Chemometrics and Intelligent Laboratory Systems 210 (2021) 104248

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Comparison of zero replacement strategies for compositional data with large numbers of zeros

Sugnet Lubbe [a], Peter Filzmoser [b,*], Matthias Templ [c]

[a] Department of Statistics and Actuarial Science, Stellenbosch University, South Africa
[b] Institute of Statistics and Mathematical Methods in Economics, Vienna University of Technology, Austria
[c] School of Engineering, Zurich University of Applied Sciences, Switzerland

ARTICLE INFO

Keywords:
Imputation
Compositional data analysis
ZeroSum regression
Microbiome data

ABSTRACT

Modern applications in chemometrics and bioinformatics result in compositional data sets with a high proportion of zeros. An example are microbiome data, where zeros refer to measurements below the detection limit of one count. When building statistical models, it is important that zeros are replaced by sensible values. Different replacement techniques from compositional data analysis are considered and compared by a simulation study and examples. The comparison also includes a recently proposed method (Templ, 2020) [1] based on deep learning. Detailed insights into the appropriateness of the methods for a problem at hand are provided, and differences in the outcomes of statistical results are discussed.

1. Introduction

On the face of it, compositional data appears to be multivariate positive observations. However, care has to be taken not to simply apply multivariate data analysis methods as per usual. The key difference is that with compositional data it is the relative information of the observations that is relevant. Consider a D-part composition $\boldsymbol{x} = [x_1, \ldots, x_D]'$ with strictly positive parts $x_1, \ldots, x_D$. The same relative information is contained in x_i/x_j and ax_i/ax_j for any non-zero scalar value a. The composition can be re-expressed as proportions, $\boldsymbol{x}^* = a\boldsymbol{x}$ by setting $a = 1/\sum x_i$. The composition $\boldsymbol{x}^*$ belongs to the $(D-1)$-standard simplex defined by

$$\left\{\boldsymbol{x}^* = [x_1^* \quad \ldots \quad x_D^*]' \,\middle|\, x_i^* > 0, \sum_{i=1}^{D} x_i^* = 1\right\}.$$

In order to apply standard multivariate data analysis methods, the compositions need to be transformed from the simplex to the Euclidean space. Two isometric transformations to achieve this goal are the centred log ratios (clr) and isometric log ratios (ilr), discussed in detail in Filzmoser, Hron and Templ [1] or Filzmoser and Hron [2].

Both the clr and ilr transformations make use of ratios of the compositional components, as well as the log of a ratio. Since a zero value in the denominator of a ratio, or the logarithm of a zero value is not valid, zeros have been excluded from the above definition. However, zeros can occur in compositional data, and there are different kinds of zeros: essential or structural zeros are values that are truly zero while rounded and count zeros occur due to imprecision, values below the detection limit or insufficient sample size. These values are not truly zero and it makes sense to replace these zeros with suitable small values in order to proceed with the usual compositional data analysis techniques (for details, see e.g. Ref. [1]. In this paper, different zero replacement alternatives are evaluated, and the interest is specifically to evaluate their performance where the majority of the observations are zero.

In applications such as chemometrics and microbiome analyses, large proportions of zero values often occur. For example, the output in high throughput sequencing data counting genes as in 16S rRNA gene sequencing, transcriptomics and metagenomics or single-nucleotide variant abundances is in the form of a table of counts. Often these tables are sparse with up to around 90% of the cells containing zero values [3]. Gloor et al. [4] discuss at length why these tables should be considered as compositional data where the total read count is irrelevant and the counts represent a random sample of the relative abundance of the molecules in the underlying ecosystem. It is therefore important to be aware of the performance of zero replacement methods in very sparse compositional count tables.

This paper is organized as follows. Section 2 briefly reviews existing methods for zero replacement in compositional data. Here, the focus is on

* Corresponding author.
E-mail address: p.filzmoser@tuwien.ac.at (P. Filzmoser).

https://doi.org/10.1016/j.chemolab.2021.104248
Received 8 October 2020; Received in revised form 8 January 2021; Accepted 8 January 2021
Available online 12 January 2021

continuous data, as for this type of data there are more methods available. We also refer to a recently developed method based on a deep learning approach. A comprehensive simulation study in Section 3 compares the different methods for their ability to replace zeros by positive values. The different methods are also compared in real microbiome data examples in Section 4. The findings are discussed in the final Section 5.

2. Methodology

The simplest option in a sample to replace zeros in a compositional part is to only consider the non-zero values of this particular part (univariate replacement). As compositional data are multivariate by their nature, important information of the observations is ignored in this approach. However, particularly with a high proportion of zeros, multivariate replacement methods suffer from the fact that more and more parts in a compositional observation have zero entries, and depending on the method, this may lead to severe computational problems.

Table 1 lists several of the existing zero replacement methods, together with a brief description and availability in the software environment R [5]. The univariate replacement methods considered here are *const* and *unif*. Both methods replace the zero value by a value between zero and the detection limit (DL). While *const* uses always the same value for a variable, *unif* draws a value from a random uniform number, and thus the multivariate data distribution might be less distorted. Note that such simple replacement strategies are also quite common in microbiome studies, where an arbitrary constant, e.g. 1, is added to each entry of the covariate matrix before performing a log-transformation (see, e.g., Ref. [6]. Whereas the choice of the positive constant is not based on a rigorous statistical theory, the method *const* selects the constant such that the bias in the covariance structure in minimized (Martín-Fernández et al. [7]. Furthermore, only zeros are replaced in our approaches, and the non-zeros are unchanged in order to avoid any bias. In any case, such univariate replacement methods are in conflict with the fact that compositional data are by definition multivariate data, which should be taken into account by a replacement method.

The R package zCompositions [8] provides several methods for the multivariate imputation of zeros and non-detects in compositional data. The methods build on an appropriate coordinate representation of the compositional data in the usual Euclidean geometry. The replacement is done in an iterative manner, and for that purpose the EM algorithm, Markov Chain Monte Carlo (MCMC) or multiple imputation are utilized. The algorithm multLN performs a multiplicative lognormal replacement, which means that for the imputation a log-normal distribution is fit and the parameters are iteratively estimated. The algorithm multRepl refers to a multiplicative simple replacement [7]. It basically uses the method *const*, but in addition the remaining non-zero parts are multiplicatively adjusted. This does not require a coordinate representation of the composition. The package zCompositions contains several other methods (the most prominent are listed in the bottom part of Table 1). However, some of the algorithms are only intended for count data, and others have computational difficulties if the proportion of zeros becomes very high. For this reason, these methods are not further considered.

Further methods for the replacement of zeros in (continuous) compositional data are implemented in the R package robCompositions [15] and described in detail in Filzmoser et al. [1]. The algorithm *BDLs* (BDL is the abbreviation of "below detection limit") is an iterative model-based procedure which performs regressions to replace the zeros. Different options for regression are implemented, such as ordinary multiple linear regression, robust regression, and partial least-squares (PLS) regression. The regression is performed in a coordinate representation of the compositional data. A similar procedure is based on k-nearest-neighbour imputation (algorithm *impKNNa*), but for a large number of zeros there are too few neighbours with non-zeros available,

Table 1
Zero imputation methods considered for the simulation study.

Method	R Function	R Pakcage	Description/comment
const	0.65*DL		The simplest method is replacing all zeros with a constant value smaller than the detection limit. Martín-Fernández et al. (2003) found that 65% of the detection limit minimizes the distortion in the covariance structure.
unif	runif(0.1*DL,DL)		Using a constant value in the majority of cells leads to underestimation of the compositional variability. Although uniform values between 0 and the detection limit (*DL*) is often used, setting the first parameter at 0.1*DL* prevents imputed values from being too close to zero.
multLN	multLN()	zCompositions	Model-based multiplicative lognormal imputation [8].
multRepl	multRepl()	zCompositions	Non-parametric multiplicative simple imputation [7].
BDLs	imputeBDLs()	robCompositions	EM-based parametric replacement using partial least squares (PLS) with a special choice of balances [9]. The PLS option is used since the classical and robust regression cannot be performed on too sparse compositional data tables.
deepImp	deepImp()	deepImp	Imputation with deep learning methods, particularly using deep artificial neural networks in an EM-based approach [10].
Methods not implEmented			
lrDA	lrDA()	zCompositions	Simulation-based Data Augmentation (DA) algorithm to impute left-censored values [11]. An initial covariance matrix needs to be specified which is challenging to estimate for very sparse compositional data tables.
lrEM	lrEM()	zCompositions	Model-based ordinary and robust Expectation-Maximisation algorithms to impute left-censored data [12, 13]. An initial covariance matrix needs to be specified which is challenging to estimate for very sparse compositional data tables.
multKM	multKM()	zCompositions	Non-parametric multiplicative Kaplan-Meier smoothing spline imputation of left-censored values [8]. In very sparse compositional data tables the number of knots is larger than non-zero observations for the spline fitting.
impKNNa	impKNNa()	robCompositions	K-nearest neighbour methods for imputation [14]. With very sparse compositional data tables too few non-zero neighbours are present.

which makes the algorithm not applicable in this context. The algorithm *deepImp* has been proposed only recently [10]. It employs artificial neural networks (ANNs) to replace zeros, but represents the compositions first in coordinates. The algorithm works like a typical EM algorithm (as described e.g. in Ref. [16] for the imputation of missing values, i.e. after an initial imputation, the missing values are sequentially updated for each variable. This means that one ANN per variable and run is fitted to impute initial missing values in this variable. Before every fit, the data are presented in isometric log-ratio coordinates, and after the fit, the inverse transformation is used to return the data in the original scale. The method has several tuning parameters, such as the number of layers, the number of neurons in each layer, or the number of training epochs for the network. The default setting is to use 10 layers, 1000 neurons in the first hidden layer, 900 in the second, …, 100 in the last hidden layer. The activation functions used are reLu [17] and adam [18] as adaptive moment estimation stochastic gradient approach that uses an adaptive

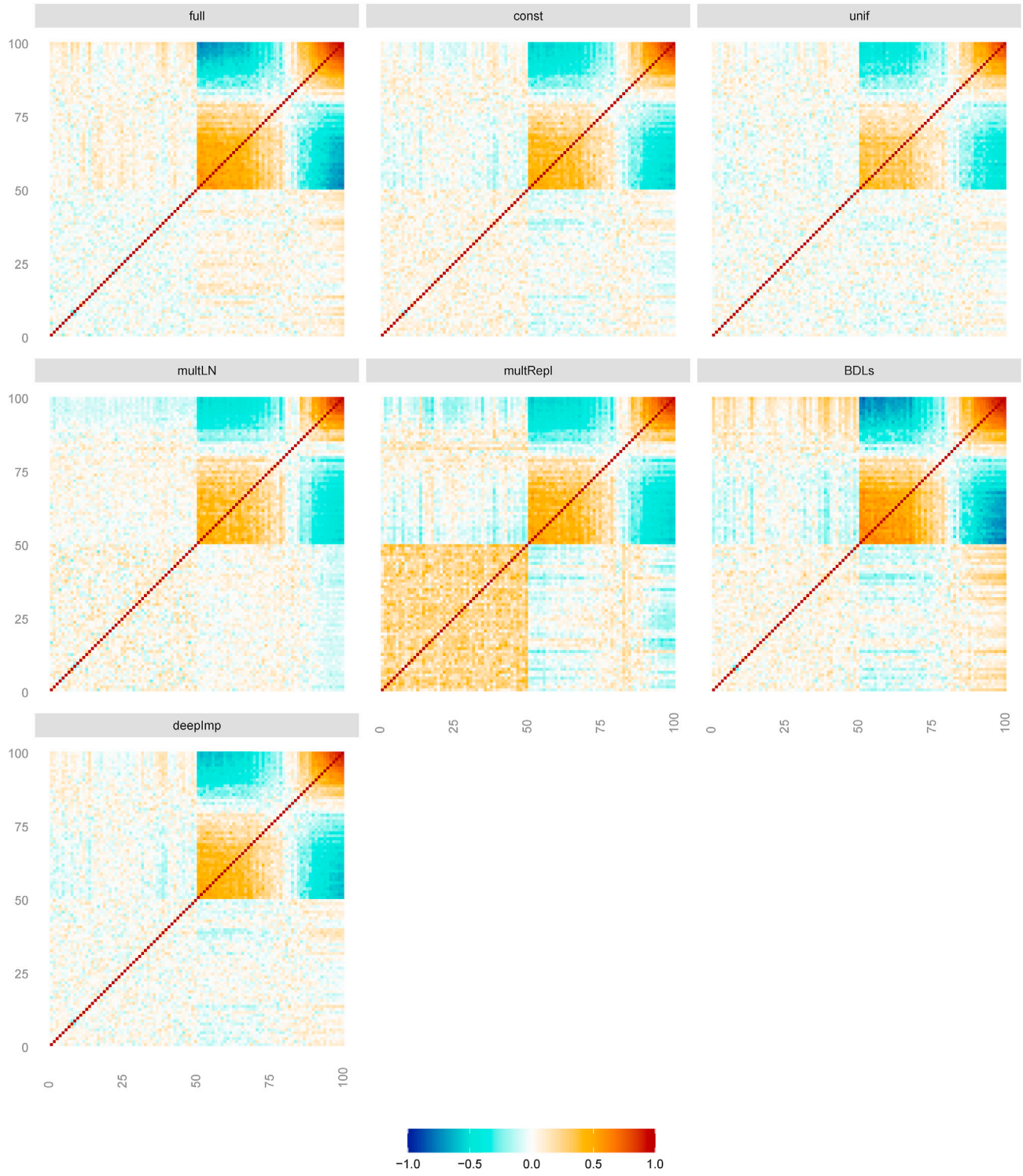

Fig. 1. Visual representation of pivot correlation matrices to compare data structure between an original simulated compositional data matrix $\boldsymbol{X}$ and different zero imputed matrices when 50% of the values are zero.

learning rate, a mean squared loss function, and mean absolute error evaluation metric. More details on this procedure and parameter settings can be found in Templ [10]. Naturally, this leads to a time-consuming procedure, and so far there is no experience of how this procedure performs with compositional data containing a high number of zeros.

3. Simulation study

In order to investigate different replacement techniques, a simulation study was performed in R [5] to compare the imputation methods listed in the upper part of Table 1. Imputation functions from the R packages zCompositions [8] and robCompositions [15] were included in the study. Further details on the replacement methods can be found in the references in the R packages, and in Filzmoser, Hron, Templ [2]. Since the simulated data is continuous, only methods which are designed for the zero replacement of continuous compositional data are included. The idea of the simulation design is to generate two blocks of data: one block of variables which are uncorrelated, and another block with correlated variables. The data have to be generated in the Euclidean geometry, and then transformed to the simplex appropriately in order to keep the block structure. Values are replaced by zeros, and the performance of the imputation methods can be evaluated according to the distortion of the correlation and distance structure.

3.1. Simulation design

More specifically, in each simulation run a data set of 200 observations on 100 parts was generated. The data generation followed the methodology described in Hron et al. [19]. Two blocks of parts are formed in the Euclidean space: $\boldsymbol{Z}_1 : 200 \times 49$ of uniform values on a 49-dimensional sphere and $\boldsymbol{Z}_2 : 200 \times 49$ from a multivariate normal distribution. The R package geozoo [20] is used to generate $\boldsymbol{Z}_1$ to obtain observations from a distribution with all pairwise correlations theoretically zero. The multivariate normal distribution used to generate $\boldsymbol{Z}_2$ has parameters $\boldsymbol{\mu} = \boldsymbol{0}$ and $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ with $\sigma_{ii} = 1$ and $\sigma_{ij} = 0.6$. A column $z_1 : 200 \times 1 = \boldsymbol{0}$ is added to link the two blocks forming a matrix $\boldsymbol{Z} : 200 \times 99 = [z_1 \quad \boldsymbol{Z}_1 \quad \boldsymbol{Z}_2]$. In the ilr transformation a set of contrasts needs to be specified. Based on so-called sequential binary partitioning contrasts, a back transformation is performed to obtain a matrix $\boldsymbol{X} : 200 \times 100$ in the simplex space. The resulting compositional data set $\boldsymbol{X}$ now consists of a block of 50 parts which are uncorrelated and a second block of highly correlated parts. See the "full" panel of Fig. 1 for a visual representation of the correlation structure in one realization of the matrix $\boldsymbol{X}$.

This panel "full" in Fig. 1 reveals that the first block (lower left) indeed does not contain any correlation structure, but the second block (upper right) shows a range of strong positive and negative correlations. One could modify the values $\sigma_{ij} = 0.6$ for this second block to obtain stronger (or weaker) correlations, but the main focus in the subsequent analysis is on the contrast between the uncorrelated first and the correlated second block. One could also think of simulation settings with more variables than observations, which, however, would lead to difficulties for some of the zero replacement methods later on (see also example section).

The simulated compositional matrix $\boldsymbol{X}$ does not contain any zero values. In the evaluation of imputation methods, the proportion of zero values in the matrix is specified, say p. In order to create a matrix $\boldsymbol{X}_0$, based on $\boldsymbol{X}$, with $p100\%$ zero values, the p-th sample quantile of each column of $\boldsymbol{X}$ is set as the detection limit (DL). After setting $\boldsymbol{X}_0 \leftarrow \boldsymbol{X}$ all observations in $\boldsymbol{X}_0$ smaller than DL are set to zero.

To evaluate how well the zero imputation procedures reproduce the original structure in the data set $\boldsymbol{X}$, the correlation matrices are compared. The pivot correlation matrix $R(\boldsymbol{X})$ as defined in Kynčlová, Hron, and Filzmoser [21], which collects correlations between orthonormal logratio coordinates capturing all relative information about the original compositional parts within a given composition (a special for of ilr coordinates, so called symmetric pivot coordinates), is computed. In addition to evaluating the relationships between the variables with correlations, relationships between the samples are compared by computing the pairwise distances between the samples, $D(\boldsymbol{X})$, using the Aitchison distance [22].

In Fig. 1, a visual representation of the $D \times D$ (100×100) pivot correlation matrices and in Fig. 2, $n \times n$ (200×200) distance matrices, are given as computed, based on one realization of $\boldsymbol{X}$ and matrices $\boldsymbol{X}_{imputed}$ obtained by imputing a proportion $p = 0.5$ of zero values in $\boldsymbol{X}_0$.

Perusal of Figs. 1 and 2 shows that all but the *multRepl* method reproduces the correlation structure and intersample distances fairly well; the algorithm *multLN* produces a distorted distance matrix, but the correlation structure is still quite well preserved. A closer look at the figures shows that the correlation structure of the 50 highly correlated variables is reflected very well by the methods *BDLs* and *deepImp*, while the correlation structure of the 50 less correlated variables is best reproduced by the method *unif*, while other algorithms introduce a small artificial correlation structure. In summary, the methods *const* and *unif* do not perform as well as the methods *BDLs* and *deepImp*. The methods *multLN* and *multRepl* reflect the correlation structure worst. This is also the case in the comparison of intersample distances (Fig. 2), where *BDLs* performs best (followed by *deepImp*), while *multLN* and *multRepl* change the distances significantly.

3.2. Evaluation measures

Two quantitative measures are defined to compare an imputed matrix with the original compositional data matrix $\boldsymbol{X}$:

$$c_{imp} = \frac{1}{D^2} R(\boldsymbol{X}) - R\left(\boldsymbol{X}_{imputed}\right)_F^2 = \frac{1}{D^2} \sum_{i=1}^{D} \sum_{j=1}^{D} \left(r_{ij} - r_{ij}^*\right)^2$$

where r_{ij} is the ij-th element of $R(\boldsymbol{X})$ and r_{ij}^* is the ij-th element of the matrix $R(\boldsymbol{X}_{imputed})$;

$$d_{imp} = \frac{1}{n^2} D(\boldsymbol{X}) - D\left(\boldsymbol{X}_{imputed}\right)_F^2 = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(d_{ij} - d_{ij}^*\right)^2$$

where d_{ij} is the ij-th element of $D(\boldsymbol{X})$ and d_{ij}^* is the ij-th element of the matrix. $D(\boldsymbol{X}_{imputed})$.

The simulation study was performed at $p = 0.01, 0.05, 0.10, 0.15, \ldots, 0.80, 0.85, 0.90, 0.99$ proportion of zeros, with 5 replicates at each proportion. From the simulation it was found that beyond 50% zero values the *multLN* and *multRepl* methods do not provide useful imputations – either the function returns an error result or imputed values outside the permissible range, i.e. negative imputations.

3.3. Simulation results

In Fig. 3 the mean c_{imp} and d_{imp} values for the 5 simulation replicates are shown. Since these are essentially measuring the deviation from the original data structure, smaller values are more desirable.

The *multLN* and *multRepl* methods do not perform well for more than 20% zeros and these methods break down at around 55% zero values. For correlations, *BDLs* performs best until about 35 or 40% zeros. For larger proportions of zeros, *deepImp* performs best, but it cannot handle greater-equal 95% zeros well. In terms of distances *BDLs* performs best until about 70% zeros. The *deepImp* method has an unexpected behaviour below 25% zeros, otherwise it is quite competitive. The simplest imputation methods, *const* and *unif*, perform reasonably well. Although *const* outperforms *unif* in terms of retaining the correlation structure, the *unif* method is superior for retaining the intersample distances with more than 50% zeros.

Next to preserving correlations and distances, it is also desirable that a zero replacement algorithm is stable. Fig. 4 represents a new simulation with more replications in order to reveal the variability of the resulting

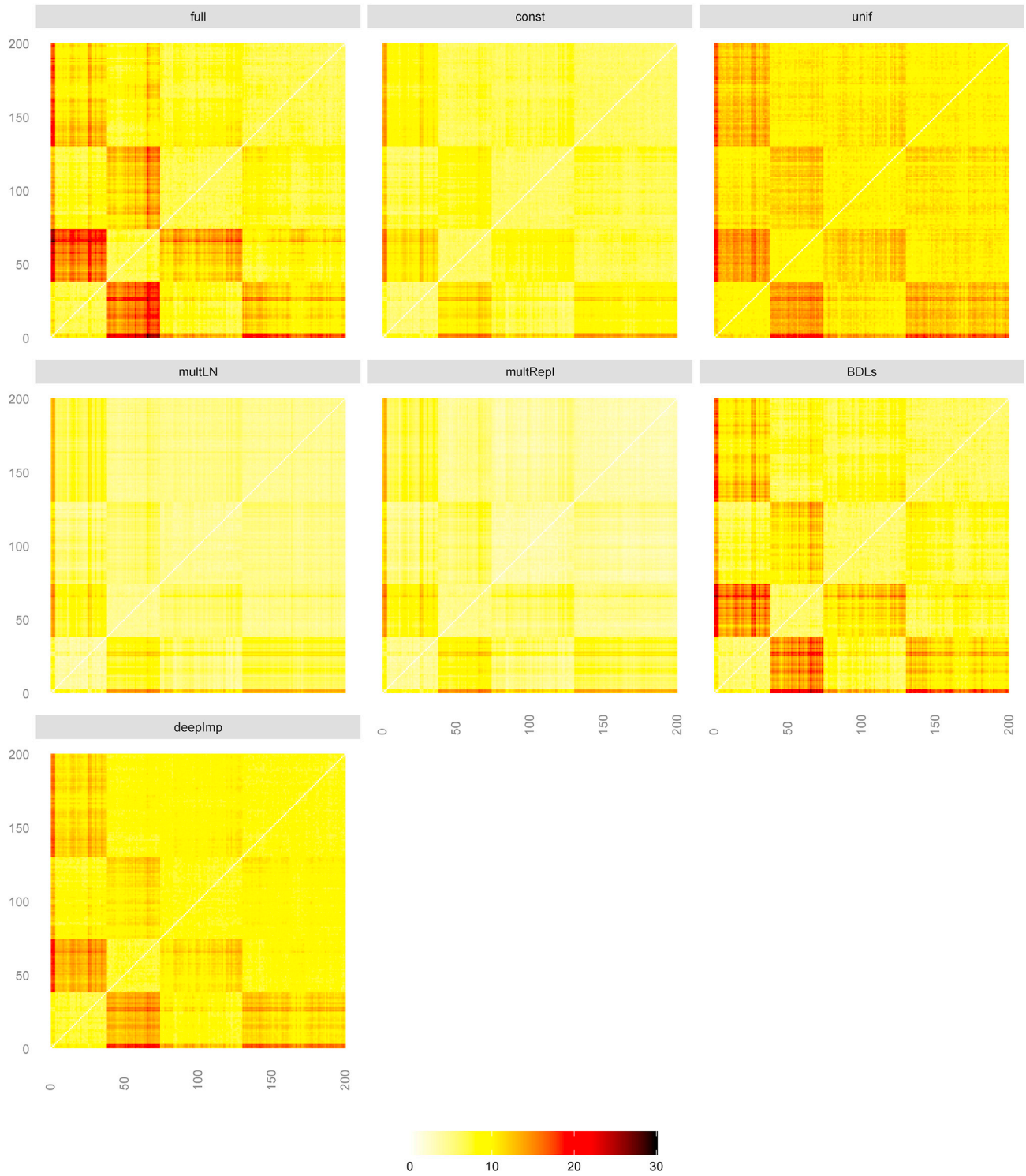

Fig. 2. Visual representation of distance matrices to compare data structure between an original simulated compositional data matrix X and different zero imputed matrices when 50% of the values are zero.

error measures. The different panels arranged in the rows refer to the proportions of zeros indicated on the right-hand side. The figure shows essentially the same picture as Fig. 3 in terms of the performance of the different methods. However, some algorithms often deliver quite different result, as seen for *BDLs*, which is an indication that the algorithm has convergence issues.

4. Examples

In this section the different zero replacement techniques are compared with real microbiome data examples. Microbiome data are count data, and if the count is not at least one, a zero is reported. Thus, the detection limit is one. We use the different replacement methods to impute values in the interval (0,1]. In order to refer to the same setting

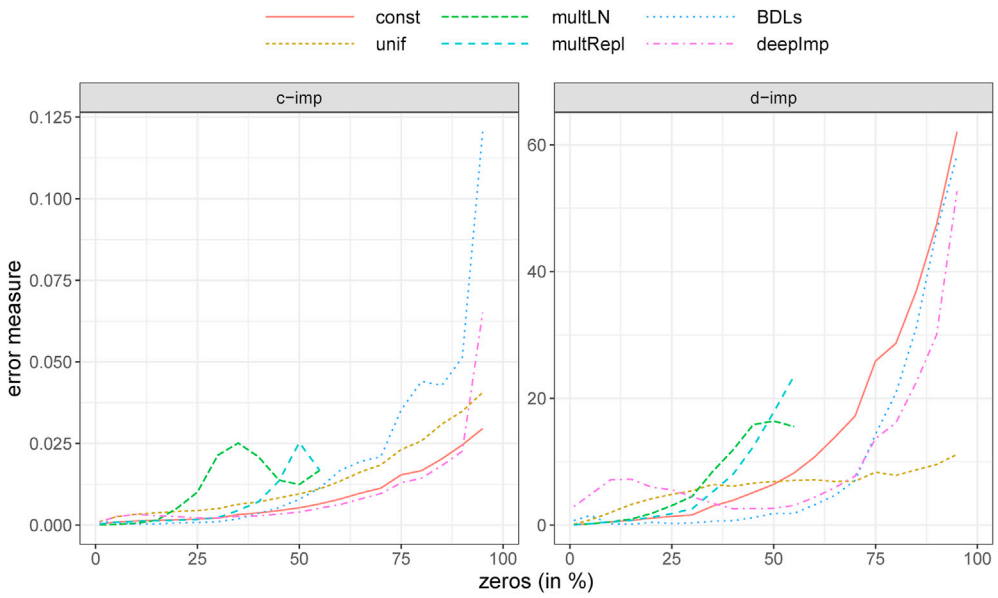

Fig. 3. The mean of 5 replicates of c_{imp} and d_{imp} values across a range of proportion of zero values.

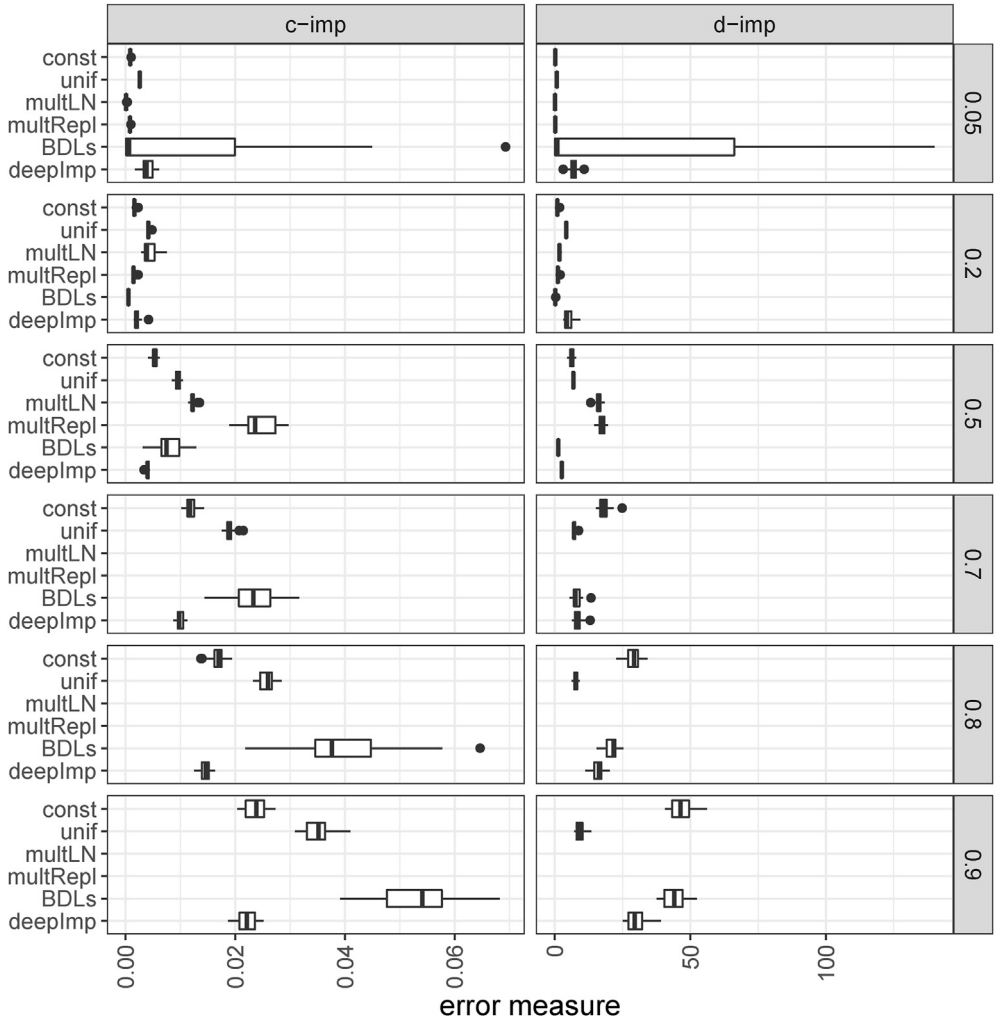

Fig. 4. The distribution of 25 replicates of c_{imp} and d_{imp} values across different proportions of zero values.

and to the same statistical task, the focus below is on regression problems, where a real response is regressed on the microbiome composition.

4.1. Example datasets

The following four data sets are considered, where the name (printed in bold) will refer to the corresponding results of the analyses:

Sulfate: Altenbuchinger et al. [23] studied the association between the microbiome composition of allogeneic stem cell transplants patients and urinary 3-indoxyl sulfate (3-IS) levels. They selected a small set of 160 operational taxonomic units (OTUs) that are jointly associated with the 3-IS levels. We reduced this set further to 119 variables by excluding OTUs with less than two non-zeros. The resulting composition has dimension 37×119, and contains 68% zero values. The response variable, the 3-IS levels, was log-transformed to better approximate normality.

BMI: In this study of Wu et al. [24], a microbiome dataset originated by high-throughput sequencing of 16S rRNA of fecal samples from 98 healthy individuals, resulting in 6674 OTUs. From these individuals, also the body mass index (BMI) is available (transformed to with the normal quantiles to follow a normal distribution), which is considered as a response in order to study the association with obesity based on the microbiome data as covariate composition. As the composition has many zeros, the OTUs were combined into bacteria species and genera, and we retained only those OTUs which are associated to at least 6 genera classes, and where the number of non-zeros was at least two. The resulting data matrix has dimension 98×78, and the fraction of zeros is 0.69.

Coffee: The same composition as for the BMI data set is used, but here we removed OTUs with prevalence less than 10%, as well as those which had zeros in more than 73 samples. The compositional data matrix finally has dimension 98×241, and the proportion of zeros is 0.49. As response variable we considered the caffeine intake, since coffee consumption has an impact on the gut microbiota [25]. The response has been transformed as in Xiao at al. (2018) to approximate normality.

Nugent: Vaginal bacterial communities of four ethnic groups of 388 North American women were sampled and pyrosequencing of barcoded 16S rRNA genes resulted in a microbiome data set [26], where only those OTUs were selected where the OTU taxa were present in at least 5% of the samples. This results in a composition of dimension 388×84, with 82% zeros. The response is a Nugent score with 11 categories, where higher score values relate to higher risk of bacterial vaginosis [26].

Various methods for regression on a compositional response are available. We select the method zeroSum proposed by Lin et al. [27], because it makes use of the compositional nature of the data by working with the so-called linear log-contrast model. Moreover, it is appropriate for high-dimensional regression because of an L1 penalty on the regression coefficients, similar to Lasso regression, which results in sparsity of the regression coefficient vector. Altenbuchinger et al. [23] further extended the estimation procedure with an elastic net penalty. The sparsity, controlled by a tuning parameter, produces many zeros in this vector and thus leads to a variable selection, where ideally the most relevant OTUs are selected for explaining the relationships to the response. zeroSum is implemented in R and available in the Github project https://github.com/rehbergT/zeroSum [23].

4.2. Zero replacement

Before starting with building a model based on zeroSum, the zeros are first replaced by the different methods. Fig. 5 compares in more detail the imputed values from the different algorithms by means of probability plots, referring to probabilities of a normal distribution. Every single imputed data point is shown here, with a specific symbol and color for each algorithm. Trivially, the methods *const* and *multRepl* result in

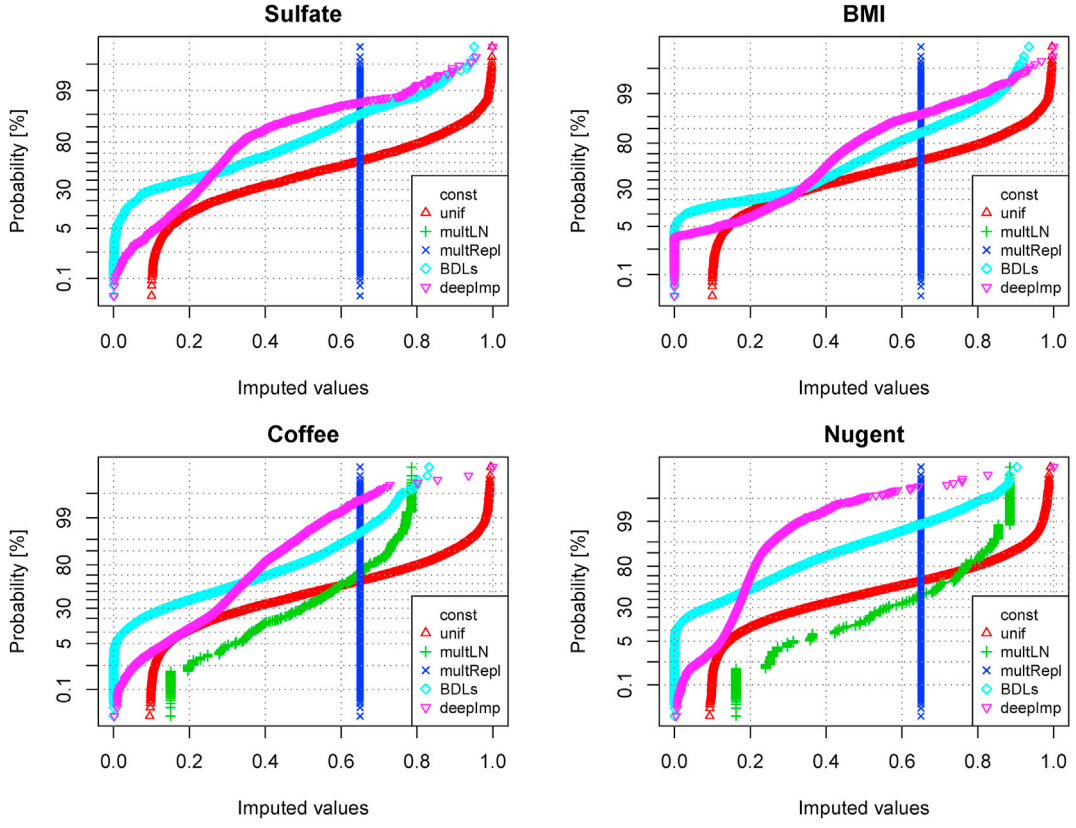

Fig. 5. Comparison of the imputed values from the different algorithms in a probability plot. The values for *const* and *multRepl* are on top of each other. *multLN* failed to give results for the data sets *Sulfate* and *BMI*.

vertically arranged points with values 0.65. Also as expected, the algorithm *unif* leads to a characteristic S shape in the range 0.05 and 1. The structure of the values from *multLN* looks somehow distorted, with several identical values at the extremes, which are still in a much smaller range than (0,1), and gaps in between. This algorithm did not yield results for the data sets *Sulfate* and *BMI*, and this happens if there are too many zeros in one variable. The algorithm *BDLs* leads to extremely left-skewed imputed values, and the maximum is still clearly away from 1. The values imputed by *deepImp* look much more symmetric, with a median clearly lower than 0.5. Note that all imputations are just "technical values", satisfying some underlying rule or model, while in reality one would have to have an integer (e.g. if the abundance would be higher). Nevertheless, the distribution of these values affects the log-ratios, which are the basis for a compositional data analysis approach. Intuitively, having continuously distributed imputed values in the whole range (0,1) is a desirable outcome of an imputation algorithm, and *deepImp* seems to satisfy this goal.

Table 2 provides insight into the computation time of the different methods for replacing the zeros. The reported time is in seconds, and computations were done on an Intel(R) Xeon(R) Gold 5120 CPU with 2.20 GHz, 16 GB RAM (under Ubuntu). The algorithms *BDLs* and *deepImp* require much more time than the other methods, but this also heavily depends on the parameter settings. For *BDLs* we used in the respective R function the parameters maxit = 50, eps = 0.1 (both for convergence), and R = 50 (number of bootstrap samples to determine the number of PLS components). For *deepImp* there are several more parameters for tuning, also related to convergence (iterations = 2, eps = 1), but also related to the specifics of the neural networks (patience.val = 40, epochs = 500, dropout = 0.1, 10 layers starting with 1000 neurons).Table 2

4.3. Regression modeling

Based on the imputed composition, we used the zeroSum procedure with the default settings, and by using only an l1 penalty, applied on the log-transformed compositional values. These are first total-sum normalized in order to refer to log-contrasts, as requested for this procedure. The tuning parameter selection is performed with 5-fold cross-validation, and we select that tuning parameter which gives the smallest error measure. This leads to a sparse model, and the cross-validated mean-square error (MSE-CV) for this model can be extracted. Since these values were quite instable, we repeated the whole tuning parameter selection procedure 50 times, resulting in 50 models. Fig. 6 shows all 50 MSE-CV values in boxplots, per imputation method. The results are quite comparable for the datasets *BMI* and *Coffee*, but there are more pronounced differences for the other two datasets. For the dataset *Sulfate*, which has about 3 times more variables than observations, *BDLs* leads to clearly higher variability. In such a setting, this imputation algorithm might bring in too complex multivariate data information, possibly also caused by a very varying number of PLS components, which leads to quite different models. For the dataset *Nugent*, the algorithm *deepImp* yields much smaller prediction errors in general compared to the other algorithms. This dataset has much more observations than the other datasets, which is a preferable setting for methods based on deep learning.

A further outcome of the different models is the vector of regression coefficients. Particular interest is in the non-zero values, as the associated variables are used to interpret the relationship of the microbiome data with the response. Here we do not compare with those OTUs considered relevant in the related literature (i.e. the potential ground truth), but we only compare the results from the different imputation algorithms. However, since zeroSum has been applied 50 times, we end up with 50 models per imputation algorithm and dataset, with corresponding sparsity of the regression coefficient vectors. We took a simple approach to aggregate this information: if a variable had a regression coefficient of zero in at least half (i.e. 25) of the models, the corresponding entry was set to zero; otherwise, the sign of the majority of the coefficients was reported. Thus, for each variable we obtain values from the set $\{-1, 0, 1\}$, which were color-coded as {blue, white, red} and presented as vertical bars in Fig. 7. The horizontal axes of the plots show the variable numbers of the four data sets, while the rows represent the replacement algorithms. Except for the *Nugent* data set, the models are highly sparse, with many regression coefficients being zero. Moreover, the methods seem to have high agreement on the non-zero coefficients, as well as on their sign. The algorithm *BDLs* gives a much more sparse solution for the *Coffee* dataset. Also for the *Nugent* data, *BDLs* involves several other variables compared to the other methods, and *unif* also leads to a somewhat different answer. Although *deepImp* had a clearly smaller prediction error for the *Nugent* data (see Fig. 6), it is surprising to see that the variables involved in the models are very similar to the algorithms *const* and *multLN*.

In practice it is common to verify if the variables associated with non-zero regression coefficients are related to the response. This can be done by plotting those variables against the response, or by computing correlations. Here, we computed correlations between the response and the variables which are reported as non-zero (per method) in Fig. 7. The correlations are summarized as absolute values in Fig. 8 by boxplots. Generally, the correlations for the datsets *BMI* and *Coffee* are much lower than for the other datasets, which indicates that there is a less clear relationship between the response and the composition. For these datasets one can see much lower medians for some algorithms – but this is also because there are only few non-zero coefficients (see Fig. 7). The algorithm *unif* has a much lower median for the *Sulfate* data, and when comparing with Fig. 7, there are several more variables involved, which seem to be not strongly connected to the response. Also for the *Nugent* data, *unif* yields the smallest median, and Fig. 7 shows the same picture that *unif* involves more and different variables compared to the other algorithms. For this datset, *deepImp* has the highest median, and also the smallest prediction error (Fig. 6), and thus it seems that this algorithm included valuable information in the zero-replaced values.

Table 2
Computation time (in seconds) for the different replacement methods for the four data sets. The second row provides the dimension of the microbiome data sets (no. of rows x no. of columns) and their percentage of zeros.

Time [s]	Sulfate	BMI	Coffee	Nugent
	39 × 119, 68%	98 × 78, 69%	98 × 241, 49%	388 × 84, 82%
const	0.000	0.007	0.001	0.000
unif	0.017	0.428	0.102	0.165
multLN	–	–	1.685	0.844
multRepl	0.003	0.013	0.015	0.079
BDLs	744	792	8362	2918
deepImp	1378	909	3894	1588

5. Discussion and conclusions

There are various reasons for the occurrence of zeros in compositional data, and these are known in the literature as rounded zeros, count zeros, and structural zeros (see, e.g., Ref. [1]. In either case, if standard compositional data analysis methods are considered, zeros cannot be processed because these methods use log-ratios as basis information. For rounded and count zeros it is common to replace the zeros by "small" values – depending on the type of data. However, if the proportion of zeros in the data gets very high, such as in microbiome data, it will be increasingly important which replacement methods are consulted.

We have compared several well-known algorithms for zero replacement, and also included a recently proposed method based on deep learning [10]. The comparison was based on simulated data and on real microbiome data. The focus in the simulation was purely on the quality of the imputed data in terms of how well the distances and correlations are preserved. In the examples it was not possible to get information about the ground truth of distances and correlations in the composition, and thus the focus was on the predictive performance of the model. We

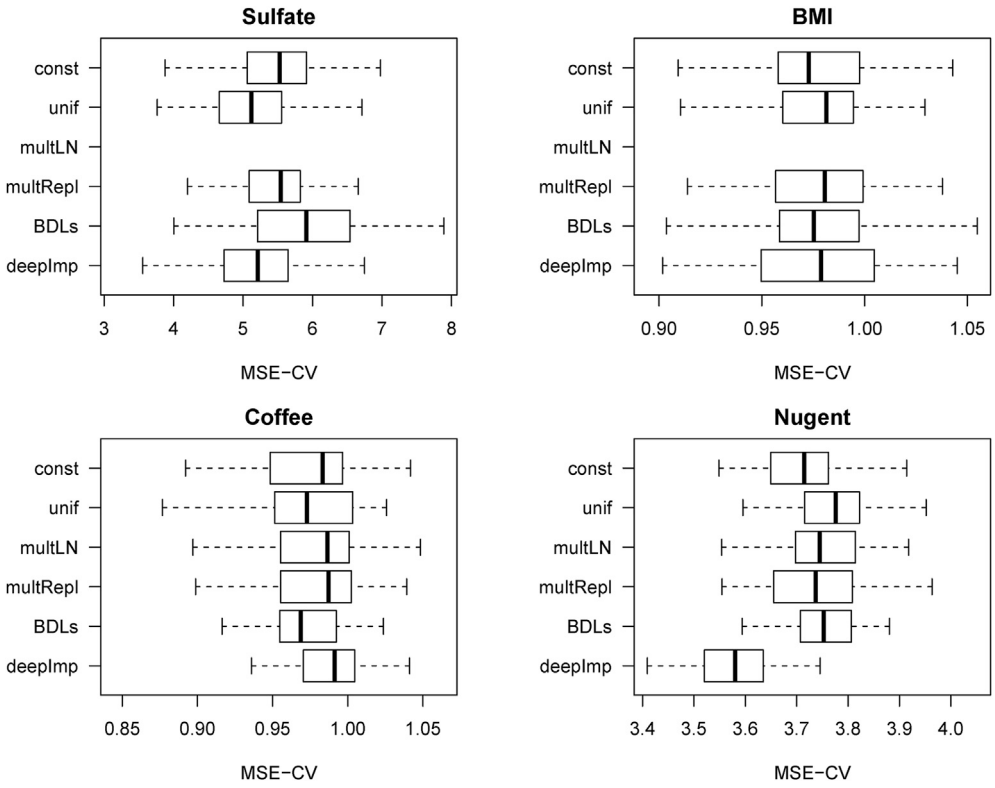

Fig. 6. Cross-validated mean-square errors (MSE-CV) for 50 replications of the zeroSum algorithm, based on different replacement methods for the four data sets. The 50 MSE-CV values are summarized in boxplots.

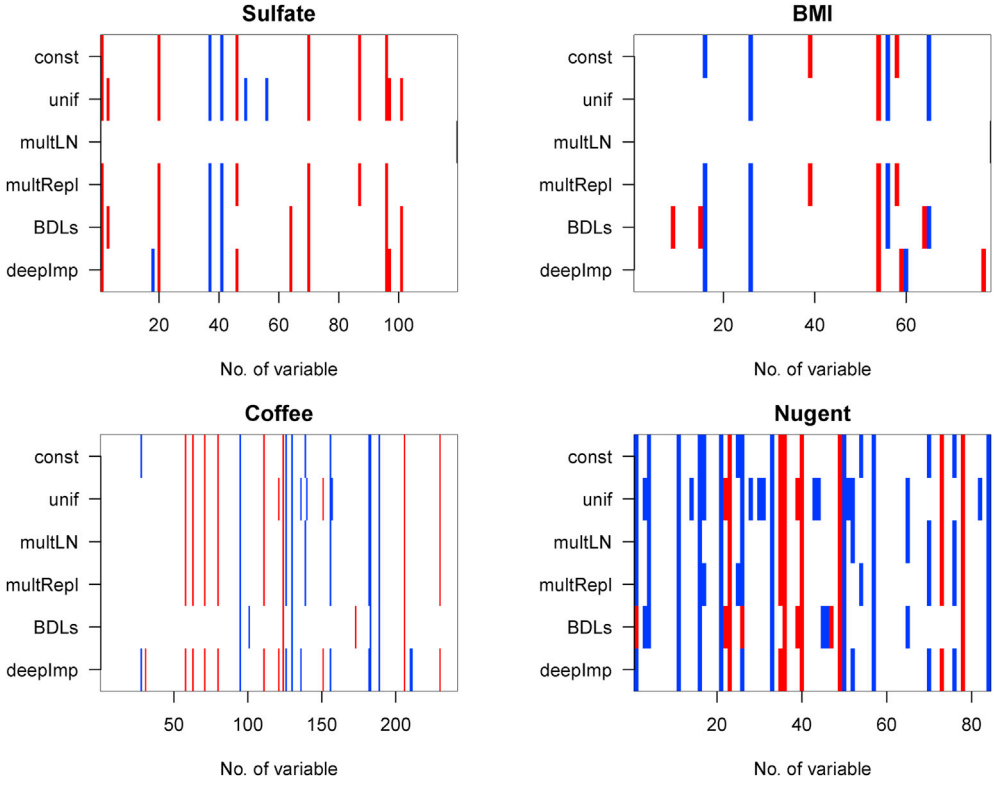

Fig. 7. Comparison of the sparsity of the regression coefficients as outcome from the models based on differently imputed data. Vertical bars indicate non-zero coefficients, blue color for negative and red color for positive coefficient (aggregated from 50 replications). (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

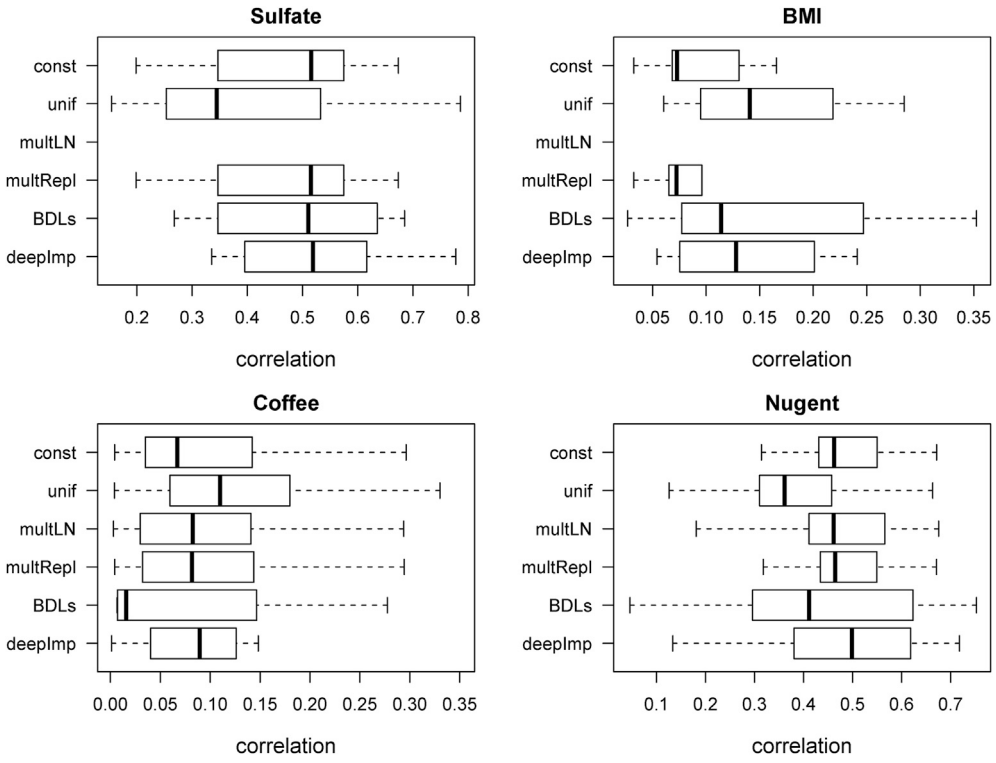

Fig. 8. Absolute correlations (summarized by boxplots) of the response with all variables corresponding to the non-zero coefficients, see Fig. 7.

considered the regression setting, where a real response was regressed on the imputed composition.

The main findings of this study are summarized and discussed below:

- Some of the considered replacement methods (*multLN*, *multRepl*) did not work if more than about half of the entries in the compositional data matrix were zero. This, however, also depends on how the zeros are distributed among the variables. Finally, the applicability also depends on how the algorithms are implemented, and there may still be possibilities to improve the behavior.
- The simplest strategies *const* and *unif* trivially work even if the percentage of zeros in the data gets very high – but also the much more complex algorithms *BDLs* and *deepImp* still work in such a situation. The price to pay is a much longer computation time (see Table 2).
- The simulation results reveal that if the zero proportion gets very high, say more than 0.8, the distances get distorted – only *unif* still gives stable results. Also the correlation structure gets distorted at some point, and *const*, *unif* and *deepImp* are quite reliable (Fig. 3). *BDLs* has the problem of yielding instable results (Fig. 4). This goes back to the selection of an appropriate number of PLS components, which is based on bootstrapping. Increasing the number of bootstrap samples could resolve the issue, however, this would increase the computation time even further.
- The example datasets did not go towards the limits concerning the proportion of zeros (only up to 82%), and thus we obtained results in almost all cases. The distribution of the imputed values naturally differs a lot for the different replacement algorithms, but since the response is not considered for the imputation, it is not so clear what the effects are on the resulting regression models. Then it also depends on the purpose of the model: Is the main focus on a reliable determination of OTUs in order to interpret the relationships with the response? Or is the focus on prediction accuracy? Often, both aspects are important, and we observed that *BDLs* and *unif* lead to somewhat different OTUs. This does not necessarily have a big impact on the prediction quality of the model, but the correlations of several of those OTUs with the response are typically lower [28].
- Fig. 6 revealed a clearly better prediction performance of the algorithm *deepImp* for the *Nugent* dataset. This dataset has almost 400 observations, and thus clearly more than the other datasets. Also our simulations have been conducted on datasets with 400 observations. Templ [10] showed (although for datasets with small amounts of zeros) that *deepImp* becomes competitive once the sample size is in the hundreds, and delivers increasingly better results with increasing sample size. The *Nugent* dataset has 82% zeros, and for such a high fraction, the errors caused by the replacement in the distance or the correlation structure explode (Fig. 3). The method *unif* is still quite reliable in this respect, but seems rather unreliable to identify the important OTUs (Fig. 7).

A final recommendation for an appropriate replacement method depends on the purpose of the analysis, on the dataset at hand (dimension, fraction of zeros), and on the available time to spend on replacing the zeros. If there is a time constraint, one should not use *BDLs* and *deepImp*. If not, and in particular if the dataset has a reasonably high number of observations (also compared to the number of variables), we recommend the algorithm *deepImp*, because this procedure seems to replace the zeros in a way which is well adjusted to the multivariate data structure imposed by the possibly few non-zeros. This is an important basis for statistical modeling, where prediction accuracy or the identification of marker variables are of concern.

Declaration of competing interest

The authors declare that they have no known competing fianancial interests or personal relationships that could have appeared to inuence the work reported in this paper.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2021.104248.

References

[1] P. Filzmoser, K. Hron, M. Templ, Applied Compositional Data Analysis. Springer: Springer Nature, 2018.

[2] P. Filzmoser, K. Hron, Compositional data analysis in chemometrics, in: S. Brown, R. Tauler, B. Walczak (Eds.), Comprehensive Chemometrics: Chemical and Biochemical Data Analysis, second ed., Elsevier, Amsterdam, 2020, pp. 641–662.

[3] S. Weiss, Z.Z. Xu, S. Peddada, A. Amir, K. Bittinger, A. Gonzalez, C. Lozupone, J.R. Zaneveld, Y. Vázquez-Baeza, A. Birmingham, E.R. Hyde, Normalization and microbial differential abundance strategies depend upon data characteristics, Microbiome 5 (2017) 27, https://doi.org/10.1186/s40168-017-0237-y.

[4] G.B. Gloor, J.M. Macklaim, V. Pawlowsky-Glahn, J.J. Egozcue, Microbiome datasets are compositional: and this is not optional, Front. Microbiol. 8 (2017) 2224, https://doi.org/10.3389/fmicb.2017.02224.

[5] R Core Team, R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, URL, https://www.R-project.org/, 2020.

[6] S. Mandal, W. Van Treuren, R.A. White, M. Eggesbø, R. Knight, S.D. Peddada, Analysis of composition of microbiomes: a novel method for studying microbial composition, *Microb. Ecol. Health Dis.* 26 (2015) 27663, https://doi.org/10.3402/mehd.v26.27663.

[7] J.A. Martín-Fernández, C. Barcelól Vidal, V. Pawlowsky-Glahn, Dealing with zeros and missing values in compositional data sets using nonparametric imputation, Math. Geol. 35 (2003) 253–278, https://doi.org/10.1023/A:1023866030544.

[8] J. Palarea-Albaladejo, J.A. Martín-Fernández, R package for multivariate imputation of left-censored data under a compositional approach, *Chemometr. Intell. Lab. Syst.* 143 (2015) 85–96, https://doi.org/10.1016/j.chemolab.2015.02.019.

[9] M. Templ, K. Hron, P. Filzmoser, A. Gardlo, Imputation of rounded zeros for high-dimensional compositional data, Chemometr. Intell. Lab. Syst. 155 (2016) 183–190, https://doi.org/10.1016/j.chemolab.2016.04.011.

[10] M. Templ, Artificial Neural Networks to Impute Rounded Zeros in Compositional Data. arXiv:2012.10300, 2020.

[11] J. Palarea-Albaladejo, J.A. Martín-Fernández, Values below detection limit in compositional chemical data, Anal. Chim. Acta 764 (2013) 32–43, https://doi.org/10.1016/j.aca.2012.12.029.

[12] J. Palarea-Albaladejo, J.A. Martín-Fernández, A modified EM alr-algorithm for replacing rounded zeros in compositional data sets, Comput. Geosci. 34 (2008) 902–917, https://doi.org/10.1016/j.cageo.2007.09.015.

[13] J.A. Martín-Fernández, K. Hron, M. Templ, P. Filzmoser, J. Palarea-Albaladejo, Model-based replacement of rounded zeros in compositional data: classical and robust approaches, Comput. Stat. Data Anal. 56 (2012) 2688–2704, https://doi.org/10.1016/j.csda.2012.02.012.

[14] K. Hron, M. Templ, P. Filzmoser, Imputation of missing values for compositional data using classical and robust methods, Comput. Stat. Data Anal. 54 (2010) 3095–3107, https://doi.org/10.1016/j.csda.2009.11.023.

[15] M. Templ, K. Hron, P. Filzmoser, robCompositions: an R-package for robust statistical analysis of compositional data, in: V. Pawlowsky-Glahn, A. Buccianti (Eds.), Compositional Data Analysis. Theory and Applications, John Wiley & Sons, Chichester (UK), 2011, pp. 341–355.

[16] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, second ed., Wiley, New York, 2002.

[17] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on Image Net classification, arXiv, 1502 (2015), 01852.

[18] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Presentations (ICLR), arXiv, abs/1412.6980, 2014.

[19] K. Hron, M. Engle, P. Filzmoser, E. Fišerová, Weighted symmetric pivot coordinates for compositional data with geochemical applications, Math. Geosci. (2020), https://doi.org/10.1007/s11004-020-09862-5.

[20] B. Schloerke, geozoo: zoo of Geometric Objects. R package version 0.5.1. https://CRAN.R-project.org/package=geozoo, 2016.

[21] P. Kynčlová, K. Hron, P. Filzmoser, Correlation between compositional parts based on symmetric balances, Math. Geosci. 49 (2017) 777–796, https://doi.org/10.1007/s11004-016-9669-3.

[22] J. Aitchison, The statistical analysis of compositional data, J. Roy. Stat. Soc. B 44 (1982) 139–160.

[23] M. Altenbuchinger, T. Rehberg, H.U. Zacharias, F. Stämmler, K. Dettmer, D. Weber, A. Hiergeist, A. Gessner, E. Holler, P.J. Oefner, R. Spang, Reference point insensitive molecular data analysis, Bioinformatics 33 (2017) 219–226, https://doi.org/10.1093/bioinformatics/btw598.

[24] G.D. Wu, J. Chen, C. Hoffmann, K. Bittinger, Y.Y. Chen, S.A. Keilbaugh, M. Bewtra, D. Knights, W.A. Walters, R. Knight, R. Sinha, Linking long-term dietary patterns with gut microbial enterotypes, Science 334 (2011) 105–108, https://doi.org/10.1126/science.1208344.

[25] M. Jaquet, I. Rochat, J. Moulin, C. Cavin, R. Bibiloni, Impact of coffee consumption on the gut microbiota: a human volunteerstudy, Int. J. Food Microbiol. 130 (2009) 117–121, https://doi.org/10.1016/j.ijfoodmicro.2009.01.011.

[26] J. Ravel, P. Gajer, Z. Abdo, G.M. Schneider, S.S. Koenig, S.L. McCulle, S. Karlebach, R. Gorle, J. Russell, C.O. Tacket, R.M. Brotman, Vaginal microbiome of reproductive-age women, Proc. Natl. Acad. Sci. Unit. States Am. 108 (Supplement 1) (2011) 4680–4687, https://doi.org/10.1073/pnas.1002611107.

[27] W. Lin, P. Shi, R. Feng, H. Li, Variable selection in regression with compositional covariates, Biometrika 1014 (2014) 785–797, https://doi.org/10.1093/biomet/asu031.

[28] J. Xiao, L. Chen, Y. Yu, X. Zhang, J. Chen, A phylogeny-regularized sparse regression model for predictive modeling of microbial community data, Front. Microbiol. 9 (2018) 3112, https://doi.org/10.3389/fmicb.2018.03112.