Received: 15 April 2021 | Revised: 6 July 2021 | Accepted: 10 August 2021

DOI: 10.1002/psp4.12704

ARTICLE

Visually guided preprocessing of bioanalytical laboratory data using an interactive R notebook (*pguIMP*)

Sebastian Malkusch[1] | **Lisa Hahnefeld**[1] | **Robert Gurke**[1,2] | **Jörn Lötsch**[1,2]

[1] Institute of Clinical Pharmacology, Goethe–University, Frankfurt am Main, Germany

[2] Fraunhofer Institute for Translational Medicine and Pharmacology ITMP, Frankfurt am Main, Germany

Correspondence
Jörn Lötsch, Goethe–University, Theodor–Stern–Kai 7, 60590 Frankfurt am Main, Germany.
Email: j.loetsch@em.uni-frankfurt.de

Funding information
This work was funded by the Landesoffensive zur Entwicklung wissenschaftlich-ökonomischer Exzellenz (LOEWE), LOEWE-Zentrum für Translationale Medizin und Pharmakologie (J.L.), in particular through the project “Reproducible Cleaning of Biomedical Laboratory Data Using Methods of Visualization, Error Correction and Transformation Implemented as Interactive R-Notebooks” (J.L.). The funders had no role in method design, data selection and analysis, decision to publish, or preparation of the manuscript.

Abstract

The evaluation of pharmacological data using machine learning requires high data quality. Therefore, data preprocessing, that is, cleaning analytical laboratory errors, replacing missing values or outliers, and transforming data adequately before actual data analysis, is crucial. Because current tools available for this purpose often require programming skills, preprocessing tools with graphical user interfaces that can be used interactively are needed. In collaboration between data scientists and experts in bioanalytical diagnostics, a graphical software package for data preprocessing called *pguIMP* is proposed, which contains a fixed sequence of preprocessing steps to enable reproducible interactive data preprocessing. As an R-based package, it also allows direct integration into this data science environment without requiring any programming knowledge. The implementation of contemporary data processing methods, including machine-learning-based imputation techniques, ensures the generation of corrected and cleaned bioanalytical data sets that preserve data structures such as clusters better than is possible with classical methods. This was evaluated on bioanalytical data sets from lipidomics and drug research using k-nearest-neighbors-based imputation followed by k-means clustering and density-based spatial clustering of applications with noise. The R package provides a Shiny-based web interface designed to be easy to use for non–data analysis experts. It is demonstrated that the spectrum of methods provided is suitable as a standard pipeline for preprocessing bioanalytical data in biomedical research domains. The R package *pguIMP* is freely available at the comprehensive R archive network (https://cran.r-project.org/web/packages/pguIMP/index.html).

Study Highlights

WHAT IS THE CURRENT KNOWLEDGE ON THE TOPIC?

The evaluation of bioanalytical data by means of classical statistics, machine-learning-driven approaches, or by pharmacokinetic–pharmacodynamic modeling places high demands on data quality, which is ensured by data preprocessing.

WHAT QUESTION DID THIS STUDY ADDRESS?

This study introduces a software package for data preprocessing that enables field experts without programming knowledge to prepare bioanalytical data for use in machine-learning-based drug discovery. Together with the software, a sequence of preprocessing steps is proposed that prevents common pitfalls in data preprocessing.

WHAT DOES THIS STUDY ADD TO OUR KNOWLEDGE?

The need for such a package is demonstrated by attributing the erroneous assignment of bioanalytical data by unsupervised machine-learning models to the loss of information attributed to faulty data preprocessing.

HOW MIGHT THIS CHANGE DRUG DISCOVERY, DEVELOPMENT, AND/OR THERAPEUTICS?

As data preprocessing errors propagate and lead to flawed model predictions, they can skew the results of machine-learning-based biomarker or drug discovery programs.

INTRODUCTION

The establishment of high-throughput experiments in clinical research provide a wealth of data of diverse structure. The goal of systems pharmacology is to integrate these high-dimensional data into complex models that aid in decision making in the process of drug discovery or drug safety assessment. However, the often nonlinear and stochastic natures of these data pose a challenge for data processing, which is why data science methods have recently made their way into systems pharmacology.[1] Although differential equation systems–based pharmacokinetic and pharmacodynamic models describing the temporal evolution of a system, such as plasma concentrations or drug effects, are well established in the preprocessing of data sets from drug research and development,[2–6] these additional data pose new challenges to the preprocessing of drug discovery and development data sets. This is where the strength of data science comes into play, being able to extract knowledge from high-dimensional data, often by using machine-learning methods (for an overview, see Badillo et al.[7]).

The analysis of biomedical data by machine learning requires data that have been cleaned of analytical laboratory errors[8,9] and are adequately transformed and preferably free of missing values, anomalies,[10] or values below the limit of quantification (LOQ).[2,5] Although likelihood-based models have been shown to be particularly suitable for handling values below LOQ in pharmacokinetics mixed-effects models,[2–6] many proposed solutions to this problem in the area of pharmacological data science are data set specific[10,11] and must be tailored to analyses that use machine-learning algorithms. For example, for the treatment of missing values in gas chromatography–mass spectrometry metabolomics, predictive k-nearest neighbors (kNN), and random forest have proven to be particularly suitable.[12] Indeed, for data cleaning of high-dimensional bioanalytical data sets used in pharmacological research for biomarker identification by machine learning,[13] data science offers a wealth of imputation methods. However, not every method is suitable for every data set, and when choosing the imputation method, it is important to consider that the replacement of missing values has direct implications for further downstream analyses, for example, biomarker identification.[14]

The addition of these data science methods to pharmacological research requires user friendly, generalizable solutions that allow measurable and documented quality control of routine preprocessing of the data, especially because the aforementioned methods are often new to the research field. Unfortunately, available solutions are often limited to simple statistics-based imputations, such as substitution by the variable mean or median,[15] or require programming skills in common data science languages (Table S1), which is why a wider range of interactive preprocessing tools for bioanalytical data in drug research environments is a recurring desire. To address this need, an interactive data engineering package called *pguIMP*† is presented that covers the preprocessing steps of bioanalytical data identified in a multidisciplinary approach by data scientists and field experts. Its components provide visually guided, interactive tools for each major preprocessing step of bioanalytical laboratory data, including visualization, transformation, normalization, outlier removal, and imputation of missing values. The design allows free choice of the provided algorithms and separate treatment of outliers and values outside the LOQs. The package preprocesses data based on established methods, including statistical hypothesis testing of the distribution of the original or transformed data, and it provides statistics-based, machine-learning-derived methods for imputing missing values and removing outliers. It was previously shown that these methods do not cover all possible sources of error that may occur and that appropriate data visualizations, as implemented in the package, can assist the field expert in identifying these otherwise undetected errors in data sets. Because *pguIMP* is based on the R programming language, it fits seamlessly into this data science environment and offers the possibility of integration into more complex workflows and the application of additional methods from the large selection offered by the R environment.

METHODS

Implementation

The programming work for this report was performed in the R language,[16] which is available free of charge in the Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/. The *pguIMP* package for the reproducible cleaning of biomedical laboratory data is available via CRAN (https://cran.r-project.org/web/packages/pguIMP/index.html). A detailed description of the package can be found at https://cran.r-project.org/web/packages/pguIMP/pguIMP.pdf. Further technical details are described in the Supplementary Information of this report. The main steps of the data preprocessing workflow that can be performed with the *pguIMP* package are described in the next sections.

Data visualizations

To examine the distribution of the values of a variable, scatter plots, box plots, histogram bar plots, and probability density function (PDF) plots are available. The PDF can be shown using the standard R implementation or using the Pareto density estimation, which is a variant that estimates the PDF using hyperspheres and facilitates visual detection of a subgroup structure in the data.[17] It was designed to be particularly useful for detecting subgroups in the data that may be of interest for evaluating drug effects. The deviation of the variable distribution from a normal distribution is shown via quantile–quantile plots (Q-Q plots).[18]

Data transformations

The transformation of skewed variable distributions into a more normal form, as implemented in *pguIMP*, follows the idea of Tukey's ladder of powers (LOP)[19] (for detailed information, see the "Data Transformation" section in the Supplementary Information). If the transformation result by Tukey's LOP is not satisfactory, *pguIMP* alternatively offers a Box-Cox power transformation[20] as well as common parameter-free transformations (e.g., the binary logarithm Lb(x) with a base of 2, the natural logarithm Ln(x) that uses Euler's number as the base, and the common or decadic logarithm Lg(x) with a base of 10).

Data normalization

The *pguIMP* package provides three common scaling methods, that is, minimum–maximum normalization, mean normalization, and z score normalization (for detailed information, see the "Data Normalization" section in the Supplementary Information).

Outlier detection

Outliers are extreme values that lie outside the expected range of values, but whose occurrence can be attributed to various causes (e.g., measurement errors, data transmission errors, legitimate extreme values). Because the occurrence of outliers may negatively impact on the generalizability of predictive models, their identification and, if necessary, elimination during data preprocessing is mandatory. The *pguIMP* package offers multiple methods for univariate outlier detection. These can be divided into statistical methods implemented as the Grubb's test for outliers[21] or machine-learning-based methods such as the density-based spatial clustering of applications with noise (DBSCAN)[22] and distance-based methods such as the one-support vector machine class[23] and the kNN method.[24]

Imputation

The *pguIMP* package offers two types of imputation methods for numerical data comprising (a) substitution by certain scalars (i.e., median or mean) or (b) by values machine learned from the available data in a multivariate manner. (So far, the available models are distance-based models such as kNN[24] or predictive mean matching [PMM][25,26] and tree-based models such as M5P[27,28] or classification and regression trees [CARTs][29] as well as subsymbolic ensemble models such as random forests.[30] The imputation of missing values by machine learning is briefly explained using the kNN algorithm as an example in the Supplementary Information.)

Evaluation

The evaluation of the R package *pguIMP* aimed to assess the suitability of the implemented workflow for real bioanalytical data. Specific parts where the choice between implemented methods could lead to significant consequences for subsequent data analyses were evaluated separately in different experimental scenarios. In particular, the extent was analyzed to which methods of data normalization, transformation, and imputation influence the structure of the data set evaluated in downstream analyses such as the detection of "healthy" versus "diseased" group structures by measuring clusters. Of note, clustering is not implemented in the *pguIMP* package, but was used from external R standard libraries as a typical type of analysis performed after preprocessing the data, for example, with the *pguIMP* package.

Data sets

Bioanalytical data sets were available from the published studies; experimental details of data collection and laboratory analyses were described in detail in the respective reports.[11,31,32] Data Set 1, which was initially used for biomarker identification for dementia,[31] includes plasma concentrations of $d = 35$ different lipid mediators and other endogenous metabolites from $n = 94$ subjects, measured by means of liquid chromatography–electrospray ionization–tandem mass spectrometry. The liquid chromatography–electrospray ionization–tandem mass spectrometry methods were validated according to the criteria by the United States Food and Drug Administration.[33] Values outside of the validated concentration limits were initially excluded, and compounds with more than 20% missing values were not further investigated. For the remaining compounds, if possible, values below lower limits of quantification (LLOQs) but above the limits of detection were imputed with the measured value as the measurement error is still considered to be lower than the error due to statistical imputation.[34] For simplicity, the data set was reduced in the filtering procedure of *pguIMP* to $d = 8$ lipid mediators (S1P, C16Sphinganin, C16Cer, C20Cer, C24Cer, C24_1Cer, C16GluCer, C16LacCer) previously identified as informative in relation to psychiatric diagnosis.[31] The reduced data set contained a total of $n = 7$ values below the LLOQs, all in C16Cer, which were initially imputed with the measured value after review by the responsible analyst based on published recommendations.[35] Data Set 2 was previously used in a pharmacogenetic experiment assessing the formation of morphine from codeine in the presence of variants in cytochrome P450 2D6.[32] The set analyzed in the present experiments contains urine concentrations of the relevant metabolites of codeine, including codeine-6-glucuronide, morphine, morphine-3-glucuronide, and morphine-6-glucuronide. All were measured by means of mass spectrometry analysis in $n = 50$ healthy subjects as described with the respective main report.[32] The data set has no missing values. Data Set 3 comprises liquid chromatography–mass spectrometry data from cell samples originally published in a tutorial on lipidomic data analysis.[11] It originally included concentrations of $d = 212$ lipids measured in $n = 18$ samples. For simplicity, the data set was reduced to 6 lipids (2, 10, 140, 170, 171, 175) during filtering, none of which had missing values. The samples are divided into three subgroups consisting of a control group (C) and two differently treated groups (here for simplicity termed A and B; for further information, see the original literature[11]).

Experimentation

To evaluate the usefulness of the *pguIMP* package, five different experiments were conducted aimed at (a) comparing the transformation methods in terms of their normalization ability, (b) comparing the substitution possibilities of values outside the quantification limits and their effects on the distribution of the processed variables (Supplementary Material: Supplementary experiment 1), (c) characterizing the randomness of the occurrence of missing values, (d) comparing imputation methods in terms of their imputation error, and (e) evaluating the consequences of outlier imputation for the subsequent detection of subgroup structures in the data set. Furthermore, two additional experiments were conducted in which, first, the direct impact of data imputation on information loss due to dimensionality reduction is investigated (Supplementary Material: Supplementary experiment 2) and, second, the consequences of outlier imputation for the subsequent detection of subgroup structures was reproduced using Weka as an alternative to *pguIMP* (Supplementary Material: Supplementary experiment 3).

Comparison of transformation methods with respect to their normalization capability

The normalization capability of various data transformation methods was validated on Data Set 1 using Tukey's LOP[19] with various values of $\lambda = [2, 1, 0, -1]$ (Equation S1). Subsequently, the distributions of the transformed variables were tested for normality using Shapiro–Wilk[36] and Lilliefors' Kolmogorov–Smirnov tests.[37]

Characterization of the randomness of missing values

For the handling of missing values, it is essential to find out whether the entries of the respective instances are missing completely at random (CAR) or not at random (NAR). For this purpose, *pguIMP* maps the relationships between missing and observed values of a variable in the form of a pairwise comparison of the instance values of the remaining variables. Subsequently, the distribution pairs are compared using the Kruskal–Wallis test.[38] In the case where missing values occur CAR, no significant difference is expected between the pairwise distributions.[14]

To analyze missing values in this context, two incomplete data sets were generated from the complete Data Set 2 using the "simulateMissings" function from the *compositions* package.[39] In each of the data sets, values were removed with a probability of $p = 0.1$ either CAR or NAR. According to the reference manual of the *compositions* package, the CAR method of the "simulateMissings" function removes a value from the data with a probability that is independent of each variable. The NAR method removes small values with a higher probability.[39] Subsequently, the two data sets were analyzed according to the method described previously using the "missing_pairs" function of the package *finalfit*.[40] The experiment was repeated 100 times, and the fraction of significantly different distributions per iteration was documented. Finally, the documented fractions of significantly different distributions in CAR and NAR data sets were compared using the independent two-group Mann–Whitney U test.[41]

Validation of imputation methods

From the complete Data Set 2, instance values were removed with a probability of $p = 0.1$ as described previously. The resulting imperfect data sets were transformed according to Tukey's ladder of powers[19] with optimized λ (Equation S1) and then minimum–maximum normalized (Equation S2). Subsequently, missing values were substituted by different values: the variable mean, the variable median, or values machine learned from the remaining variables using different models (CART,[29] kNN,[24] PMM[25,26]). Finally, the imputed data were transformed back to their original state, and substituted values were compared with the original values in the form of the root mean squared percentage error (RMSPE).[42] The experiment was performed with missing values simulated CAR and NAR, and each procedure was repeated 100 times.

Estimation of consequences of outlier imputation for data set subgroup structure determination

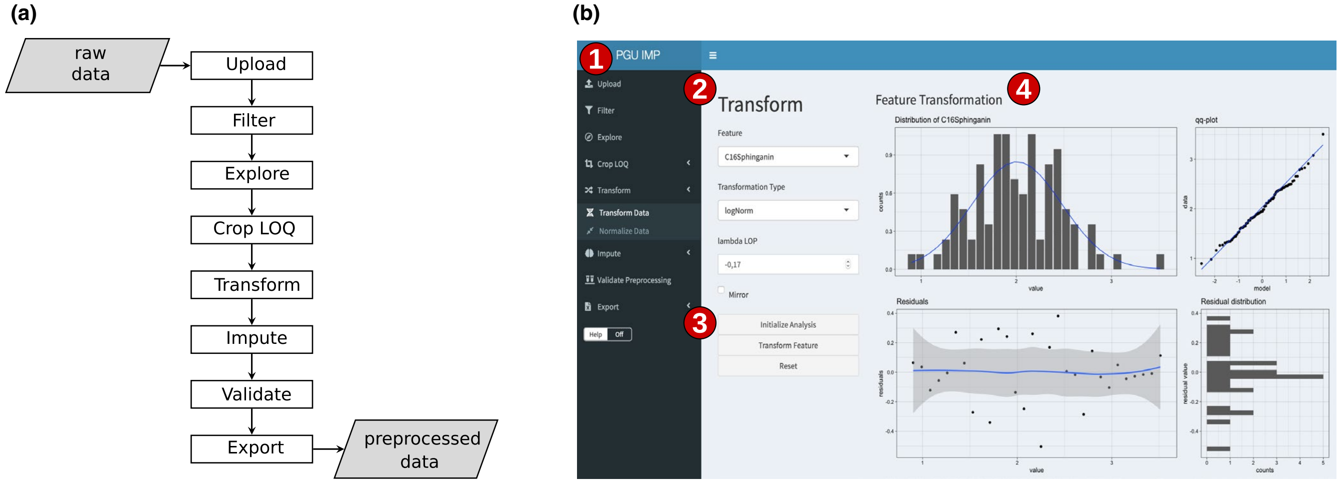
Consequences of outliers and their imputation were further evaluated in the context of clustering as a common task in biomedical data analysis to identify subgroups. As mentioned previously, clustering is not implemented in the *pguIMP* package because it is not part of the data preprocessing. The present experiment deals with the consequences of imputation for clustering as an example of a typical downstream analysis. For this purpose, an attempt was made to reconstruct the original group structure of Data Set 3 from the dimensionality reduced data sets (X_{PCA}) using centroid-based and density-based methods of unsupervised learning. The results were compared with the known three-group structure described previously. Centroid-based clustering was realized by training a k-means model[43,44] on the X_{PCA} data using the "kmeans" function of the *stats* package.[16] Graphical validation of the procedure was done with a scatter plot of the X_{PCA} data color coded by the true underlying group structure, superimposing the proposed groups as convex hulls. Alternatively, density-based clustering was performed by training the ordering of points to identify the clustering structure (OPTICS) algorithm implemented in the *dbscan* package[45] on the X_{PCA} data with hyperparameters $\varepsilon = 2$ and minPoints = 3. The hierarchical clustering structure was visualized as a dendrogram and reachability distance plot. Subsequently, the DBSCAN cluster structure was extracted from the OPTICS result by defining a reachability-distance threshold that would result in a clustering solution matching the true number of clusters using the "extractDBSCAN" function of the *dbscan* package.[45] Graphical data representation was realized using the *ggplot2* package, which is part of the *tidyverse* package.[46]

RESULTS

The workflow of data preprocessing with *pguIMP*, including visual inspection, error correction, outlier detection, and imputation of missing values, is shown as a flowchart in Figure 1a, and insights into the graphical user interface of the package are given in Figure 1b.

Quantification of the normalization capability of transformation methods

For various transformations commonly used for bioanalytical data of concentrations in biological materials, it was observed that the Ln(x) transformation resulted in the smallest deviation from normality as indicated by nonsignificant outcomes of three different tests comparing the observed with a normal distribution of the data (Figure 1b and Table S2). This is consistent with the common independent observations that bioanalytical variables are often positively skewed and that a logarithmic transformation often results in a normal distribution of the variable, for example, for the common continuous noncompartmental pharmacokinetic data.[47]



Transform

Feature: C16Sphinganin

Transformation Type: logNorm

lambda LOP: -0,17

□ Mirror

Initialize Analysis

Transform Feature

Reset

FIGURE 1 (a) Flowchart of the data engineering pipeline as it is used in the *pguIMP* package. The sequence of the individual processes is predefined. The user can choose from different algorithms under each subprocess and adjust the respective process parameters. The user can return to all subprocesses and change algorithms or optimize their parameters if the validation results of the pipeline created are not satisfactory. The result of such an iterative optimization routine is an individual, problem-specific preprocessing pipeline that prepares the data set for the following chemometric analyses. (b) Screenshot of the graphical user interface of *pguIMP*. (1) The navigation menu under which the individual preprocessing steps are listed. In the example shown, the Transform process is selected. (2) The user can select the parameters for the respective analysis. In the case presented, the user would like to log-normally transform the lipid mediator C16Sphinganin. (3) The user ran the preprocessing step using the parameters chosen in (2). (4) After the preprocessing step has been performed, a graphical validation of the process is shown. In the particular case, the deviation of the transformed lipid mediator distribution from a normal distribution is depicted via an overlay (upper left) of the transformed lipid mediator distribution (bar diagram) and the normal distribution (line plot): the residuals between the two distributions (lower left), a quantile–quantile plot (upper right), and the residual distribution (lower right). (LOQ, limit of quantification)

Characterization of the randomness of missing values

Figure 2a visualizes the entries of the distribution matrix of a data set with CAR-simulated misses as an example. Here, none of the pairs of distributions shown exhibit significant differences. Contrasted in Figure 2b are the entries of the distribution matrix of a data set with NAR-simulated defects. Of the distribution pairs, 35% that have significant differences are shown. After repeating the experiment 100 times, it is found that the distribution matrix of the data sets with NAR-simulated misses has a significantly higher fraction of difference distributions than do the entries of the distribution matrix of the data sets with CAR-simulated misses ($p = 8.9772 \times 10^{-5}$, Mann–Whitney U test; Figure 2c).

Validation of different imputation methods

In general, substituting missing values with the variable mean or variable median results in a higher RMSPE than does substitution with machine-learned values from the remaining variables (Figure 3). For missing values occurring CAR, the median error of the mean-based/median-based substitution is in a similar range as the median error of the machine-learning-based substitution methods (Figure 3a). However, the dispersion of the error values is higher with mean-based/median-based substitution. The situation is different for missing values that occur NAR (Figure 3b). Here, all methods make a bigger mistake than they do in imputing CAR missing values. However, the median error of mean-based/median-based substitution is an order of magnitude higher than the median error of machine-learning-based imputation methods, suggesting a systematic error of the first two methods. For Data Set 2, kNN-based imputation seems to provide the most robust results regardless of the nature of the missing values.

Consequences of outlier imputation for data set subgroup structures

Consistent with the results of the preceding experiments, clustering methods such as k-means and DBSCAN were able to reproduce the true cluster separation in Data Set 3 when trained on the kNN-imputed data set (Figure 4l–p). By contrast, both clustering algorithms produced erroneous solutions when trained with either the control data set (Figure 4) or when outliers had been substituted by the

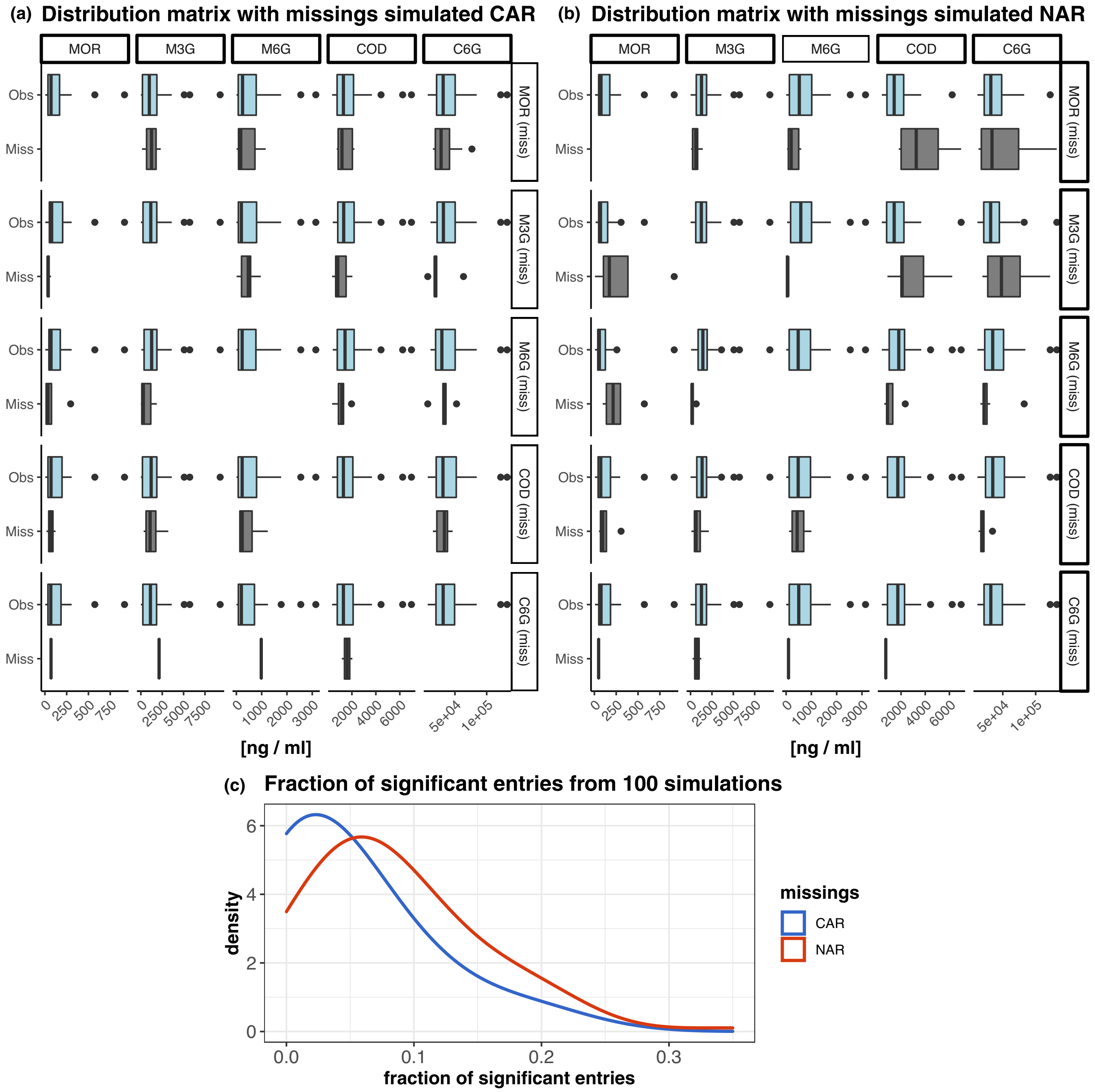


FIGURE 2 Inference about the origin of missing values. Missing values have been simulated either (a) completely at random (CAR) or (b) not at random (NAR). For each variable, the data set was divided into two groups. The first group comprises the instances that were observed in the respective variable (Obs). The second group comprises the instances that were missing in the respective variable (Miss). The value distributions of the two groups were plotted for the remaining variables. This procedure is repeated row-wise for all variables resulting in a distribution matrix. (c) The probability density function of the sum of significantly different groups per distribution matrix throughout 100 experiments. Significance was tested using the Kruskal–Wallis test with $\alpha = 0.05$. (C6G, codeine-6-glucuronide; COD, codeine; M3G, morphine-3-glucuronide; M6G, morphine-6-glucuronide; MOR, morphine)

variable median (Figure 4). Precisely, the k-means model mislabeled seven instances on the control data set and two instances in the median-imputed data set, whereas the DBSCAN model mislabeled five instances when on the control data set and two instances on the median-imputed data set.

DISCUSSION

The growing importance of machine learning in pharmacological research is accompanied by the advent of high-dimensional biomedical data sets. In contrast to classical pharmacometric models, which are based on

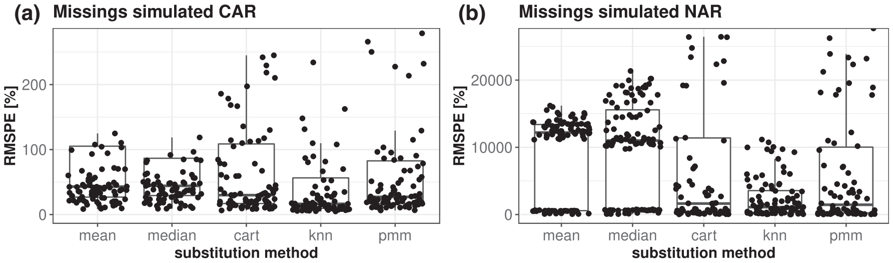


FIGURE 3 Errors of various imputation methods. Missing values have been simulated either (a) completely at random (CAR) or (b) not at random (NAR) and were subsequently substituted either by the variable mean or median value. Alternatively, the substitution values were machine learned from the remaining variables using the classification and regression tree (CART), k-nearest neighbors (knn), or predictive mean matching (pmm) algorithm. The error is calculated as root mean squared percentage error (RMSPE)

pharmacological principles and can therefore be interpreted physiologically, in machine-learning models the relationships are learned by a computer on the basis of a data set.[7] Such a data set consists of a list of data points, so-called instances, which in turn represent the entities to be studied, such as patients or individual cells. The data points in turn represent the totality of measured variables. These variables can be any measurable parameter: categorical variables such as gender or the division into smokers and nonsmokers are just as possible as numerical variables such as age, height, weight, or a biomarker concentration. All of these variables collected for an entity can be combined arbitrarily to a data point. The number of variables in a data point is also referred to as dimensionality. Pharmacological data sets used to train machine-learning models usually have a high dimensionality because they often contain different omics data or gene expression profiles of the entities. In general, machine-learning models are designed to handle high-dimensional data sets. However, their performance depends largely on the form in which the data are presented to them. It is not uncommon to train the models on derived data sets with transformed and normalized variable values. In addition, most machine-learning models do not tolerate missing values, which is why missing values are usually imputed before training. Because of the high variability of data sets, a high degree of domain knowledge is necessary for data preprocessing.[7]

The *pguIMP* package for tracible and reproducible preprocessing of pharmacological data sets offers a variety of algorithms for data transformation, normalization, and imputation, which can be variably combined to form a data set specific, individual solution. This is done with the use of interactive dashboards on which the results of the individual analyses are presented graphically. This type of graphical result presentation provides a platform that can either be used by domain experts and data scientists to discuss the results of data preprocessing from different perspectives. In this way, a suitable solution for a data set can be approached individually.

During the development of the package, great emphasis was placed on the traceability and reproducibility of the data preprocessing, as these are basic concepts of good laboratory practices.[48] The traceability is guaranteed by the documentation of the package as well as the public availability of the source code.[49,50] The reproducibility of individual data preprocessing routines is guaranteed by detailed reports that archive results and all decisions and settings made by the user. At this point, it should be mentioned that some of the machine-learning models used for outlier detection and missing value imputation rely to some degree on randomness. The required random numbers are generated on the software side by so-called random number generators using deterministic mathematical functions. These generators deliver the same sequence of random numbers for the same starting point, the so-called seed value. Here, the reproducibility of the results of these randomness-based algorithms is guaranteed by the fact that the seed values are also archived.

One of the most challenging tasks in data preprocessing is dealing with missing values. The reasons for their occurrence in bioanalytical data sets are diverse. On the one hand, stochastically occurring errors in data acquisition can lead to a situation where the information content of individual measurements cannot be trusted. On the other hand, regular measurements of values below the LLOQ may have been removed from the data set on purpose and reported as "<LLOQ" instead. However, it has been shown that models trained on data containing values below LLOQ can be less erroneous than models trained on data where instances with variable values below the LLOQ were discarded or where the critical values were replaced by the variable LLOQ/2 during preprocessing[35] (Supplementary Material: Supplementary experiment 1). It is therefore essential for a preprocessing routine to separately process measurements that are missing because of stochastically occurring errors and measurements that are highly error prone because of a low value. From experience, however, one will be confronted with different

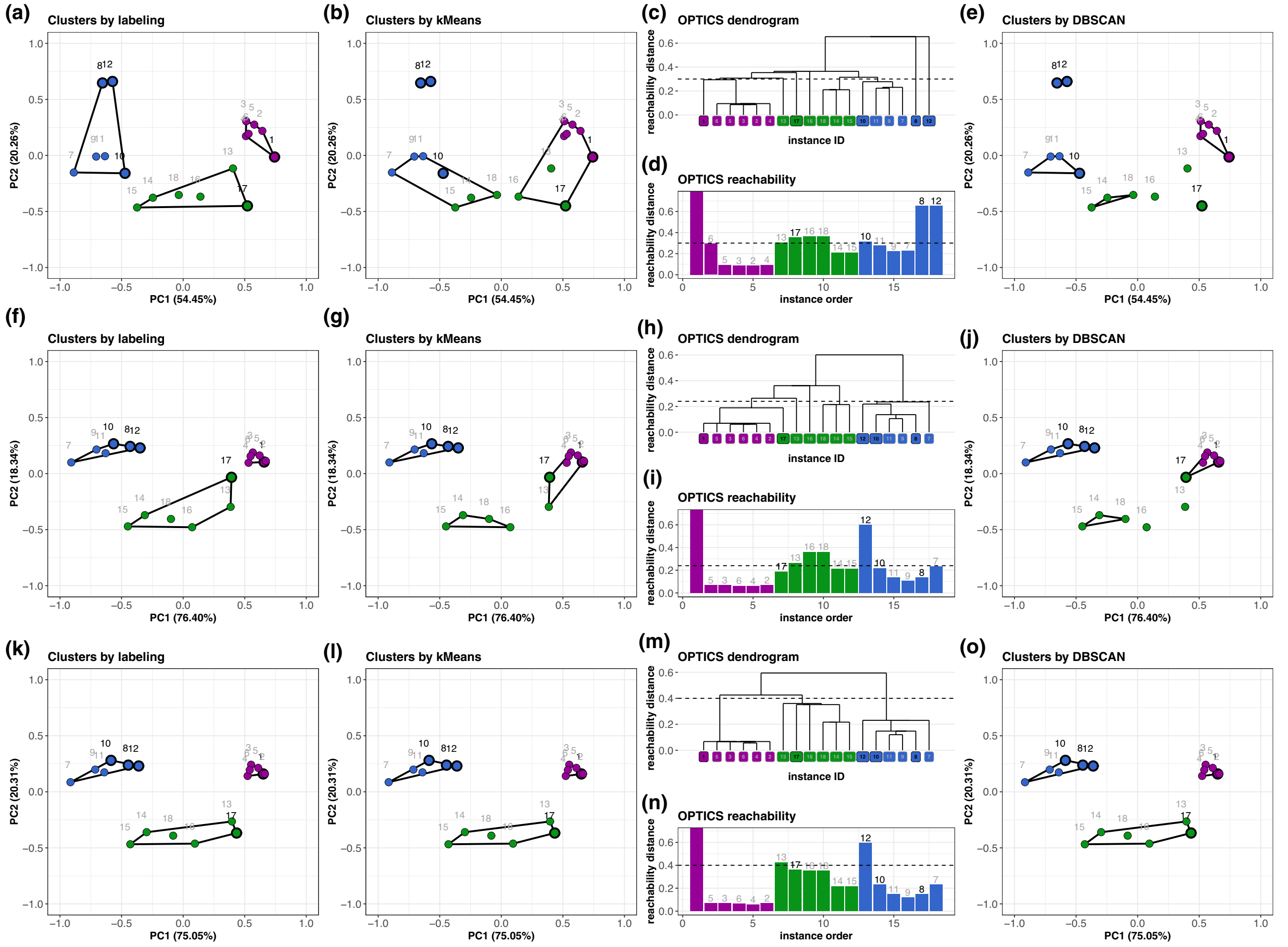


FIGURE 4 Graphical validation of the effect of data preprocessing on unsupervised cluster analysis using factorial instance plots on the principal component map. For all experiments, data preprocessing incorporated data transformation (Ln) and normalization (minimum–maximum). Outliers are defined variable-wise by using Grubb's test for outliers with α = 0.05. Variable values deviating from normality were identified in five instances (1, 8, 10, 12, 17). Further preprocessing incorporated three different methods of outlier handling: outliers were left untouched (Row 1; a–e), variable values in outlier instances were replaced by the respective variable median (Row 2; f–j), and variable values in outlier instances were imputed based on the remaining instances via k-nearest neighbors (Row 3; k–o). The cluster separation as proposed by various unsupervised cluster analysis methods trained on the first two principal components of the preprocessed data are each shown column-wise. (a, f, k) Black polygons visualize the cluster separation according to the original labeling of the data set. (b, g, l) Black polygons visualize the cluster separation following k-means clustering. (c, h, m) Dendrogram according to the ordering of points to identify the clustering structure (OPTICS). (d, i, n) Reachability plot according to OPTICS. (e, j, o) Black polygons visualize the cluster separation following density-based spatial clustering of applications with noise (DBSCAN) as extracted from the OPTICS analysis by applying a distance threshold (dashed line in c, h, m and d, i, n). The color code visualizes the true cluster separation as proposed by the original data labeling (Treatment A, blue; Treatment B, green; control, magenta). The numbers represent the instances of the data. Gray numbers indicate instances with regular variable values, black numbers indicate outlier instances. (ID, instance identification label; PC1, principal component 1; PC2, principal component 2)

data sets during data preprocessing. In few cases, it is a data set where missing values are because of stochastically occurring measurement errors and the measurements of variables whose value is below the LLOQ are given. Often, values below the LLOQ are masked or removed before the data is passed on. As a result, no information about the origin of the missing values is available. In this case, *pguIMP* offers two imputation approaches. In the first approach, the missing values are simply replaced by a variable characteristic such as the variable mean or median. In the second approach, the missing values are machine learned from the remaining values of all variables using predictive models. Recent benchmarks of different imputation methods show that machine-learning-based imputation methods such as kNN and random forests mostly outperform simple mean imputation when applied to data sets comprising missing values introduced to complete data sets either as CAR or NAR.[14]

The results of the benchmark study are consistent with the results presented here. For this reason, machine-learning-based imputation methods are usually preferable to simple replacement with the variable mean or median. However, there are conceivable situations in which a model generalizes poorly and cannot make meaningful predictions about missing values. Most machine-learning-based models predict the value of a missing variable based on the remaining variables' values. In case the machine-learning-based imputation methods do not provide satisfactory results, the *pguIMP* package will report an error message to the user and offers the possibility to substitute missing values by the variable mean or median. However, the choice of a suitable substitution method should always be preceded by an analysis of the origin of the missing values to avoid systematic errors, such as those caused by substituting the variable mean or median for NAR missing values.

The performance of classification models or cluster models, as used in pharmacological research to identify biomarkers from high-dimensional data sets,[13] is directly influenced by the upstream data preprocessing. The results of recent benchmark studies show that classifier models show higher performance when erroneous training data have been previously cleaned using machine-learning-based models, such as kNN or random forests.[14] These results could be reproduced within this study using downstream analyses such as clustering or dimension reduction (Supplementary Material S2) subsequently to data cleansing using the *pguIMP* package. With its strategy of taking into account the reason for the occurrence of missing data during imputation, *pguIMP* stands out from previous graphical solutions whose range of possible imputation methods in data preprocessing filters is limited to fix values or statistical solutions[15] (Supplementary Material: Supplementary experiment 3).

CONCLUSIONS

The R package *pguIMP* for the visually guided preprocessing of bioanalytical laboratory data was developed in close collaboration between data scientists and field experts in bioanalytical diagnostics. It provides a graphical user interface designed to be easy to use even for non–data analysis experts, and its application programming interface is also accessible from the command line using R scripts. It is available free of charge under version 3 of the GNU General Public License version 3 (GPLv3).

CONFLICT OF INTEREST

The authors declared no competing interests for this work.

AUTHOR CONTRIBUTIONS

S.M., L.H., R.G., and J.L. wrote the manuscript. S.M. and J.L. designed the research. J.L. aquired the funding. S.M. performed the research. S.M. analyzed the data.

ORCID

Sebastian Malkusch https://orcid.org/0000-0001-6766-140X
Lisa Hahnefeld https://orcid.org/0000-0002-0382-5695
Robert Gurke https://orcid.org/0000-0001-8218-1295
Jörn Lötsch https://orcid.org/0000-0002-5818-6958

ENDNOTE

† The name is a compound acronym: the domain name "pgu" stands for Pharmacology of Goethe University and "IMP" for imputation package.

REFERENCES

1. Hart T, Xie L. Providing data science support for systems pharmacology and its implications to drug discovery. *Expert Opin Drug Discov*. 2016;11(3):241-256.
2. Beal SL. Ways to fit a PK model with some data below the quantification limit. *J Pharmacokinet Pharmacodyn*. 2001;28(5):481-504.
3. Ahn JE, Karlsson MO, Dunne A, Ludden TM. Likelihood based approaches to handling data below the quantification limit using NONMEM VI. *J Pharmacokinet Pharmacodyn*. 2008;35(4):401-421.
4. Bergstrand M, Karlsson MO. Handling data below the limit of quantification in mixed effect models. *AAPS J*. 2009;11(2):371-380.
5. Senn S, Holford N, Hockey H. The ghosts of departed quantities: approaches to dealing with observations below the limit of quantitation. *Stat Med*. 2012;31(30):4280-4295.
6. Irby DJ, Ibrahim ME, Dauki AM, et al. Approaches to handling missing or "problematic" pharmacology data: Pharmacokinetics. *CPT: Pharm Syst Pharmacol*. 2021;10(4):291-308.
7. Badillo S, Banfai B, Birzele F, et al. An introduction to machine learning. *Clin Pharmacol Ther*. 2020;107(4):871-885.
8. Hyotylainen T, Oresic M. Bioanalytical techniques in nontargeted clinical lipidomics. *Bioanalysis*. 2016;8(4):351-364.
9. Lötsch J. Data visualizations to detect systematic errors in laboratory assay results. *Pharmacol Res Perspect*. 2017;5(6):e00369.
10. Kotsiantis SB, Kanellopoulos D, Pintelas PE. Data preprocessing for supervised leaning. *Proceedings of World Academy of Science, Engineering and Technology* 2006;12(2):278-283.
11. Checa A, Bedia C, Jaumot J. Lipidomic data analysis: Tutorial, practical guidelines and applications. *Anal Chim Acta*. 2015;885:1-16.
12. Gromski PS, Xu Y, Kotze HL, et al. Influence of missing values substitutes on multivariate analysis of metabolomics data. *Metabolites*. 2014;4(2):433-452.
13. Davis KD, Aghaeepour N, Ahn AH, et al. Discovery and validation of biomarkers to aid the development of safe and effective pain therapeutics: challenges and opportunities. *Nat Rev Neurol*. 2020;16(7):381-400.
14. Jäger S, Allhorn A, Bießmann F. A Benchmark for data imputation methods. *Frontiers in Big Data*. 2021;4(48).

15. Srivastava S. Weka: a tool for data preprocessing, classification, ensemble, clustering and association rule mining. *Int J Computer Appl*. 2014;88(10):26-29.
16. Ihaka R, Gentleman R. R: a language for data analysis and graphics. *J Computat Graph Stat*. 1996;5(3):299-314.
17. Ultsch A, Thrun MC, Hansen-Goos O, Lötsch J. Identification of molecular fingerprints in human heat pain thresholds by use of an interactive mixture model R toolbox (AdaptGauss). *Int J Mol Sci*. 2015;16(10):25897-25911.
18. Wilk MB, Gnanades R. Probability plotting methods for analysis of data. *Biometrika*. 1968;55(1):1-11.
19. Tukey JW. *Exploratory data analysis*, vol. 2. Reading, Mass.; 1977.
20. Box GEP, Cox DR. An analysis of transformations. *J Roy Stat Soc B*. 1964;26(2):211-252.
21. Grubbs FE. Sample criteria for testing outlying observations. *Ann Math Stat*. 1950;21(1):27-58.
22. Ester M, Kriegel H-P, Sander J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*. 1996;96:226-231.
23. Moya MM, Hush DR. Network constraints and multi-objective optimization for one-class classification. *Neural Networks*. 1996;9(3):463-474.
24. Cover T, Hart P. Nearest neighbor pattern classification. *IEEE Trans Inf Theory*. 1967;13(1):21-27.
25. Rubin DB. Statistical matching using file concatenation with adjusted weights and multiple imputations. *J Bus Econ Stat*. 1986;4(1):87-94.
26. Little RJA. Missing-data adjustments in large surveys. *J Bus Econ Stat*. 1988;6(3):287-296.
27. Quinlan JR. Learning with continuous classes. In: Proceedings from the 5th Australian Joint Conference on Artificial Intelligence; November 16-18. Vol. 92 1992; Hobart, Tasmania.
28. Wang Y, Witten IH. *Induction of model trees for predicting continuous classes*; 1996.
29. Breiman L. *Classification and regression trees*. Wadsworth International Group; 1984:358 p.
30. Ho TK. The random subspace method for constructing decision forests. *IEEE T Pattern Anal*. 1998;20(8):832-844.
31. Gurke R, Etyemez S, Prvulovic D, et al. A data science-based analysis points at distinct patterns of lipid mediator plasma concentrations in patients with dementia. *Front Psychiatry*. 2019;10:41.
32. Lötsch J, Rohrbacher M, Schmidt H, Doehring A, Brockmöller J, Geisslinger G. Can extremely low or high morphine formation from codeine be predicted prior to therapy initiation? *PAIN®*. 2009;144(1–2):119-124.
33. US Food and Drug Administration. *Bioanalytical Method Validation Guidance for Industry*. Silver Spring, MD: US Food and Drug Administration; 2018.
34. Harel O, Perkins N, Schisterman EF. The use of multiple imputation for data subject to limits of detection. *Sri Lankan J Appl Stat*. 2014;5(4):227-246.
35. Keizer RJ, Jansen RS, Rosing H, et al. Incorporation of concentration data below the limit of quantification in population pharmacokinetic analyses. *Pharmacol Res Perspect*. 2015;3(2):e00131.
36. Shapiro SS, Wilk MB. An analysis of variance test for normality (complete samples). *Biometrika*. 1965;52:591-611.
37. Lilliefors HW. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *J Am Stat Assoc*. 1967;62(318):399-402.
38. Kruskal WH, Wallis WA. Use of ranks in one-criterion variance analysis. *J Am Stat Assoc*. 1952;47(260):583-621.
39. van den Boogaart KG, Tolosana-Delgado R, Bren M. *Compositions: Compositional Data Analysis*. R package version 2.0-1; CRAN; 2021. Accessed September 6, 2021. https://CRAN.R-project.org/package=compositions
40. Harrison E, Drake T, Ots R. *finalfit: Quickly Create Elegant Regression Results Tables and Plots when Modelling*. R package version 1.0.2; CRAN; 2020. Accessed September 6, 2021. https://CRAN.R-project.org/package=finalfit
41. Mann HB, Whitney DR. On a test of whether one of two random variables is stochastically larger than the other. *Ann Math Stat*. 1947;18(1):50-60.
42. Shcherbakov MV, Brebels A, Shcherbakova NL, et al. A survey of forecast error measures. *World Appl Sci J*. 2013;24(24):171-176.
43. MacQueen J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley symposium on Mathematical Statistics and Probability*; 1967; Oakland, CA.
44. Lloyd SP. Least-squares quantization in Pcm. *IEEE Trans Inf Theory*. 1982;28(2):129-137.
45. Hahsler M, Piekenbrock M, Doran D. dbscan: Fast density-based clustering with R. *J Stat Softw*. 2019;91(1):1-30.
46. Wickham H, Averick M, Bryan J, et al. Welcome to the tidyverse. *J Open Source Softw*. 2019;4(43):1686.
47. Lacey LF, Keene ON, Pritchard JF, Bye A. Common noncompartmental pharmacokinetic variables: are they normally or log-normally distributed? *J Biopharm Stat*. 1997;7(1):171-178.
48. Jena GB, Chavan S. Implementation of Good Laboratory Practices (GLP) in basic scientific research: Translating the concept beyond regulatory compliance. *Regul Toxicol Pharmacol*. 2017;89:20-25.
49. Ferrero E, Brachat S, Jenkins JL, et al. Ten simple rules to power drug discovery with data science. *PLoS Comput Biol*. 2020;16(8):e1008126.
50. Peng RD. Reproducible research in computational science. *Science*. 2011;334(6060):1226-1227.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: Malkusch S, Hahnefeld L, Gurke R, Lötsch J. Visually guided preprocessing of bioanalytical laboratory data using an interactive R notebook (*pguIMP*). *CPT Pharmacometrics Syst Pharmacol*. 2021;10:1371–1381. https://doi.org/10.1002/psp4.12704