# Supplementary information on visually guided preprocessing of bioanalytical laboratory data using an interactive R notebook (pguIMP)

Sebastian Malkusch1, Lisa Hahnefeld1, Robert Gurke1,2, and Jörn Lötsch1,2

1 Institute of Clinical Pharmacology, Goethe - University, Theodor Stern Kai 7, 60590 Frankfurt am Main, Germany

2 Fraunhofer Institute for Translational Medicine and Pharmacology ITMP, Theodor-Stern-Kai 7, 60596 Frankfurt am Main, Germany

**Correspondence to:** Dr. Sebastian Malkusch, Goethe - University, Theodor - Stern - Kai 7, 60590 Frankfurt am Main, Germany, e-mail: [malkusch@med.uni-frankfurt.de](mailto:malkusch@med.uni-frankfurt.de); Phone: +49-69-6301-87818, Fax: +49-69-6301-7637

[Technical details 4](#_Toc79150101)

[Implementation 4](#_Toc79150102)

[Installation 4](#_Toc79150103)

[Dependencies 4](#_Toc79150104)

[Data transformation 5](#_Toc79150105)

[Data normalization 5](#_Toc79150106)

[Machine-learned imputation of missing values by k-nearest neighbors 7](#_Toc79150107)

[Exploratory statistical hypothesis testing 7](#_Toc79150108)

[Documentation 8](#_Toc79150109)

[Supplementary experiment 1: Comparison of substitution methods for LOQ outliers 9](#_Toc79150110)

[Introduction 9](#_Toc79150111)

[Data sets 9](#_Toc79150112)

[Methods 9](#_Toc79150113)

[Comparison of substitution methods for LOQ outliers 10](#_Toc79150114)

[Supplementary experiment 2: Assessment of the impact of data imputation on the information loss due to dimension reduction 11](#_Toc79150115)

[Introduction 11](#_Toc79150116)

[Data sets 11](#_Toc79150117)

[Methods 11](#_Toc79150118)

[Results 12](#_Toc79150119)

[Supplementary experiment 3: Impact of data imputation on data projection using Weka 13](#_Toc79150120)

[Introduction 13](#_Toc79150121)

[Data sets 13](#_Toc79150122)

[Methods 13](#_Toc79150123)

[Results 14](#_Toc79150124)

[Glossary 15](#_Toc79150125)

[References 16](#_Toc79150126)

[Supplementary Tables 19](#_Toc79150127)

[Supplementary Figures 23](#_Toc79150128)

## Technical details

### Implementation

The *pguIMP* package is free and open-source software (FOSS). The source code is freely available at GitHub (<https://github.com/SMLMS/pguIMP>) and can be redistributed and/or modified under the terms of the GNU general public license as published by the Free Software Foundation, either version 3 (GPLv3) of the License, or any later version. The programming work was performed in the R language 1 using the R software package 2 (version 4.0.3 for macOS and openSUSE Linux), which is available free of charge in the Comprehensive R Archive Network (CRAN) at [http://CRAN.R-project.org/](http://cran.r-project.org/).

### Installation

The installation from CRAN is done using according to the R standard, i.e., using the command “install.packages(“pguIMP”)”. The installation includes the source code and the methods contained in the *pguIMP* package can be used in own R scripts to create individual, problem-specific workflows or as an interactive dashboard (via the “pguIMP()” command) that follows a specific preprocessing workflow (Figure 1 A in the main article).

### Dependencies

The *pguIMP* package consists of a collection of R6 classes 3 and was created using the “roxygen2” package 4. It’s cross-platform applicability was tested on different operation systems (Windows 10, macOS 10.15.7, openSUSE Leap 15.2 and Ubuntu 20.04.2 LTS). Visual data representation is realized using the “*ggplot2*” package 5. The package imports various functions from other packages, such as the PDE is taken from “*DataVisualizations*” 6, power transformations are realized using the “*rcompanion*” 7 and “*MASS*” 8 packages. Outliers detection is done with the packages “outliers” 9, “dbscan” 10, “e1071” 11 and “OutlierDetection” 12, and for data imputation, the packages “DMwR” 13, “RWeka” 14 and “mice” 15. After installation, the “pguIMP” graphical user interface (GUI) is accessible via the R commands *library(“pguIMP”)* followed by *pguIMP::IMPgui()*. A detailed documentation is provided with the R package and can be currently also be found at <https://user.uni-frankfurt.de/~malkusch/pguHelp.html>.

### Data transformation

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) 16, i.e.

|  |  |
| --- | --- |
| , | (Equation S 1) |

where x is the original variable and x' is the transformed variable. The transformation result can be influenced by choosing the λ-parameter. Appropriate starting points are λ=2 for negative skews and λ=0 for positive skews. The quality of the data transformation procedure is validated by statistical hypothesis tests against the normal distribution, as described in the subsection "statistical hypothesis testing“ below.

### Data normalization

Various data science methods do not perform well when their numerical input has different scales. For example, machine learning models that use the distance between data set instances (e.g., k-nearest neighbors (kNNs) 17, support vector machines (SVMs) 18, density-based spatial clustering of applications with noise (DBSCAN) 19, ordering points to identify the clustering structure (OPTICS)) 20. In addition, some models, such as artificial neural networks, require numerical data within a certain range of values (e.g.: [0;1]) 21. Therefore, data preprocessing occasionally includes scaling of the data with respect to the subsequent data analysis pipeline. The *pguIMP* package offers three common scaling methods, i.e., min-max normalization defined as

|  |  |
| --- | --- |
| , | (Equation S 2) |

where x denotes the original feature, x' is the normalized feature. It scales the feature values to a range between 0 and 1. Further scaling methods include the mean normalization

|  |  |
| --- | --- |
| , | (Equation S 3) |

which shifts the min-max normalization result so that it is centered around the origin. Here, x ̅ is the arithmetic mean of the original feature x. The third scaling method is the z-score normalization, sometimes referred to as z-standardization

|  |  |
| --- | --- |
| , | (Equation S 4) |

Where σ is the standard deviation of the original feature x. Z-score normalization results in a distribution with unit variance that is centered around the origin but does not tie the values to a specific range.

### Machine-learned imputation of missing values by k-nearest neighbors

The imputation of missing values by machine learning is briefly explained at the kNN algorithm as an example. This algorithm estimates the values of a continuous numerical variable based on the position of its nearest neighbors in the data space. Given an *n+1*-dimensional data space and a data set with *m* measurements. Measurement *i* is missing the measurement value of dimension *j*. In what follows, the variable *j* will be referred to as the label. In a reduced n-dimensional data set with the label variable j removed, the Euclidean distances between measurement *i* and all other measurements are calculated. These distances are sorted in ascending order and the k-nearest multivariate neighbors of measurement *i* are determined. The missing value is now determined from variable j in the form of the weighted average of the k-nearest multivariate neighbors determined in the previous step. This is just one example of an imputation method based on machine learning. The value s of k can be set between 2 an 10 neighbors, the default is k = 3 neighbors following tests with the present sample data sets described below.

### Exploratory statistical hypothesis testing

Although the *pguIMP* package is designed as a tool to be used prior to actual data analysis, the availability of some ad hoc statistical tests has proven useful during its development. Some basic statistical tests have been included in the *pguIMP* package to allow ad-hoc testing of observations made during data preprocessing or to validate the data transformation and imputation within the interactive visual environment. Providing a full data analysis tool is not intended as this can be found in the R environment or alternative data analysis software. For normality testing, the Lilliefors’ Kolmogorov-Smirnov test 22, and Anderson-Darling test 23 have been implemented from the package *nortest* 24 and the Shapiro-Wilk test 25 from the package *stats* 2. In addition, to exclude that data imputation introduced group differences not present in the original data, the two-sample Kolmogorov-Smirnov test 26 and Wilcoxon-Mann-Whitney test 27 from the package *stats* 2 are available.

### Documentation

The *pguIMP* package generates a detailed report about all steps of data preprocessing performed on a data set, including the documentation of parameters set for statistical and machine-learning based methods and of the results of statistical hypothesis testing such as the outcome of normality tests. To ensure exact reproducibility of the data preprocessing, all user-selected parameters are documented. These include the selected variables, including their respective filter parameters, the hyper parameters for the transformation according to Tukey's LOP, as well as the variable-specific statistics required for variable scaling (minimum, maximum, mean, standard deviation). If one considers the reproducibility of computer-aided data evaluation, machine learning methods take on a special role as these methods often use randomness to ensure that they actually learn the function to be approximated for the problem. This randomness computationally provided by so-called pseudo-random number generators whose results can be reproduced exactly, if the initial value, the so-called seed, is known. Therefore, the *pguIMP* package ensures the reproducibility of the included machine learning processes by allowing the user to specify the seed for the respective process. For later traceability, these seed values are also documented in the final report.

## Supplementary experiment 1: Comparison of substitution methods for LOQ outliers

### Introduction

The lower limit of quantification as defined by the United States Food and Drug Administration is the limit at which the tolerable accuracy and precision as well increases from a maximum of 15 % to a maximum of 20 %. Therefore, incorporation of values below the lower levels of quantification (LLOQ) lead to a systematic increase in inter-assay variance 28, 29. However, compared to the loss of information caused by stochastically occurring measurement errors, values below the LLOQ thus contain a certain amount of information characterized by an intolerable accuracy and precision of ≤ 20%. 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 LLOQ were discarded or where the critical values were replaced by the variable LLOQ/2 during preprocessing 28. In support of the imputation experiments in the main text, where missing values were artificially introduced into the data set, in this experiment values that lie below the LLOQ are replaced using various methods. The results of the imputed values are then compared with the original measured values. It should be noted that these values are subject to high measurement inaccuracy.

### Data sets

The experiment was performed with data set 1 as described in the main text.

### Methods

Comparison of data set #1 to the LLOQ and the upper limit of quantification (ULOQ) resulted in 7 values in C16Cer that fell below the LLOQ. These values were either substituted with the LLOQ or 0.5 • LLOQ. Subsequently, the PDF of the substituted data was compared with that of the original data by applying two-sample Kolmogorov-Smirnov and Wilcoxon's signed rank tests. The precision of the imputation was quantified as root mean squared percentage error (RMSPE) 30, using the original values as the reference.

### Comparison of substitution methods for LOQ outliers

None of the applied substitution methods significantly altered the C16Cer variable value distribution of data set #1 (Figure S1 and Table S3). However, the deviation of the imputed values from their respective original ones, quantified as RMSPE, indicated that replacing the missing values by 0.5 LLOQ of the respective variable lead to the largest deviation of the imputed values, whereas the smallest deviation was obtained by imputation with the LLOQ.

## Supplementary experiment 2: Assessment of the impact of data imputation on the information loss due to dimension reduction

### Introduction

Dimensionality reduction of high-dimensional data sets is a standard analysis in data science and usually precedes further analyses, such as cluster analysis. Ideally, it is characterized by a high compression rate with low information loss. The compression rate is given by the by the number of dimensions the data set is reduced to. The reconstruction error indicates how much information is lost through dimension reduction. The Reconstruction error therefore represents a suitable metric to demonstrate the impact of different data preprocessing pipelines on subsequent downstream analyses 31. In this complementary experiment, we show the effect of imputing outliers using different methods offered by the *pguIMP* package on information loss during dimensionality reduction.

### Data sets

The experiment was performed with data set 3 as described in the main text.

### Methods

In order to quantify the impact of upstream data preprocessing methods on projecting high-dimensional data onto a low-dimensional plane, two different preprocessing scenarios were performed on a sample data set before dimension reduction. Afterwards the information loss due to dimension reduction was calculated in form of the reconstruction error. For this purpose two sample data sets were created from data set #3 by substituting outliers. First, data were Ln(x) transformed and min-max normalized (Equation S1). Second, imputation sites were introduced by replacing outliers identified with Grubb's tests 32 (α=0.05) with missing values. Third, missing values were imputed either by variable medians or using a kNN model 17 trained on the remaining instances (neighbors = 2, iterations = 10, seed = 42). The transformed and normalized original data served as control. Dimensionality reduction was achieved by means of principal component analysis (PCA) 33 on centered data using the "prcomp" method of the *stats* package 2. The first two principal components were kept since they had eigenvalues > 1 34, 35. The result of the PCA is a n∗2 matrix (XPCA) with n rows (number of instances). To quantify the loss of information due to the dimensionality reduction process, the projected data were transformed back from the two-dimensional space to their original coordinate space and the reconstruction error was quantified using the mean squared error (MSE) as metric 30.

### Results

Five outliers were identified in data set #3 by means of Grubb’s tests. Depending on the imputation method, their substitution directly affected the result of PCA-based dimensionality reduction (Figure 4 of the main article). The cumulative proportion of variance explained by the first two principal components (PCs) increased for both the imputed data sets by 26 %, to a total of 95 %, as compared to the control data set with where PC1 and PC2 explained together 75 % of the total variance. Imputation of the outliers resulted in a reduction of the reconstruction error (Table S4). The quantification of the reconstruction error by calculating the MSE (MSEcontrol = 0.023, MSEmedian = 0.004, MSEkNN = 0.003) shows a reduction of the MSE by a factor of 75 % when using the machine learning-based kNN method instead of the median substitution.

## Supplementary experiment 3: Impact of data imputation on data projection using Weka

### Introduction

In this supplementary experiment, the data preprocessing routine as offered by the “pguIMP” software package presented here will be directly compared to similar solution developed with the Weka tool. For the purpose of comparison the Experiments described in the sections “assess the impact of data preprocessing on subsequent analysis of high-dimensional data structures” and “estimation of consequences of outlier imputation for data set subgroup structures” of the main text were re-performed using the Weka tools for data preprocessing.

### Data sets

The experiment was performed with data set 3 as described in the main text.

### Methods

The data preprocessing pipeline was designed using the Weka (v. 3.8.5.) KnowledgeFlow Environment. The final pipeline (Figure S2) includes the following steps: The data is log-transformed and min-max normalized. Subsequently, outliers are detected using the “InterquartileRange” class (class attributes were defined as extremeValuesFactor = 6.0 and outlierFactor = 1.0) and replaced by missing values. Finally the introduced missis were imputed by the respective attribute’s mean value calculated from the remaining instances using the ReplaceMissingValues object. Further data analysis was performed using the R software package (v. 4.0.3) and is described in the methods sections “assess the impact of data preprocessing on subsequent analysis of high-dimensional data structures” and “estimation of consequences of outlier imputation for data set subgroup structures” of the main text.

### Results

Weka's outlier detection identified exactly the same 5 outliers as “pguIMP”. However, the outlier imputation of Weka is limited to the arithmetic mean value of the respective attribute. As expected, the imputation of the outliers with the respective attribute mean via Weka as well as via “pguIMP” led to exactly the same result: the cumulative proportion of variance explained by the first two PCs was 0.95 % and a reconstruction error of , which is close to the results obtained by median imputation. Training unsupervised clustering methods such as k-Means and DBSCAN with the Weka-imputed (or mean-imputed) data yielded similar results as training with median-imputed data: the k-Means model mislabeled two instances in the Weka-imputed data set, while the DBSCAN model mislabeled two instances and characterized one instance as noise (Figure S3).

## Glossary

CRAN Comprehensive R Archive Network

DBSCAN density-based spatial clustering of applications with noise

FOSS free and open-source software

GPLv3 GNU general public license as published by the Free Software Foundation version 3

GUI graphical user interface

kNN k-nearest neighbors

LLOQ lower limit of quantification

LOP ladder of powers

MSE mean squared error

OPTICS ordering points to identify the clustering structure

PC principal component

PCA principal component analysis

RMSPE root mean squared percentage error

SVM support vector machine

ULOQ upper limit of quantification

## References

1. Ihaka, R.; Gentleman, R., R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics* **1996,** *5* (3), 299-314.

2. R Core Team *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing: Vienna, Austria, 2020.

3. Chang, W. *R6: Encapsulated Classes with Reference Semantics*, 2020.

4. Wickham, H.; Danenberg, P.; Csárdi, G.; Eugster, M. *roxygen2: In-Line Documentation for R*, 2020.

5. Wickham, H., *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York: 2016.

6. Thrun, M. C.; Ultsch, A., Effects of the payout system of income taxes to municipalities in Germany. In *Socio-Economic Modelling and Forecasting*, Papiez; M.; Smiech; S, Eds. Foundation of the Cracow University of Economics: 2018; Vol. No 1, pp 533-542.

7. Mangiafico, S. *rcompanion: Functions to Support Extension Education Program Evaluation*, 2020.

8. Venables, W. N.; Ripley, B. D., *Modern Applied Statistics with S*. Fourth ed.; Springer: New York, 2002.

9. Komsta, L. *outliers: Tests for outliers*, 2011.

10. Hahsler, M.; Piekenbrock, M.; Doran, D., dbscan: Fast Density-Based Clustering with R. *Journal of Statistical Software* **2019,** *91* (1), 1-30.

11. Meyer, D.; Dimitriadou, E.; Hornik, K.; Weingessel, A.; Leisch, F. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2020.

12. Tiwari, V.; Kashikar, A. *OutlierDetection: Outlier Detection*, 2019.

13. Torgo, L., *Data Mining with R, learning with case studies*. Chapman and Hall/CRC.

14. Hornik, K.; Buchta, C.; Zeileis, A., Open-Source Machine Learning: R Meets Weka. *Computational Statistics* **2009,** *24* (2), 225-232.

15. van Buuren, S.; Groothuis-Oudshoorn, K., mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software* **2011,** *45* (3), 1-67.

16. Tukey, J. W.; others, *Exploratory data analysis*. Reading, Mass.: 1977; Vol. 2.

17. Cover, T.; Hart, P., Nearest neighbor pattern classification. *IEEE transactions on information theory* **1967,** *13* (1), 21-27.

18. Cortes, C.; Vapnik, V., Support-Vector Networks. *Machine Learning* **1995,** *20* (3), 273-297.

19. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; others, A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, 1996; Vol. 96, pp 226-231.

20. Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; Sander, J., OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* **1999,** *28* (2), 49-60.

21. Kotsiantis, S. B.; Kanellopoulos, D.; Pintelas, P. E., Data Preprocessing for Supervised Leaning. *Proceedings of World Academy of Science, Engineering and Technology, Vol 12* **2006,** *12* (2), 278-283.

22. Lilliefors, H. W., On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association* **1967,** *62* (318), 399-402.

23. Anderson, T. W.; Darling, D. A., Asymptotic Theory of Certain Goodness of Fit Criteria Based on Stochastic Processes. *Ann Math Stat* **1952,** *23* (2), 193-212.

24. Gross, J.; Ligges, U. *nortest: Tests for Normality*, 2015.

25. Shapiro, S. S.; Wilk, M. B., An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* **1965,** *52*, 591-&.

26. Massey, F. J., The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association* **1951,** *46* (253), 68-78.

27. Mann, H. B.; Whitney, D. R., On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics* **1947,** *18* (1), 50-60, 11.

28. Keizer, R. J.; Jansen, R. S.; Rosing, H.; Thijssen, B.; Beijnen, J. H.; Schellens, J. H.; Huitema, A. D., Incorporation of concentration data below the limit of quantification in population pharmacokinetic analyses. *Pharmacol Res Perspect* **2015,** *3* (2), e00131.

29. United States Food and Drug Administration, Bioanalytical Method Validation Guidance for Industry. Administration, F. a. D., Ed. 2018.

30. Shcherbakov, M. V.; Brebels, A.; Shcherbakova, N. L.; Tyukov, A. P.; Janovsky, T. A.; Kamaev, V. A. e., A survey of forecast error measures. *World Applied Sciences Journal* **2013,** *24* (24), 171-176.

31. Lotsch, J.; Malkusch, S.; Ultsch, A., Optimal distribution-preserving downsampling of large biomedical data sets (opdisDownsampling). *PLoS One* **2021,** *16* (8), e0255838.

32. Grubbs, F. E., Sample Criteria for Testing Outlying Observations. *Ann Math Stat* **1950,** *21* (1), 27-58.

33. Pearson, K., On lines and planes of closest fit to systems of points in space. *Philos Mag* **1901,** *2* (7-12), 559-572.

34. Guttman, L., Some necessary conditions for common-factor analysis. *Psychometrika* **1954,** *19* (2), 149-161.

35. Kaiser, H. F., The varimax criterion for analytic rotation in factor analysis. *Psychometrika* **1958,** *23* (3), 187-200.

36. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V. B., Julia: A fresh approach to numerical computing. *SIAM Review* **2017,** *59* (1), 65-98.

37. Van Rossum, G.; Drake Jr, F. L., *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam: 1995; Vol. 620.

## Supplementary Tables

Table S1: An exemplary overview of available software for data preprocessing in languages commonly used in biomedical data science (such as Julia 36 , Python 37 or R 2)

|  |  |  |  |
| --- | --- | --- | --- |
| Pre-Processing Step | Julia | Python | R |
| Data manipulation | DataFrames.jl55 | Pandas[56,](bookmark://_ENREF_56) [57](bookmark://_ENREF_57) | Tibble[58](bookmark://_ENREF_58) |
| Data visualization | Gadfly.jl59 | Seaborne[60](bookmark://_ENREF_60) | Ggplot2[31](bookmark://_ENREF_31) |
| Data normalization | BoxCoxTrans.jl | Scikit-learn[61,](bookmark://_ENREF_61) [62](bookmark://_ENREF_62) | caret[63](bookmark://_ENREF_63) |
| Data scaling | StatsBase.jl | Scikit-learn[61,](bookmark://_ENREF_61) [62](bookmark://_ENREF_62) | caret[63](bookmark://_ENREF_63) |
| Impute missing values | Impute.jl | Scikit-learn[61,](bookmark://_ENREF_61) [62](bookmark://_ENREF_62) | mice[41](bookmark://_ENREF_41), Amelia II[64](bookmark://_ENREF_64) |
| Detect outliers | OneClassActiveLearning.jl65 | Scikit-learn[61,](bookmark://_ENREF_61) [62](bookmark://_ENREF_62) | outliers[35](bookmark://_ENREF_35) |

Supplemental Figure 1

Table S2: Quality control of the feature distribution after Tukey’s LOP transformation using λ=[2,1,0,−1]. Normality was tested using either the Shapiro-Wilk test (SW), the Kolmogorov-Smirnov test (KS) or the Anderson-Darling test (AD). The given values are the respective p values.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | λ=2 | | | λ=1 | | | λ=0 | | | λ−1 | | |
| Test:  Feature: | SW | KS | AD | SW | KS | AD | SW | KS | AD | SW | KS | AD |
| S1P | 2.04 • 10-04 | 2.86 • 10-01 | 4.99 • 10-03 | 1.80 • 10-01 | 7.37 • 10-01 | 4.34 • 10-01 | 7.29 • 10-01 | 6.49 • 10-01 | 4.73 • 10-01 | 1.66 • 10-02 | 2.07 • 10-01 | 9.50 • 10-03 |
| C16Sphinganin | 2.20 • 10-16 | 6.09 • 10-06 | 3.70 • 10-24 | 3.31 • 10-09 | 8.66 • 10-02 | 1.79 • 10-07 | 8.40 • 10-01 | 8.56 • 10-01 | 8.54 • 10-01 | 4.72 • 10-05 | 2.47 • 10-01 | 1.42 • 10-04 |
| C16Cer | 1.32 • 10-07 | 3.33 • 10-02 | 2.94 • 10-08 | 2.18 • 10-03 | 3.94 • 10-01 | 7.80 • 10-03 | 1.73 • 10-01 | 7.96 • 10-01 | 2.05 • 10-01 | 6.43 • 10-04 | 3.08 • 10-01 | 2.75 • 10-04 |
| C20Cer | 1.34 • 10-10 | 1.85 • 10-03 | 3.27 • 10-12 | 2.34 • 10-04 | 1.65 • 10-01 | 7.56 • 10-04 | 3.13 • 10-01 | 6.08 • 10-01 | 2.46 • 10-01 | 2.56 • 10-06 | 7.99 • 10-02 | 5.35 • 10-07 |
| C24Cer | 2.04 • 10-09 | 2.21 • 10-02 | 2.36 • 10-11 | 1.24 • 10-03 | 2.50 • 10-01 | 2.23 • 10-03 | 2.50 • 10-01 | 8.42 • 10-01 | 2.86 • 10-01 | 6.10 • 10-09 | 7.98 • 10-02 | 2.06 • 10-08 |
| C24\_1Cer | 1.00 • 10-08 | 4.23 • 10-03 | 1.29 • 10-11 | 4.49 • 10-04 | 3.44 • 10-02 | 1.70 • 10-04 | 1.38 • 10-01 | 7.09 • 10-01 | 9.80 • 10-02 | 3.84 • 10-06 | 1.55 • 10-01 | 4.31 • 10-06 |
| C16GluCer | 2.29 • 10-06 | 6.29 • 10-02 | 4.63 • 10-06 | 8.12 • 10-02 | 5.74 • 10-01 | 1.42 • 10-01 | 9.13 • 10-01 | 8.85 • 10-01 | 8.09 • 10-01 | 2.56 • 10-04 | 1.66 • 10-01 | 7.45 • 10-04 |
| C16LacCer | 1.40 • 10-07 | 9.86 • 10-03 | 1.00 • 10-08 | 8.65 • 10-03 | 2.20 • 10-01 | 1.02 • 10-02 | 4.66 • 10-01 | 7.44 • 10-01 | 4.09 • 10-01 | 1.25 • 10-05 | 1.96 • 10-01 | 7.24 • 10-05 |

Table S3: Validation of different imputation methods. The 7 concentrations of the lipid mediator C16Cer that fell below the lower limit of quantification (LLOQ) were each replaced by the LLOQ value, half of the LLOQ value. The root mean squared percentage error (RMSPE) between the 7 originally measured concentrations and the respectively imputed concentrations was determined. Significant deviation of the original distribution of C16Cer concentrations from either of the cleaned distributions of C16Cer concentrations is excluded according to the Kolmogorov-Smirnov test (test statistic: d.Kolmogorov, p value: p.Kolmogorov) and the Wilcoxon-Rank test (test statistic: w.Wilcoxon, p value: p.Wilcoxon) each performed with a significant level of α=0.05.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| imputation method | RMSPE | d.Kolmogorov | p.Kolmogorov | w.Wilcoxon | p.Wilcoxon |
| LLOQ | 1.12 | 0.07 | 0.96 | 4393.5 | 0.95 |
| 0.5 \* LLOQ | 4.51 | 0.07 | 0.96 | 4442.5 | 0.95 |

Table S4 Information content of dimensionality reduced data. The fraction of variance explained by the first two principal components (PCs) is given by PC1 and PC2 and their sum for the associated cumulative fraction. The information loss between the original data and the data reconstructed from the first two PCs is quantified by the mean squared error (MSE).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data set | PC1 | PC2 |  | MSE |
| Control | 5.44 · 10-01 | 2.03 · 10-01 | 7.47 · 10-01 | 2.34 · 10-02 |
| Median imputed | 7.64 · 10-01 | 1.83 · 10-01 | 9.47 · 10-01 | 4.31 · 10-03 |
| kNN imputed | 7.51 · 10-01 | 2.03 · 10-01 | 9.54 · 10-01 | 2.96 · 10-03 |

## Supplementary Figures

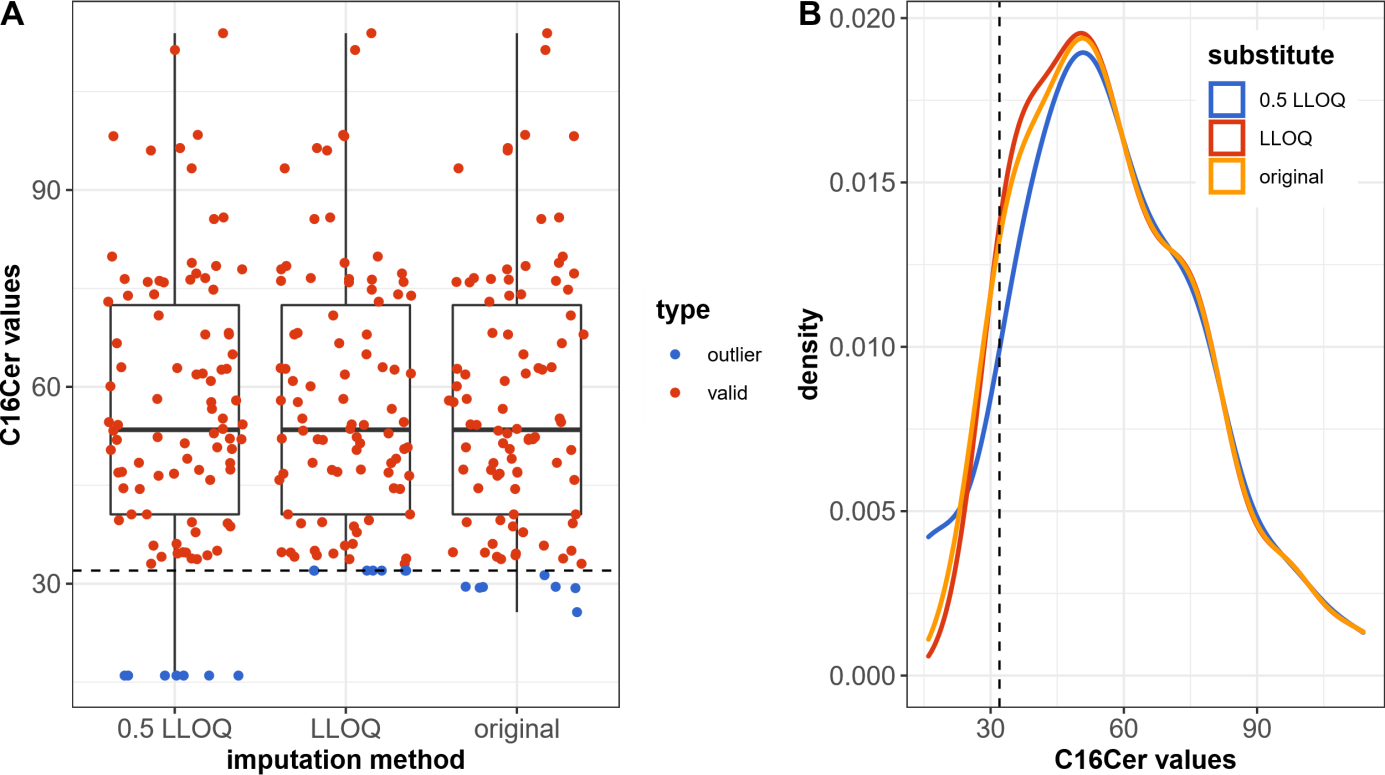


Figure S1: Graphical validation of the performance of multiple imputation methods used to correct plasma C16Cer concentrations falling below lower limit of quantification (LLOQ). A) Box plot of the plasma C16Cer concentration. Individual measurements are depicted as dots that were slightly jittered around their true value to prevent overlapping events. Critical instances that originally fall under LLOQ are highlighted in blue, while non-critical instances are shown in red. The LLOQ is indicated by the horizontal dashed line. B) Probability density function (PDF) of the C16Cer concentration before and after imputation of critical values. The LLOQ is indicated by the vertical dashed line.

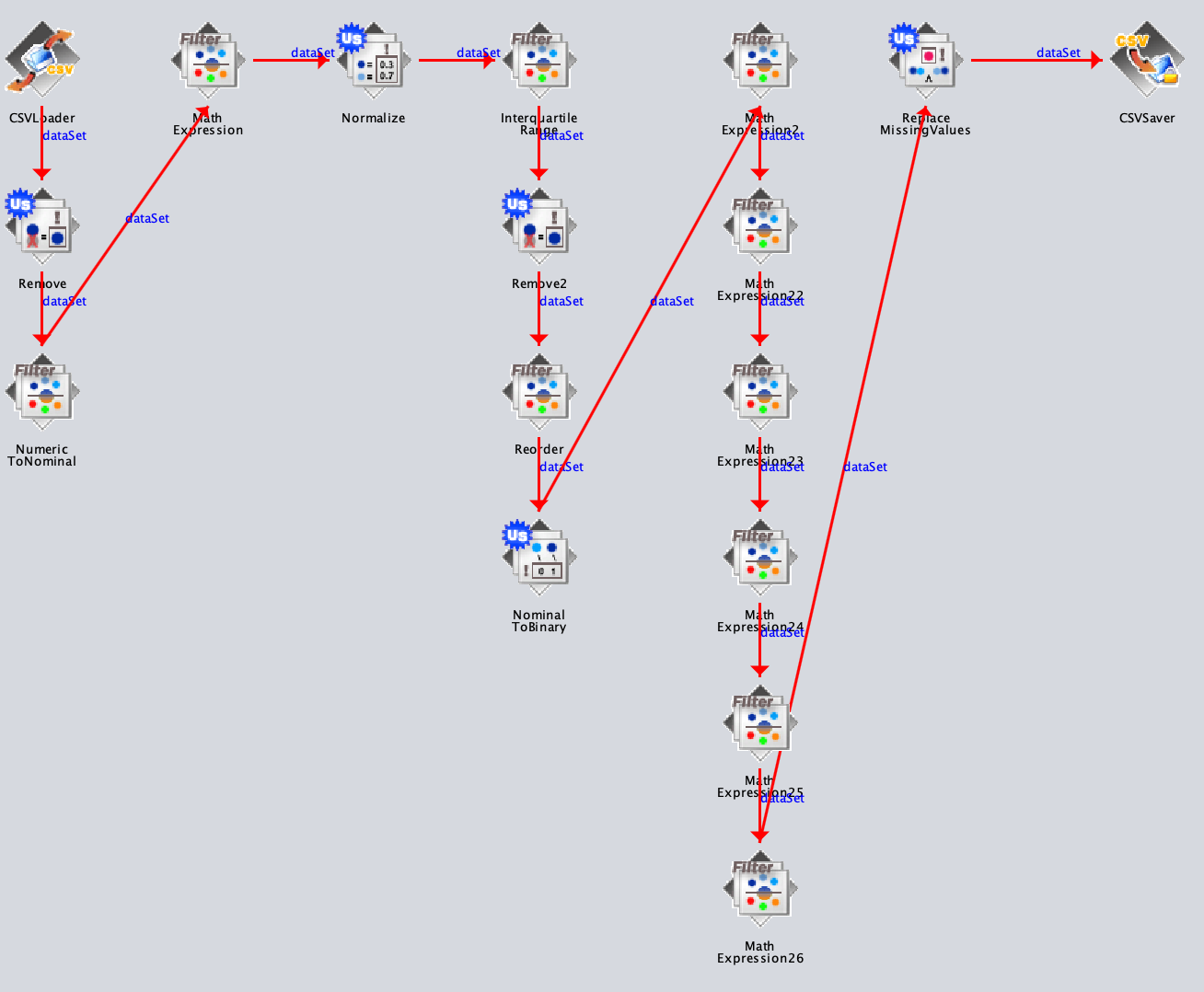


Figure S2: Data preprocessing pipeline designed by the Weka “KnowledgeFlow” environment.



Figure S3: 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 (min-max). Outliers are by Weka’s the “InterquartileRange” class. Unnormal feature values were identified in 5 instances (1,8,10,12,17). Unnormal values in outlier instances were replaced by the respective feature mean. A) Black polygons visualize the cluster separation according to the original labeling of the data set. B Black polygons visualize the cluster separation following k-Means clustering. C) Dendrogram according to OPTICS clustering. D) Reachability plot according to OPTICS ordering. E) Black polygons visualize the cluster separation following DBSCAN clustering as extracted from the OPTICS analysis by applying a distance threshold (dashed line in C and D). 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 feature values, black numbers indicate instances with at least one unnormal feature value.