measure. LSA has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively

Figures

	Imputation : LLS4, error metric : RAE					**Imputation : KNN1, error metric : RAE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	2	3	2	1		10	10	10	9	
Site 2	2	2	2	1		10	10	10	10	
Site 3	2	2	1	2		10	10	10	9	
Site 4	2	3	2	1		10	10	10	10	
Site 5	3	2	2	2		10	10	10	9	
Site 6	2	2	2	2		10	10	10	9	
Avg	2.17	2.33	1.83	1.5	1.96	10	10	10	9.33	9.83

Table 8: Imputation test Ranking based on RAE error metric with 5% deletion - After 5% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the RAE measure. By performing 1000 simulations and averaging, we obtain a single RAE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the RAE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the RAE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and RAE measure. With RAE, LSA has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

	Imputation : ROW, error metric : RAEL2					**Imputation : KNN1, error metric : RAEL2**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	3		10	10	10	9	
Site 3	1	1	3	4		10	10	10	9	
Site 4	1	3	1	1		10	10	10	10	
Site 5	1	1	1	4		10	10	10	9	
Site 6	1	1	1	3		10	10	10	9	
Avg	1	1.33	1.33	2.67	1.58	10	10	10	9.17	9.79

Table 9: Imputation test Ranking based on RAEL2 error metric with 5% deletion - After 5% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the RAEL2 measure. By performing 1000 simulations and averaging, we obtain a single RAEL2 value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the RAEL2 values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the RAEL2. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and RAEL2 measure. ROW has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively

	Imputation : LSA, error metric : RMSE					**Imputation : KNN1, error metric : RMSE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	1		10	10	10	9	
Site 3	1	1	1	1		10	10	10	9	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	1		10	10	10	9	
Site 6	1	1	1	1		10	10	10	9	
Avg	1	1	1	1	1	10	10	10	9.17	9.79

Table 10: Imputation test Ranking based on RMSE error metric with 10% deletion - After 10% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the RMSE measure. By performing 1000 simulations and averaging, we obtain a single RMSE value for every imputation method. This process is repeated fo r all the 6 sites and 4 samples. Based on the RMSE values, the imputation methods are ranked for each site/sample combi nation from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank val ues across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the RMSE. The numbers in the last row represent the average ra nk across the 6 sites and the overall average for the given imputation method and RMSE measure. With RMSE, LSA has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

	Imputation : ROW, error metric : LRMSE					**Imputation : KNN1, error metric : LRMSE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	10	
Site 2	1	1	1	5		10	10	10	10	
Site 3	1	1	2	3		10	10	10	10	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	3		10	10	10	10	
Site 6	1	1	1	4		10	10	10	10	
Avg	1	1	1.17	2.83	1.50	10	10	10	10	10

Table 11: Imputation test Ranking based on LRMSE error metric with 10% deletion - After 10% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the LRMSE measure. By performing 1000 simulations and averaging, we obtain a single LRMSE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the LRMSE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the LRMSE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and LRMSE measure. ROW has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

	Imputation : LSA, error metric : NRMSE					**Imputation : KNN1, error metric : NRMSE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	1		10	10	10	9	
Site 3	1	1	1	1		10	10	10	9	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	1		10	10	10	9	
Site 6	1	1	1	1		10	10	10	9	
Avg	1	1	1	1	1	9.83	10	10	9.17	9.79

Table 12: Imputation test Ranking based on NRMSE error metric with 10% deletion - After 10% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the NRMSE measure. By performing 1000 simulations and averaging, we obtain a single NRMSE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the NRMSE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the NRMSE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and NRMSE measure. LSA has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

	Imputation : LLS4, error metric : RAE					**Imputation : KNN1, error metric : RAE**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	3	3	2	1		10	10	10	10	
Site 2	2	2	2	2		10	10	10	10	
Site 3	2	2	1	2		10	10	10	10	
Site 4	2	3	2	1		10	10	10	10	
Site 5	2	2	2	1		10	10	10	10	
Site 6	2	2	2	2		10	10	10	10	
Avg	2.17	2.33	1.83	1.5	1.96	10	10	10	10	10

Table 13: Imputation test Ranking based on RAE error metric with 10% deletion - After 10% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the RAE measure. By performing 1000 simulations and averaging, we obtain a single RAE value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the RAE values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the RAE. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and RAE measure. With RAE, LLS with k=4 has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

	Imputation : ROW, error metric : RAEL2					**Imputation : KNN1, error metric : RAEL2**				
	Sample A	Sample B	Sample C	Sample D		Sample A	Sample B	Sample C	Sample D	
Site 1	1	1	1	1		10	10	10	9	
Site 2	1	1	1	4		10	10	10	9	
Site 3	1	1	1	4		10	10	10	9	
Site 4	1	1	1	1		10	10	10	10	
Site 5	1	1	1	4		10	10	10	9	
Site 6	1	1	1	3		10	10	10	9	
Avg	1	1	1	2.83	1.46	10	10	10	9.17	9.79

Table 14: Imputation test Ranking based on RAEL2 error metric with 10% deletion - After 10% deletion, the missing value is imputed using 10 different imputation methods. Imputation methods are compares using the RAEL2 measure. By performing 1000 simulations and averaging, we obtain a single RAEL2 value for every imputation method. This process is repeated for all the 6 sites and 4 samples. Based on the RAEL2 values, the imputation methods are ranked for each site/sample combination from lowest to highest, 1 being lowest and 10 being highest. For each imputation method, we average the rank values across the 6 test sites and then average across the 4 samples. This table depicts the overall lowest ranked and the overall highest ranked imputation method as measured by the RAEL2. The numbers in the last row represent the average rank across the 6 sites and the overall average for the given imputation method and RAEL2 measure. ROW has the best performance and KNN with $k = 1$ has the worst performance in terms of ranking, respectively.

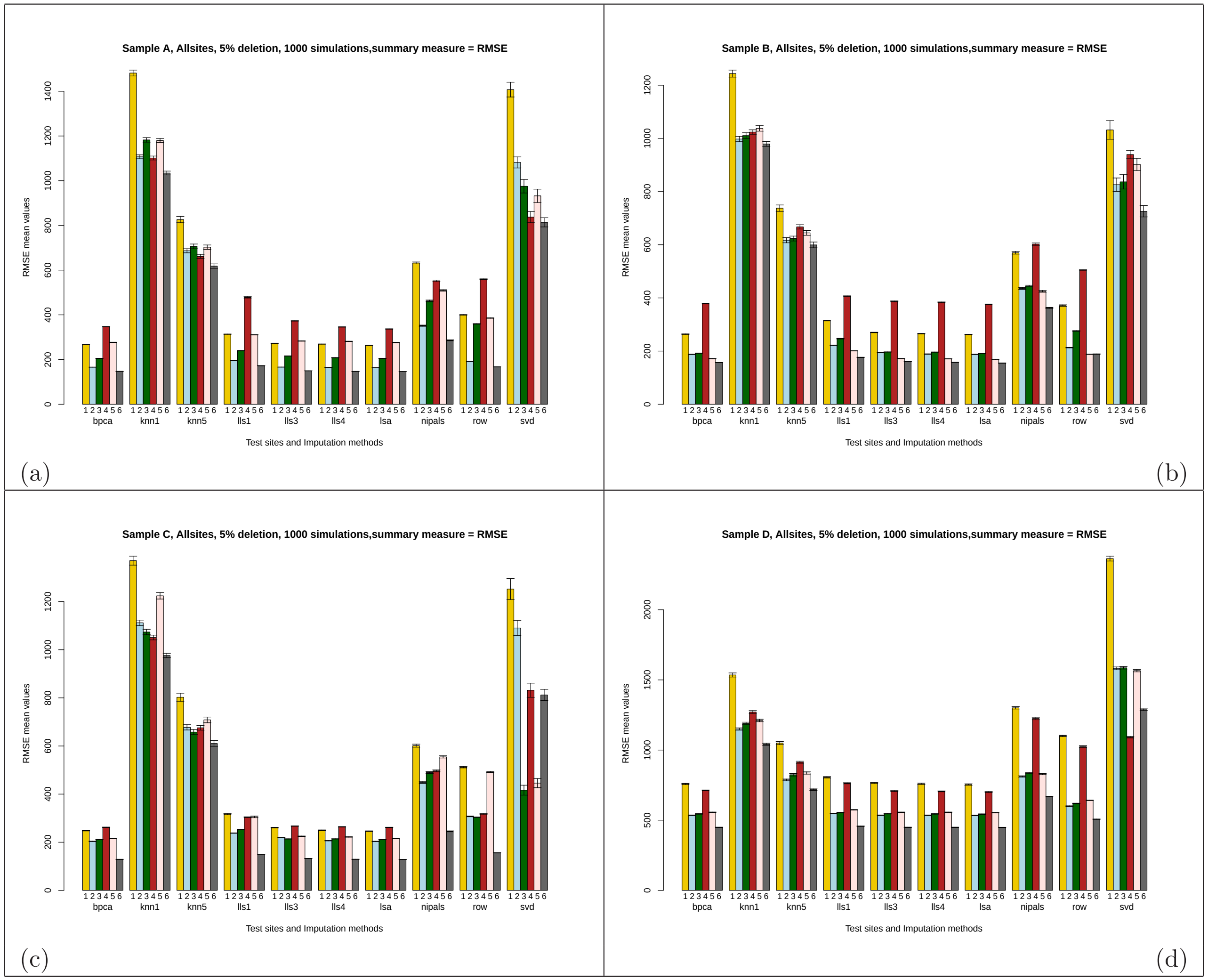

Figure 2: RMSE values seen as barplot with error bars representing the variance : RMSE values are represented o n the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with $k = 1, 5$, lls with $k = 1, 3, 4$, lsa, nipals, row, svd). RMSE values for (a) sample A , (b) sample B ,(c) sample C , and (d) sample D for th e imputation tests across sites for 5% missing values averaged over 1000 simulations.Figure shows the performance of the ten imputation tests using the RMSE metric with 5% deletion of values.1000 simulations were performed, where each simulation generated a dataset containing 5% missing values by randomly removing values from the complete matrix.Mis sing values were imputed using the ten imputation tests. The results are compared using the root mean square error (RM SE). Among all the tests, LSA has the best performance as it has the lowest RMSE value for all four samples across all six sites. KNN with k=1 has highest RMSE value and has the worst performance for samples A, B and C for all sites,sam ple D for site 4 only. SVD has the highest RMSE value for sample D for sites 1,2,3,5 and 6.

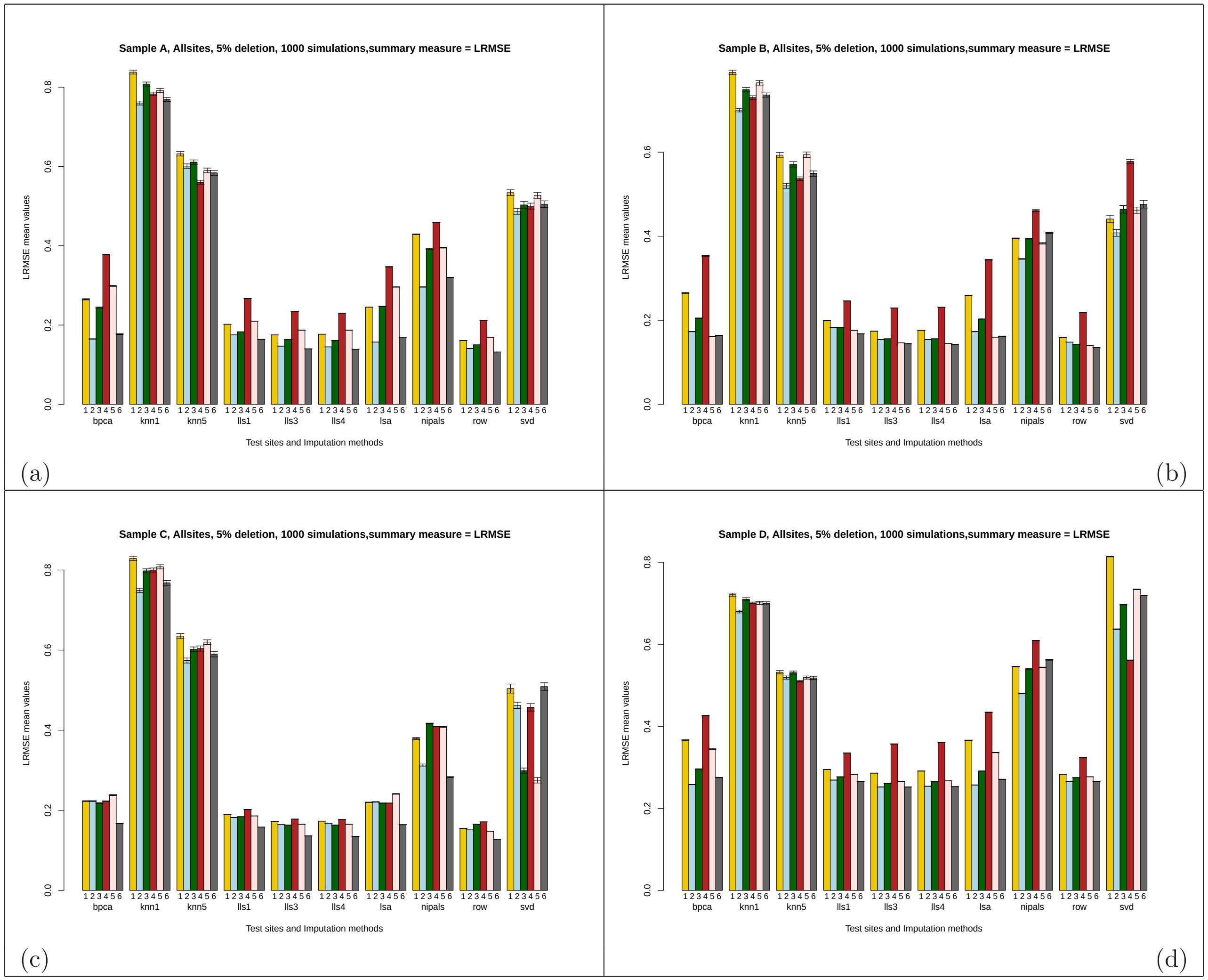

Figure 3: **LRMSE values seen as barplot with error bars representing the variance :** LRMSE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with $k = 1$, 5, lls with $k = 1$, 3, 4, lsa, nipals, row, svd). LRMSE values for (a) sample A , (b) sample B ,(c) sample C , and (d) sample D for the imputation tests across sites for 5% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the LRMSE metric with 5% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 5% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the log root mean square error (LRMSE). ROW has the best performance as it has the lowest value for samples A and B across all six sites. For sample C, row has the best performance in sites 1,2,4,5 and 6 whereas LLS $k = 3$, 4 has the best performance in site 3. For sample D, row has the best performance in sites 1, and 4 whereas LLS $k = 3$ has the best performance in sites 2,3,5, and 6. KNN with $k = 1$ has the highest value and worst performance for samples A, B and C across all six sites. For sample D, SVD has the worst performance for sites 1,5 and 6 but KNN with $k = 1$ has the worst performance in sites 2,3, and 4.

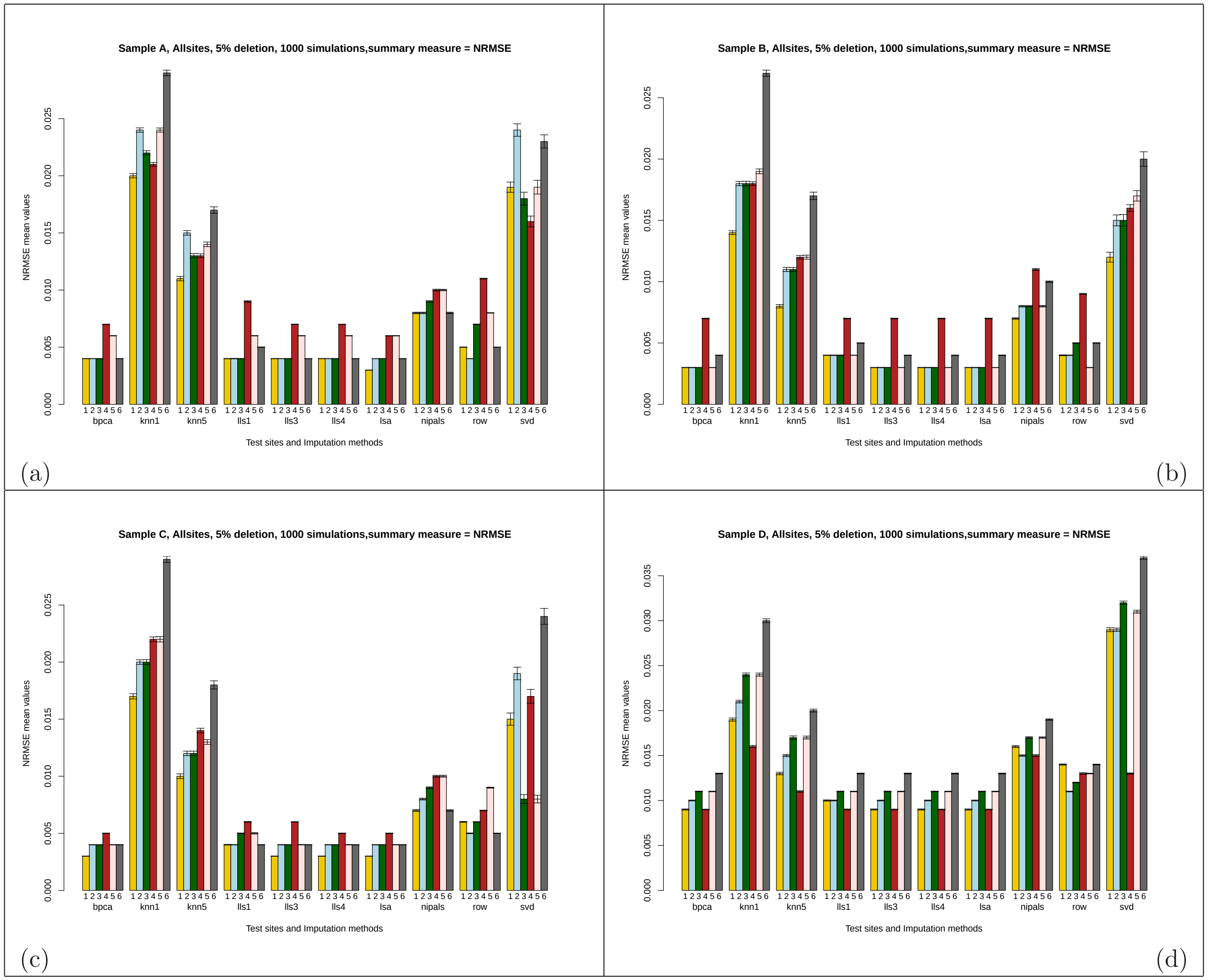

Figure 4: **NRMSE values seen as barplot with error bars representing the variance :** NRMSE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with $k = 1$, 5, lls with $k = 1$, 3, 4, lsa, nipals, row, svd). NRMSE values for (a) sample A , (b) sample B ,(c) sample C , and (d) sample D for the imputation tests across sites for 5% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the NRMSE metric with 5% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 5% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the normalized root mean square error (NRMSE). LSA has the best performance as it has the lowest NRMSE value for sample A across all six sites. LSA, BPCA, and LLS with $k =$ 3, 4 all have the lowest value and best performance for samples B, C and D across all six sites. KNN with $k = 1$ has the highest value and worst performance for samples A, B and C across all six sites. For sample D, SVD has the worst performance for sites 1,2,3,5 and 6 but KNN with $k =$ 1 has the worst performance in site 4.

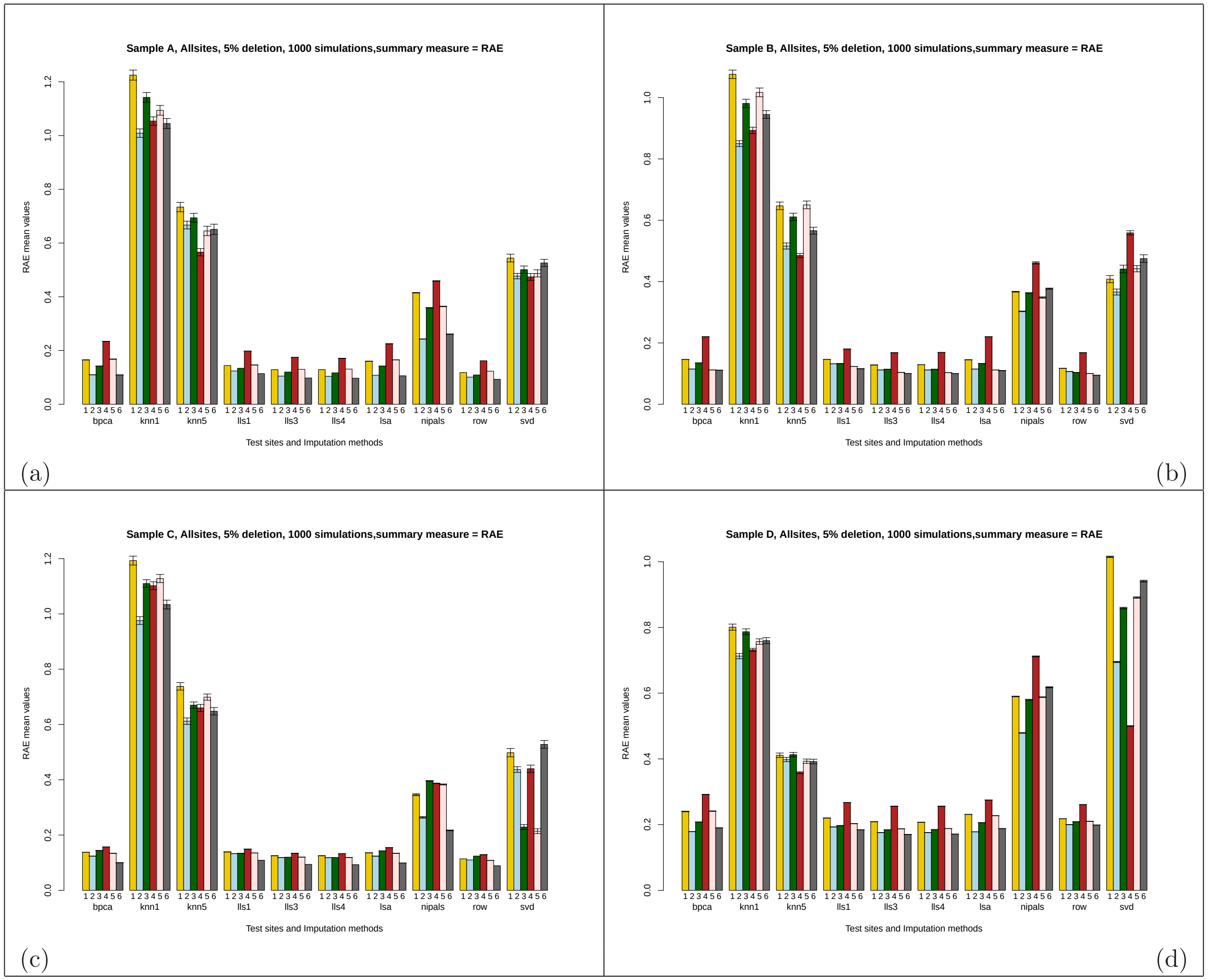

Figure 5: **RAE values seen as barplot with error bars representing the variance :** RAE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with k = 1, 5, lls with k = 1, 3, 4, lsa, nipals, row, svd). RAE values for (a) sample A , (b) sample B , (c) sample C , and (d) sample D, for the imputation tests across sites for 5% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the RAE metric with 5% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 5% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the RAE error. Among all the tests, ROW has the best performance as it has the lowest RAE value for samples A,B and C across all six sites whereas lls3 has the lowest value for sample D across all sites. KNN with $k = 1$ has highest RAE value and has the worst performance for samples A, B and C for all sites,sample D for sites 2 and 4 . SVD has the highest RAE value for sample D for sites 1,3,5 and 6.

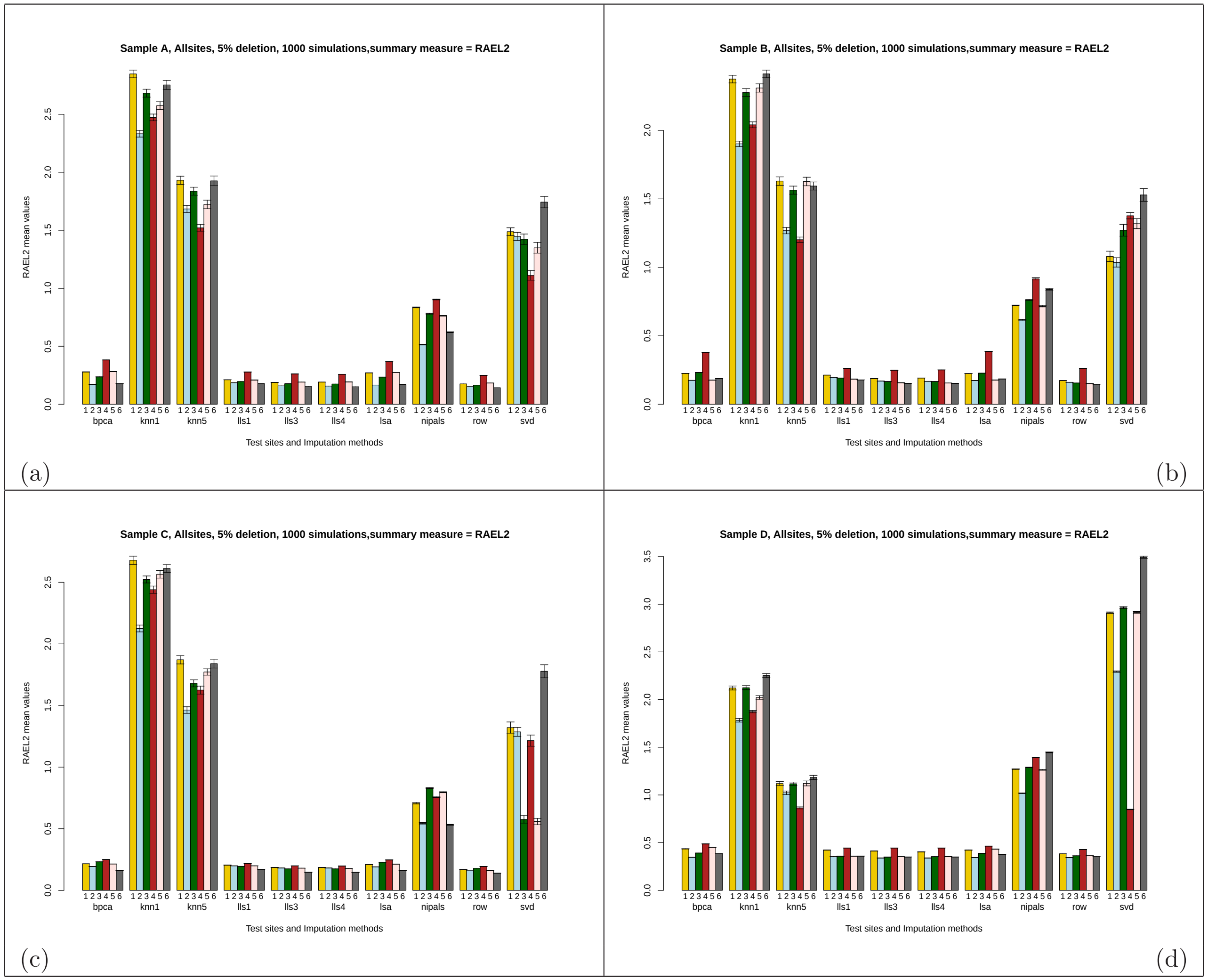

Figure 6: **RAEL2 values seen as barplot with error bars representing the variance :** RAEL2 values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with $k = 1$, 5, lls with $k = 1$, 3, 4, lsa, nipals, row, svd). RAEL2 values for (a) sample A , (b) sample B , (c) sample C , and (d) sample D, for the imputation tests across sites for 5% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the RAEL2 metric with 5% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 5% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the RAEL2 error. ROW has the best performance as it has the lowest RAE value for samples A,B and C across all six sites. For sample D, ROW has the lowest value for sites 1 and 4 whereas LLS with $k = 3$ have the best performance in sites 2,3,5, and 6. KNN with $k = 1$ has highest RAE value and has the worst performance for samples A, B and C for all sites, and for sample D in site 1. SVD has the highest RAEL2 value for sample D for sites 1,2,3,5 and 6.

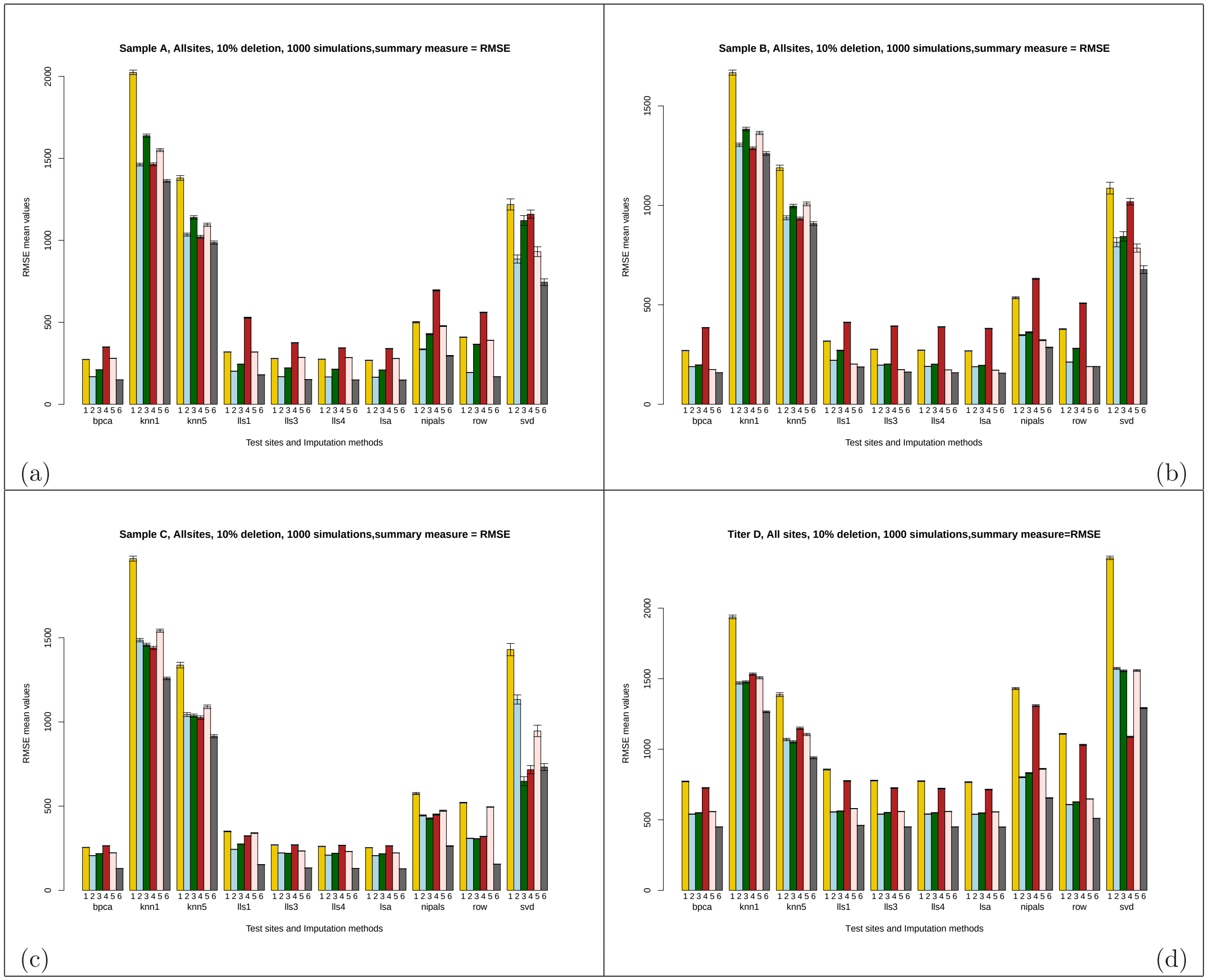

Figure 7: **RMSE values seen as barplot with error bars representing the variance :** RMSE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4 ,5 and 6) and 10 imputation tests (bpca, knn with k = 1, 5, lls with k = 1, 3, 4, lsa, nipals, row, svd). RMSE values for (a) sample A, (b) sample B,(c) sample C, and (d) sample D for the imputation tests across sites for 10% missing values averaged over 1000 simulations. The figure shows the performance of the ten imputation tests using the RMSE metric with 10% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 10% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the root mean square error (RMSE). Among all the tests, LSA has the best performance as it has the lowest RMSE value for all four samples across all six sites. KNN with k=1 has highest RMSE value and has the worst performance for samples A, B and C for all sites,sample D for site 4 only. SVD has the highest RMSE value for sample D for sites 1,2,3,5 and 6.

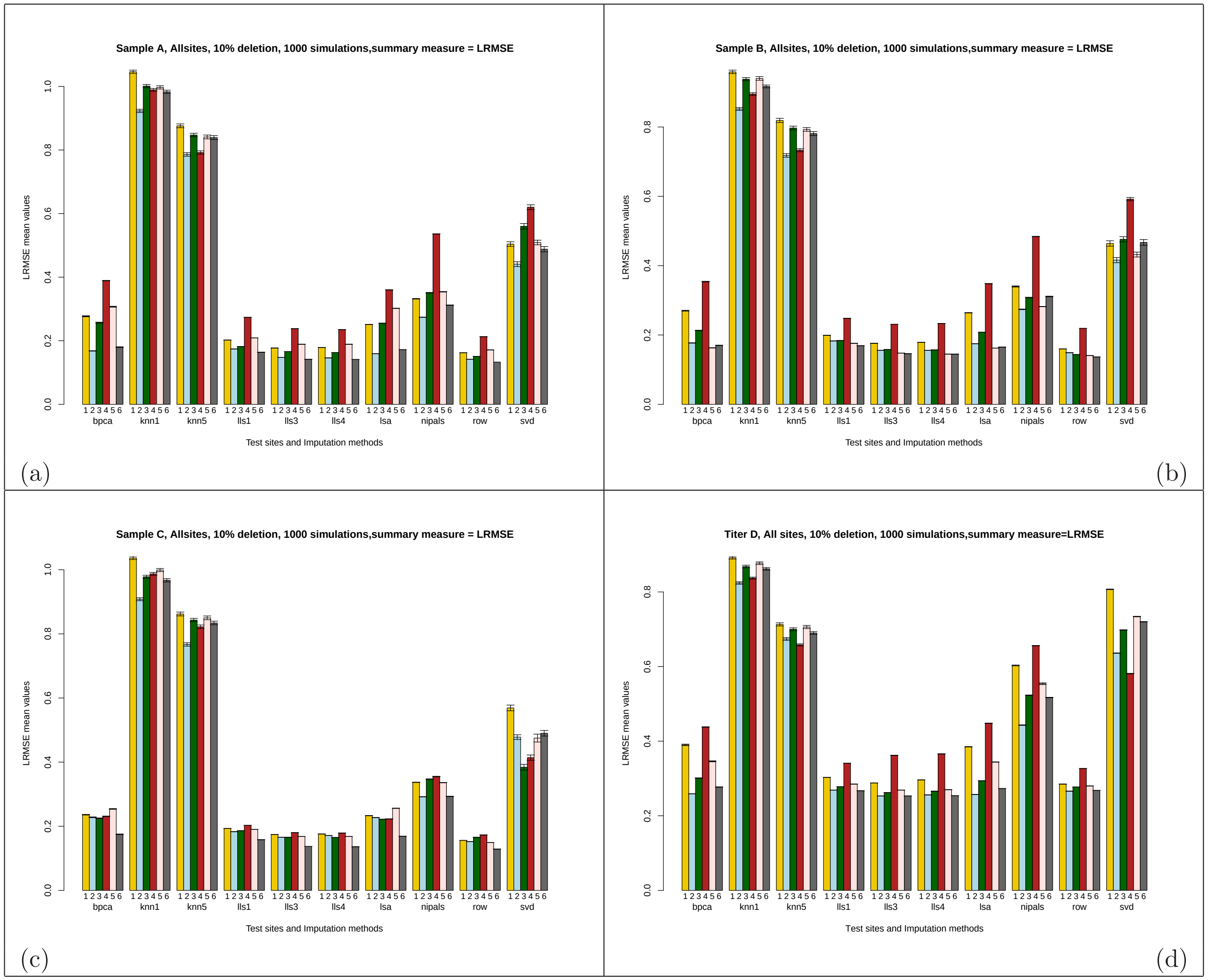

Figure 8: **LRMSE values seen as barplot with error bars representing the variance :** LRMSE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with k = 1, 5, lls with k = 1, 3, 4, lsa, nipals, row, svd). LRMSE values for (a) sample A, (b) sample B, (c) sample C, and (d) sample D for the imputation tests across sites for 10% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the LRMSE metric with 10% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 10% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the log root mean square error (LRMSE). ROW has the best performance as it has the lowest value for samples A and B across all six sites. For sample C, row has the best performance in sites 1,2,4,5 and 6 whereas LLS k = 4 has the best performance in site 3. For sample D, row has the best performance in sites 1, and 4 whereas LLS k = 3 has the best performance in sites 2,3,5, and 6. KNN with k = 1 has the highest value and worst performance for samples A, B, C and D across all six sites.

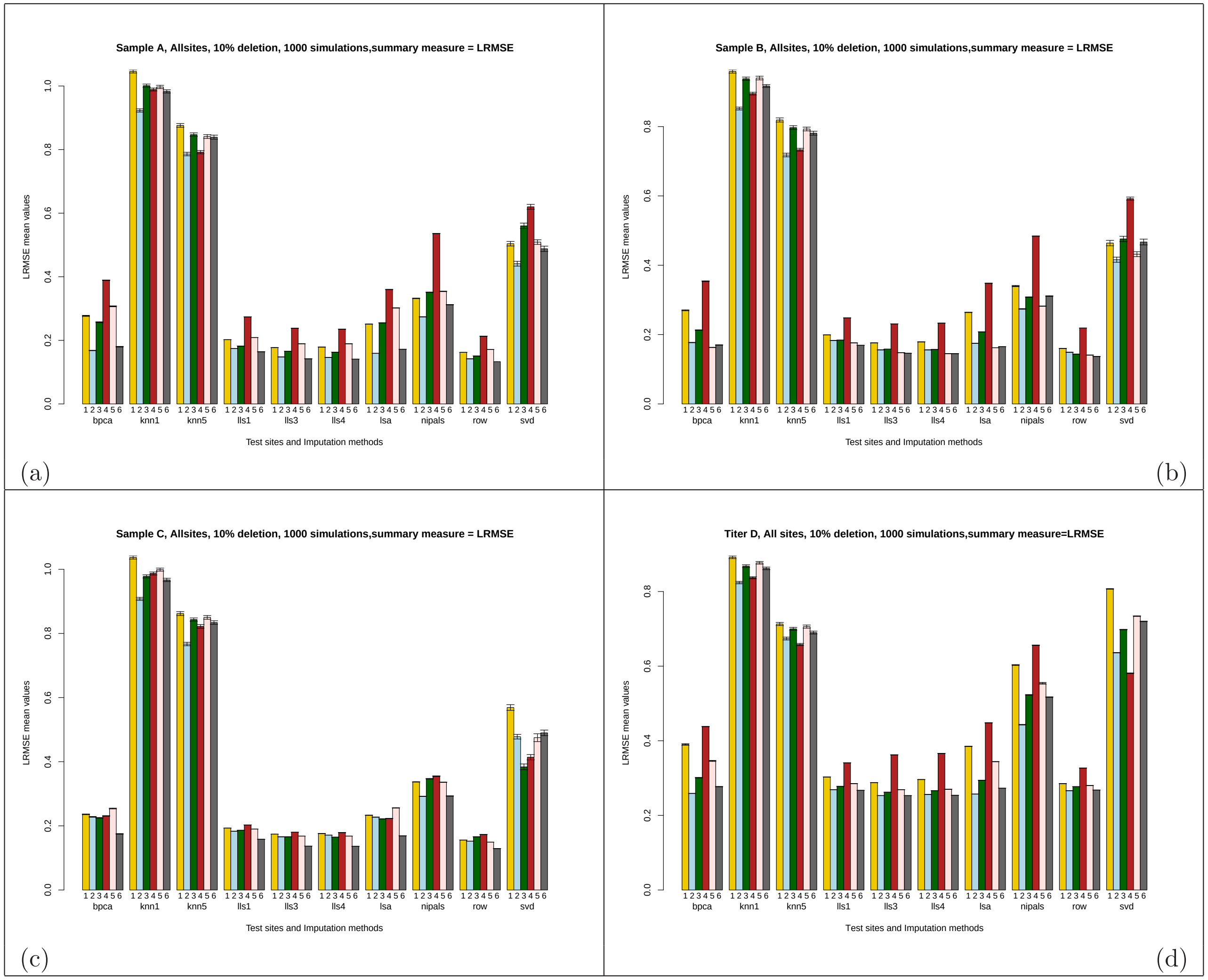

Figure 9: **NRMSE values seen as barplot with error bars representing the variance :** NRMSE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with k = 1, 5, lls with k = 1, 3, 4, lsa, nipals, row, svd). NRMSE values for (a) sample A, (b) sample B, (c) sample C, and (d) sample D for the imputation tests across sites for 10% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the NRMSE metric with 10% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 10% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the normalized root mean square error (NRMSE). LSA has the best performance as it has the lowest NRMSE value for sample A (in site 4), sample B (in site 3), sample C (in site 4), and sample D (in site 1). LSA, BPCA, and LLS with k = 4, all have the lowest value and best performance for sample A (sites 1,2,3,5, and 6), sample B (sites 1,2,4,5, and 6), sample C (sites 1,2,3,5, and 6), and sample D (sites 2,3,4,5, and 6) respectively. KNN with k = 1 has the highest value and worst performance for samples A, B and C across all six sites. For sample D, SVD has the worst performance for sites 1,2,3,5 and 6 but KNN with k = 1 has the worst performance in site 4.

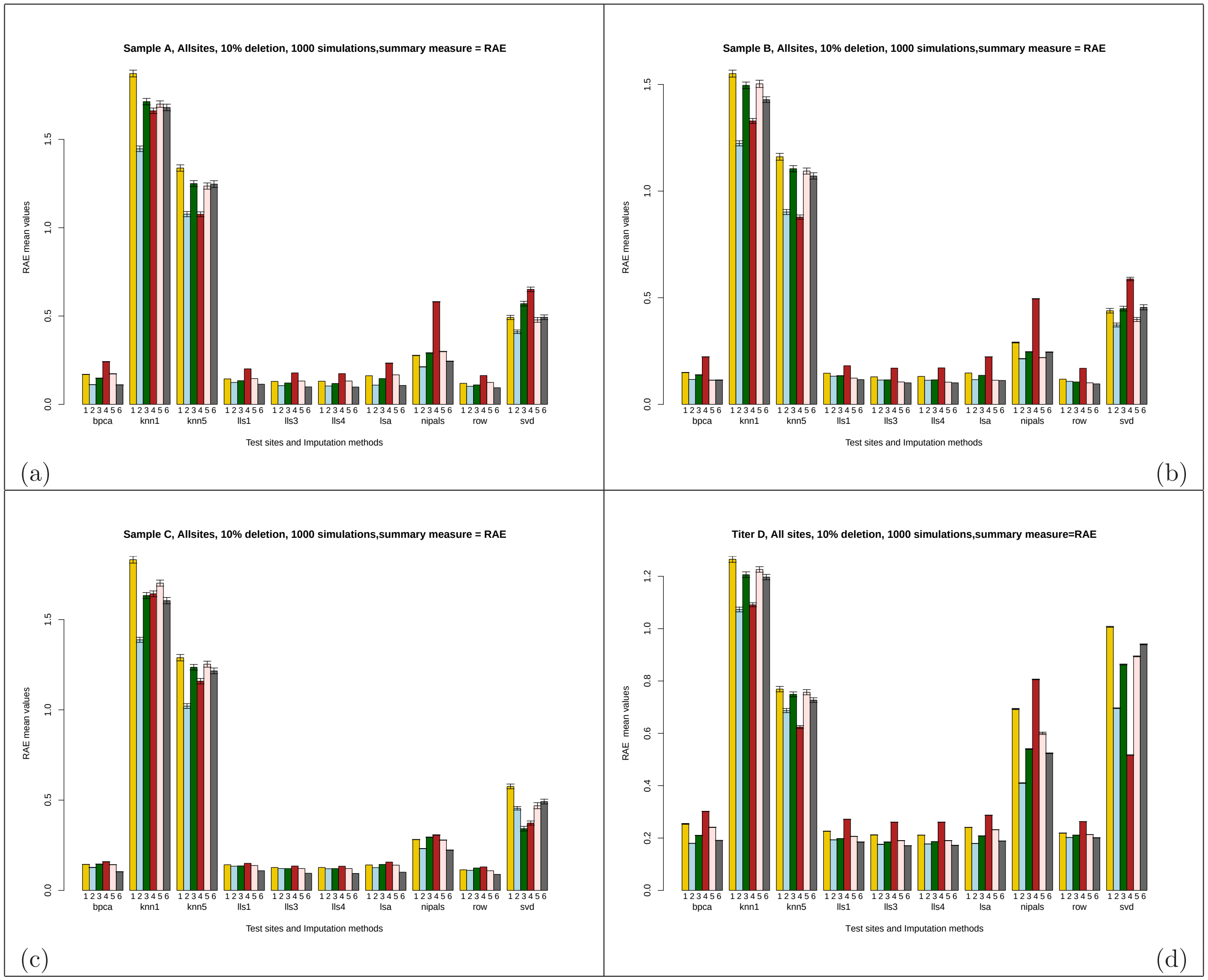

Figure 10: **RAE values seen as barplot with error bars representing the variance :** RAE values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with k = 1, 5, lls with k = 1, 3, 4, lsa, nipals, row, svd). RAE values for (a) sample A, (b) sample B, (c) sample C, and (d) sample D, for the imputation tests across sites for 10% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the RAE metric with 10% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 10% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the RAE error. Among all the tests, ROW has the best performance as it has the lowest RAE value for samples A and B across all six sites. For sample C, ROW has the best performance as it has the lowest RAE value for sites 1,2,4,5, and 6, whereas LLS with k = 3, 4 has the lowest value for site 3. For sample D, LLS with k = 3 has the best performance for sites 2,3,4,5, and 6, whereas LLS with k = 4 has the lowest value for site 1. KNN with k = 1 has highest RAE value and has the worst performance for samples A, B, C and D for all sites.

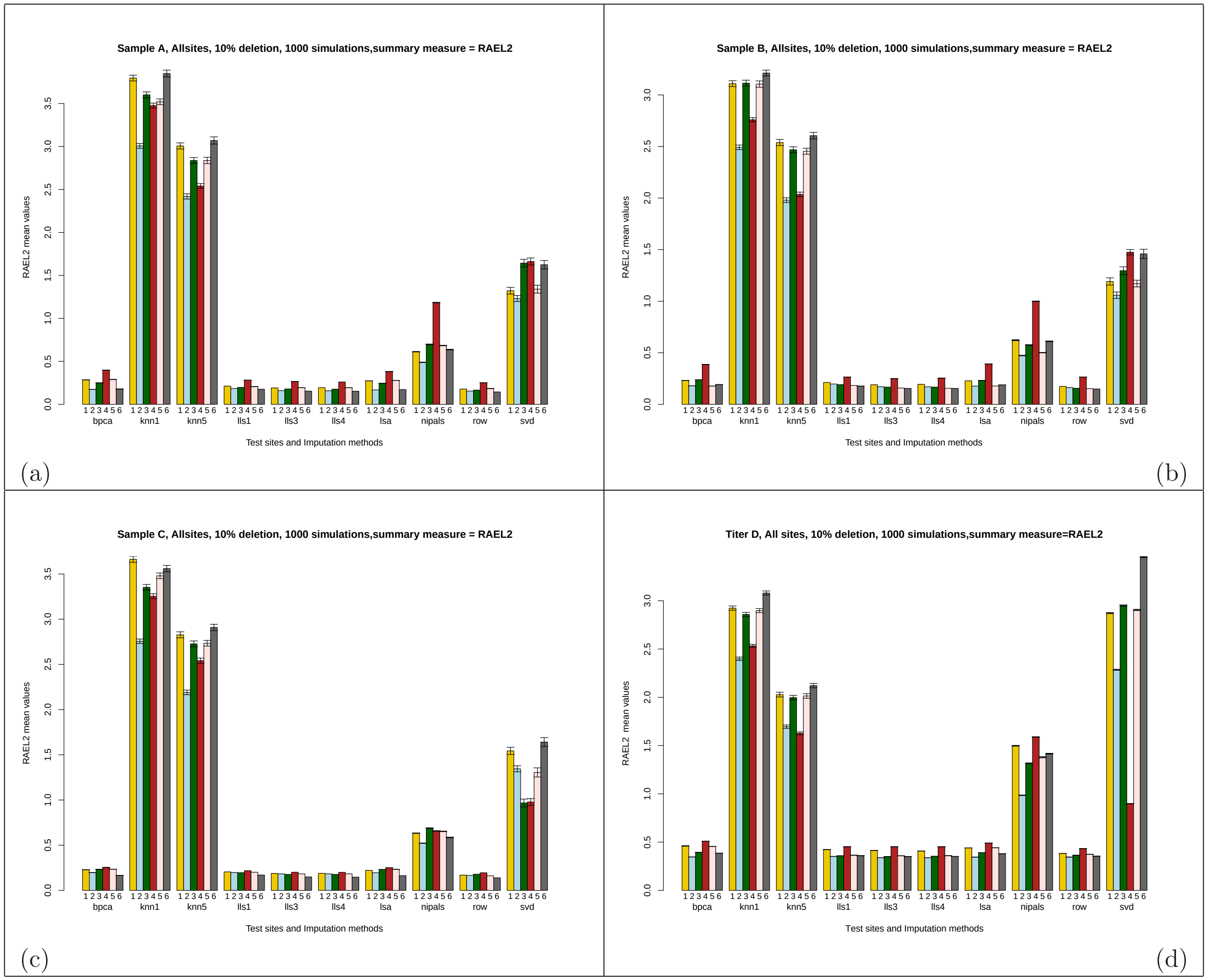

Figure 11: **RAEL2 values seen as barplot with error bars representing the variance :** RAEL2 values are represented on the y axis. The x axis has the 6 sites (1,2,3,4,5 and 6) and 10 imputation tests (bpca, knn with k = 1, 5, lls with k = 1, 3, 4, lsa, nipals, row, svd). RAEL2 values for (a) sample A, (b) sample B, (c) sample C, and (d) sample D, for the imputation tests across sites for 10% missing values averaged over 1000 simulations. This figure shows the performance of the ten imputation tests using the RAEL2 metric with 10% deletion of values. 1000 simulations were performed, where each simulation generated a dataset containing 10% missing values by randomly removing values from the complete matrix. Missing values were imputed using the ten imputation tests. The results are compared using the RAEL2 error. ROW has the best performance as it has the lowest RAEL2 for samples A and B across all six sites. For sample B, ROW has the best performance in sites 1,2,3,5, and 6, whereas LLS with k = 3 has the best performance in site 4. For sample D, ROW has the best performance in sites 1, and 4 whereas LLS with k = 3 has the best performance in sites 2,3,5, and 6. KNN with k = 1 has highest value and has the worst performance for samples A, B and C for all sites, and for sample D in sites 1 and 4 . SVD has the highest RAEL2 value for sample D for sites 2,3,5 and 6.