Package 'zCompositions'

June 23, 2026

Type Package

Title Treatment of Zeros, Left-Censored and Missing Values in Compositional Data Sets

Version 1.6.2

Date 2026-06-20

Maintainer Javier Palarea-Albaladejo <javier.palarea@udg.edu>

Depends R (>= 2.14.0), methods, MASS, NADA, truncnorm

ByteCompile yes

Description Principled methods for the imputation of zeros, left-censored and missing data in compositional data sets (Palarea-Albaladejo and Martin-Fernandez (2015) <doi:10.1016/j.chemolab.2015.02.019>).

License GPL (>= 2)

Repository CRAN

RoxygenNote 7.3.3

Encoding UTF-8

URL https://github.com/Japal/zCompositions

BugReports https://github.com/Japal/zCompositions/issues

NeedsCompilation no

Author Javier Palarea-Albaladejo [cre, aut] (ORCID: <https://orcid.org/0000-0003-0162-669X>), Josep Antoni Martin-Fernandez [aut] (ORCID: <https://orcid.org/0000-0003-2366-1592>)

Date/Publication 2026-06-23 05:10:02 UTC

Contents

`cmultRepl` *Bayesian-Multiplicative replacement of count zeros*

Description

This function implements methods for imputing zeros in compositional count data sets based on a Bayesian-multiplicative replacement.

Usage

```
cmultRepl(X, label = 0,
         method = c("GBM","SQ","BL","CZM","user"), output = c("prop","p-counts"),
             frac = 0.65, threshold = 0.5, adjust = TRUE, t = NULL, s = NULL,
             z.warning = 0.8, z.delete = TRUE, suppress.print = FALSE,
             delta = NULL)
```

Arguments

`X`	Count data set (`matrix` or `data.frame` class).
`label`	Unique label (`numeric` or `character`) used to denote count zeros in `X` (default `label=0`).
`method`	Geometric Bayesian multiplicative (`GBM`, default); square root BM (`SQ`); Bayes-Laplace BM (`BL`); count zero multiplicative (`CZM`); user-specified hyper-parameters (`user`).
`output`	Output format: imputed proportions (`prop`, default) or *pseudo*-counts (`p-counts`).
`frac`	If `method="CZM"`, fraction of the upper threshold used to impute zeros (default `frac=0.65`). Also, fraction of the lowest estimated probability used to adjust imputed proportions falling above it (when `adjust=TRUE`).

threshold	For a vector of counts, factor applied to the quotient 1 over the number of trials (sum of the counts) used to produce an upper limit for replacing zero counts by the CZM method (default threshold=0.5).
adjust	Logical vector setting whether imputed proportions falling above the lowest estimated probability for a multinomial part must be adjusted or not (default adjust=TRUE).
t	If method="user", user-specified *t* hyper-parameter of the Dirichlet prior distribution for each count vector (row) in X. It must be a matrix of the same dimensions as X.
s	If method="user", user-specified *s* hyper-parameter of the Dirichlet prior distribution for each count vector (row) in X. It must be a vector of length equal to the number of rows of X.
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).
suppress.print	Suppress printed feedback (suppress.print=FALSE, default).
delta	This argument has been deprecated and replaced by frac (see package's NEWS for details).

Details

Zero counts, assumed to be due to under-reporting or limited sampling, are imputed under a Bayesian paradigm (GBM, SQ or BL method) by posterior estimates of the multinomial probabilities generating the counts, assuming a Dirichlet prior distribution. The argument method sets the Dirichlet hyper-parameters t (priori estimates of multinomial probabilities) and s (*strength*). The user can specify their own by setting method="user" and entering them as t and s arguments. Note that, under certain circumstances (see references for details), these methods can generate imputed proportions falling above the lowest estimated probability of a multinomial part (c/n, where c is the count and n is the number of trials). In such cases, the imputation is adjusted by using a fraction (frac) of the minimum c/n for that part. Lastly, the non-zero parts are multiplicatively adjusted according to their compositional nature.

On the other hand, method="CZM" uses multiplicative simple replacement (multRepl) on the matrix of estimated probabilities. The upper limit and the fraction used are specified by, respectively, the arguments threshold and frac. Suggested values are threshold=0.5 (so the upper limit for a multinomial probability turns out to be $0.5/n$), and frac=0.65 (so the imputed proportion is 65% of the upper limit).

Value

By default (output="prop") the function returns an imputed data set (data.frame class) in proportions (estimated probabilities). Alternatively, these proportions are re-scaled to produce a compositionally-equivalent matrix of *pseudo*-counts (output="p-counts") which preserves the ratios between parts.

When adjust=TRUE and verbose=TRUE, the number of times, if any, an imputed proportion was adjusted to fall below the minimum estimated multinomial probability is printed.

References

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. Bayesian-multiplicative treatment of count zeros in compositional data sets. Statistical Modelling 2015; 15: 134-158.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

zPatterns

Examples

```
data(Pigs)

# GBM method and matrix of estimated probabilities
Pigs.GBM <- cmultRepl(Pigs)
```

LPdata *La Paloma data set*

Description

96 samples of a 15-part geochemical composition in micrograms/gram from La Paloma stream (Venezuela) including 6.11% values below the limit of detection (coded as 0). For more details see Montero-Serrano et al. (2010).

Usage

```
data(LPdata)
```

Format

A data.frame with 96 observations on the following 15 variables.

`Cr` a numeric vector

`B` a numeric vector

`P` a numeric vector

`V` a numeric vector

`Cu` a numeric vector

`Ti` a numeric vector

`Ni` a numeric vector

`Y` a numeric vector

Sr a numeric vector

La a numeric vector

Ce a numeric vector

Ba a numeric vector

Li a numeric vector

K a numeric vector

Rb a numeric vector

References

Montero-Serrano JC, Palarea-Albaladejo J, Martin-Fernandez JA, and Martinez-Santana M and Gutierrez-Martin JV. Multivariate analysis applied to chemostratigraphic data: identification of chemofacies and stratigraphic correlation, Sedimentary Geology 2010; 228(3-4): 218-228 .

Examples

```
data(LPdata)

zPatterns(LPdata,label=0)
```

LPdataZM *La Paloma data set (incl. zeros and missing data)*

Description

96 samples of a 15-part geochemical composition in micrograms/gram from La Paloma stream (Venezuela). For more details see Montero-Serrano et al. (2010).

Duplicate of the LPdata data set including 2.36% missing at random cells (35.42% samples with missing data; coded as NA) along with 6.11% values below the limit of detection (coded as 0).

Usage

```
data(LPdataZM)
```

Format

A data.frame with 96 observations on the following 15 variables.

Cr a numeric vector

B a numeric vector

P a numeric vector

V a numeric vector

Cu a numeric vector

Ti a numeric vector

Ni a numeric vector

Y a numeric vector

Sr a numeric vector

La a numeric vector

Ce a numeric vector

Ba a numeric vector

Li a numeric vector

K a numeric vector

Rb a numeric vector

References

Montero-Serrano JC, Palarea-Albaladejo J, Martin-Fernandez JA, and Martinez-Santana M and Gutierrez-Martin JV. Multivariate analysis applied to chemostratigraphic data: identification of chemofacies and stratigraphic correlation, Sedimentary Geology 2010; 228(3-4): 218-228 .

See Also

LPdata

Examples

```
data(LPdataZM)

zPatterns(LPdataZM,label=0) # Show zero patterns

zPatterns(LPdataZM,label=NA) # Show missingness patterns
```

---

lrDA *Log-ratio DA algorithm*

Description

This function implements a simulation-based Data Augmentation (DA) algorithm to impute left-censored values (e.g. values below detection limit, rounded zeros) via coordinates representation of compositional data sets which incorporate the information of the relative covariance structure. Alternatively, this function can be used to impute missing data. Multiple imputation estimates can be also obtained from the output.

Usage

```
lrDA(X, label = NULL, dl = NULL,
        ini.cov=c("lrEM","complete.obs","multRepl"), frac = 0.65,
      imp.missing = FALSE, n.iters = 1000, m = 1, store.mi = FALSE, closure = NULL,
        z.warning = 0.8, z.delete = TRUE, delta = NULL)
```

Arguments

X	Compositional data set (matrix or data.frame class).
label	Unique label (numeric or character) used to denote zeros/unobserved values in X.
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X. If NULL the column minima are used as thresholds.
ini.cov	Initial estimation of the log-ratio covariance matrix. It can be based on lrEM estimation ("lrEM", default), complete observations ("complete.obs") or multiplicative simple replacement ("multRepl").
frac	If ini.cov="multRepl", parameter for initial multiplicative simple replacement (multRepl) (default = 0.65).
imp.missing	If TRUE then unobserved data identified by label are treated as missing data (default = FALSE).
n.iters	Number of iterations for the DA algorithm (default = 1000).
m	Number of multiple imputations (default = 1).
store.mi	Logical value. If m>1 creates a list with m imputed data matrices. (store.mi=FALSE, default).
closure	Closure value used to add a residual part if needed when multiplicative simple replacement is used to initiate the DA algorithm, either directly (ini.cov="multRepl") or as part of lrEM estimation (ini.cov="lrEM") (see ?multRepl).
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).
delta	This argument has been deprecated and replaced by frac (see package's NEWS for details).

Details

After convergence of the Markov chain Monte Carlo (MCMC) iterative process to its steady state, this function imputes unobserved compositional parts by simulated values from their posterior predictive distributions through coordinates representation, given the information from the observed data. For left-censoring problems, it allows for either single (vector form) or multiple (matrix form, same size as X) limits of detection by component. In dl, any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces imputed data sets on the same scale as the input data set. If X is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

The common conjugate normal inverted-Wishart distribution with non-informative Jeffreys prior has been assumed for the model parameters in the coordinates space. Under this setting, convergence is expected to be fast (n.iters set to 1000 by default). Besides, considering EM parameter estimates as initial point for the DA algorithm (ini.cov="lrEM") assures faster convergence by

starting near the centre of the posterior distribution. Note that the estimation of the covariance matrix requires a regular data set, i.e. having more observations than variables in the data.

By setting `m` greater than 1, the procedure also allows for multiple imputations of the censored values drawn at regular intervals after convergence. In this case, in addition to the burn-in period for convergence, `n.iters` determines the gap, large enough to prevent from correlated values, between successive imputations. The total number of iterations is then `n.iters*m`. By default, a single imputed data set results from averaging the `m` imputations in the space of coordinates. If `store.mi=TRUE`, a list with `m` imputed data sets is generated instead.

In the case of censoring patterns involving samples containing only one observed component, these are imputed by multiplicative simple replacement (`multRepl`) and a warning message identifying them is printed.

Missing data imputation

This function can be employed to impute missing data by setting `imp.missing = TRUE`. For this case, the argument `label` indicates the unique label for missing values. The argument `dl` is ignored as it is meaningless here.

Value

A `data.frame` object containing the imputed compositional data set expressed in the original scale, or a `list` of imputed data sets if multiple imputation is carried out (`m>1`) and `store.mi=TRUE`.

References

Palarea-Albaladejo J, Martin-Fernandez JA, Olea, RA. A bootstrap estimation scheme for chemical compositional data with nondetects. Journal of Chemometrics 2014; 28: 585-599.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

`zPatterns`, `lrEM`, `lrSVD`, `multRepl`, `multLN`, `multKM`, `cmultRepl`

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
              38.09,7.62,23.68,9.70,20.91,0.00,
              27.67,7.15,13.05,32.04,6.54,13.55,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

# Imputation by single simulated values
X_lrDA <- lrDA(X,label=0,dl=rep(1,6),ini.cov="multRepl",n.iters=150)
```

```
# Imputation by multiple imputation (m = 5, one imputation every 150 iterations)
X_milrDA <- lrDA(X,label=0,dl=rep(1,6),ini.cov="multRepl",m=5,n.iters=150)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrDA2 <- lrDA(X,label=0,dl=mdl,ini.cov="multRepl",n.iters=150)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La))  # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

## Not run:  # May take a little while
LPdata2_lrDA <- lrDA(LPdata2,label=0,dl=dl2)
## End(Not run)

## Not run:  # May take a little while
# Treating zeros as missing data for illustration purposes only
LPdata2_lrDAmiss <- lrDA(LPdata2,label=0,imp.missing=TRUE,closure=10^6)
## End(Not run)
```

---

lrEM *Log-ratio EM algorithm*

Description

This function implements model-based ordinary and robust Expectation-Maximisation algorithms to impute left-censored data (e.g. values below detection limit, rounded zeros) via coordinates representation of compositional data which incorporate the information of the relative covariance structure.

This function can be also used to impute missing data instead by setting `imp.missing = TRUE` (see `lrEMplus` to treat censored and missing data simultaneously).

Usage

```
lrEM(X, label = NULL, dl = NULL, rob = FALSE,
        ini.cov = c("complete.obs", "multRepl"), frac = 0.65,
        tolerance = 0.0001, max.iter = 50, rlm.maxit = 150,
        imp.missing = FALSE, suppress.print = FALSE, closure = NULL,
        z.warning = 0.8, z.delete = TRUE, delta = NULL)
```

Arguments

X	Compositional data set (matrix or data.frame class).
label	Unique label (numeric or character) used to denote zeros/unobserved values in X.
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X. If NULL the column minima are used as thresholds.
rob	Logical value. FALSE provides maximum-likelihood estimates of model parameters (default), TRUE provides robust parameter estimates.
ini.cov	Initial estimation of either the log-ratio covariance matrix (ML estimation) or unobserved data (robust estimation). It can be based on either complete observations ("complete.obs", default) or multiplicative simple replacement ("multRepl").
frac	If ini.cov="multRepl", parameter for initial multiplicative simple replacement of left-censored data (see multRepl) (default = 0.65).
tolerance	Convergence criterion for the EM algorithm (default = 0.0001).
max.iter	Maximum number of iterations for the EM algorithm (default = 50).
rlm.maxit	If rob=TRUE, maximum number of iterations for the embedded robust regression estimation (default = 150; see rlm for details).
imp.missing	If TRUE then unobserved data identified by label are treated as missing data (default = FALSE).
suppress.print	Suppress printed feedback (suppress.print = FALSE, default).
closure	Closure value used to add a residual part if needed when ini.cov="multRepl" is used (see ?multRepl).
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).
delta	This argument has been deprecated and replaced by frac (see package's NEWS for details).

Details

After convergence, this function imputes unobserved compositional data by their estimated conditional expected values through coordinates representation, given the information from the observed data. For left-censoring problems, it allows for either single (vector form) or multiple (matrix form, same size as X) limits of detection by component. In dl, any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If X is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

Under maximum likelihood (ML) estimation (default, rob=FALSE), a correction factor based on the residual covariance obtained by censored regression is applied for the correct estimation of the

conditional covariance matrix in the maximisation step of the EM algorithm. This is required in order to obtain the conditional expectation of the sum of cross-products between two components in the case that both involve imputed values. Note that the procedure is based on the oblique additive log-ratio (alr) transformation to simplify calculations and alleviates computational burden. Nonetheless, the same results would be obtained using an isometric log-ratio transformation (ilr). Note also that alr requires at least one complete column. Otherwise, a preliminary imputation, e.g. by `multRepl` or `multLN`, of the most simplest censoring pattern may be enough. The argument `ini.cov` determines how the initial estimation of the log-ratio covariance matrix required to start the EM process is worked out. Note that the estimation of the covariance matrix, and hence the lrEM routine, requires a regular data set, i.e. having more observations than variables in the data.

Under robust estimation (`rob=TRUE`), the algorithm requires ilr transformations in order to satisfy requirements for robust estimation methods (MM-estimation by default, see `rlm` function for more details). An initial estimation of nondetects is required to get the algorithm started. This can be based on either the subset of fully observed cases (`ini.cov="complete.obs"`) or a multiplicative simple replacement of all nondetects in the data set (`ini.cov="multRepl"`). Note that the robust regression method involved includes random elements which can, occasionally, give rise to `NaN` values getting the routine execution halted. If this happened, we suggest to simply re-run the function once again.

Note that conditional imputation based on log-ratio coordinates cannot be conducted when there exist censoring patterns including samples with only one observed component. As a workaround, `lrEM` applies multiplicative simple replacement (`multRepl`) on those and a warning message identifying the problematic cases is printed. Alternatively, it might be sensible to simply remove those non-informative samples from the data set.

Missing data imputation

When `imp.missing = TRUE`, unobserved values are treated as general missing data and imputed by their conditional expectation using the EM algorithm. Either maximum-likelihood or robust estimation can be used through the `rob` argument. For this case, the argument `label` indicates the unique label for missing values. The algorithm can be initiated using either `"complete.obs"` or `"multRepl"` (for missing data) as specified by the `ini.cov` argument. The argument `dl` is ignored.

Value

A `data.frame` object containing the imputed compositional data set expressed in the original scale. The number of iterations required for convergence is also printed (this can be suppressed by setting `suppress.print=TRUE`).

References

Martin-Fernandez J.A., Hron K., Templ M., Filzmoser P., Palarea-Albaladejo J. Model-based replacement of rounded zeros in compositional data: classical and robust approaches. Computational Statistics & Data Analysis 2012; 56: 2688-2704.

Palarea-Albaladejo J, Martin-Fernandez JA, Gomez-Garcia J. A parametric approach for dealing with compositional rounded zeros. Mathematical Geology 2007; 39: 625-45.

Palarea-Albaladejo J, Martin-Fernandez JA. A modified EM alr-algorithm for replacing rounded zeros in compositional data sets. Computers & Geosciences 2008; 34: 902-917.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. Analytica Chimica Acta 2013; 764: 32-43.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

zPatterns, lrSVD, lrDA, multRepl, multLN, multKM, cmultRepl

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
              38.09,7.62,23.68,9.70,20.91,0.00,
              27.67,7.15,13.05,32.04,6.54,13.55,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_lrEM <- lrEM(X,label=0,dl=rep(1,6),ini.cov="multRepl")
X_roblrEM <- lrEM(X,label=0,dl=rep(1,6),ini.cov="multRepl",rob=TRUE,tolerance=0.001)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrEM2 <- lrEM(X,label=0,dl=mdl,ini.cov="multRepl")

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La))  # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

LPdata2_lrEM <- lrEM(LPdata2,label=0,dl=dl2)
LPdata2_roblrEM <- lrEM(LPdata2,label=0,dl=dl2,rob=TRUE,tolerance=0.005)

# Two subsets of limits of detection (using e.g. robust parameter estimation)
 # Using a subset of LPdata for faster execution
data(LPdata) # data (ppm/micrograms per gram)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La))
dl2 <- c(2,1,0,0,0,1,0.6,1,0,0,632,10)
 # DLs for first 50 samples of LPdata2
dl2a <- matrix(rep(1,50),ncol=1)%*%dl2
 # DLs for last 46 samples of LPdata
```

```
dl2b <- matrix(rep(1,46),ncol=1)%*%c(1,0.5,0,0,0,0.75,0.3,1,0,0,600,8)

mdl <- rbind(dl2a,dl2b)
LPdata2_roblrEM <- lrEM(LPdata2,label=0,dl=mdl,rob=TRUE,tolerance=0.005)

# Treating zeros as general missing data for illustration purposes only
LPdata2_miss <- lrEM(LPdata2,label=0,imp.missing=TRUE)
```

`lrEMplus` *Log-ratio EM algorithm (plus)*

Description

This function implements an extended version of the log-ratio EM algorithm (`lrEM` function) to simultaneously deal with both zeros (i.e. data below detection limit, rounded zeros) and missing data in compositional data sets.

Note: zeros and missing data must be labelled using 0 and `NA` respectively to use this function.

Usage

```
lrEMplus(X, dl = NULL, rob = FALSE,
            ini.cov = c("complete.obs", "multRepl"), frac = 0.65,
            tolerance = 0.0001, max.iter = 50, rlm.maxit = 150,
            suppress.print = FALSE, closure = NULL,
            z.warning = 0.8, z.delete = TRUE, delta = NULL)
```

Arguments

`X`	Compositional data set (`matrix` or `data.frame` class).
`dl`	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as `X`. If `NULL` the column minima are used as thresholds.
`rob`	Logical value. `FALSE` provides maximum-likelihood estimates of model parameters (default), `TRUE` provides robust parameter estimates.
`ini.cov`	Initial estimation of either the log-ratio covariance matrix (ML estimation) or unobserved data (robust estimation). It can be based on either complete observations (`"complete.obs"`, default) or multiplicative simple replacement (`"multRepl"`).
`frac`	If `ini.cov="multRepl"`, parameter for initial multiplicative simple replacement of left-censored data (see `multRepl`) (default = 0.65).
`tolerance`	Convergence criterion (default = 0.0001).
`max.iter`	Maximum number of iterations (default = 50).
`rlm.maxit`	If `rob=TRUE`, maximum number of iterations for the embedded robust regression estimation (default = 150; see `rlm` for details).
`suppress.print`	Suppress printed feedback (`suppress.print = FALSE`, default).
`closure`	Closure value used to add a residual part if needed when `ini.cov="multRepl"` is used (see `?multRepl`).

z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).
delta	This argument has been deprecated and replaced by frac (see package's NEWS for details).

Details

The procedure starts with an initial imputation of either zeros (using simple replacement with frac*dl) or missing values (using geometric mean imputation from observed data) depending of which problem is the least frequent in the data set. Subsequently, iterative calls to lrEM replace zeros and missing data alternately until convergence to a stable solution or the maximum number of iterations is reached.

See ?lrEM for more details.

Value

A data.frame object containing the imputed compositional data set in the same scale as the original. The number of iterations required for convergence is also printed (this can be suppressed by setting suppress.print=TRUE).

References

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J. Model-based replacement of rounded zeros in compositional data: classical and robust approaches. Computational Statistics & Data Analysis 2012; 56: 2688-2704.

Palarea-Albaladejo J, Martin-Fernandez JA, Gomez-Garcia J. A parametric approach for dealing with compositional rounded zeros. Mathematical Geology 2007; 39: 625-45.

Palarea-Albaladejo J, Martin-Fernandez JA. A modified EM alr-algorithm for replacing rounded zeros in compositional data sets. Computers & Geosciences 2008; 34: 902-917.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. Analytica Chimica Acta 2013; 764: 32-43.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

lrEM

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
# (Note that zeros and missing in the same row or column are allowed)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,41.42,0.00,NA,6.80,12.05,
```

```
                NA,35.13,7.96,14.28,35.12,7.51,
                10.85,46.40,31.89,10.86,0.00,0.00,
                10.85,16.27,NA,9.16,19.57,44.15,
                38.09,7.62,23.68,9.70,20.91,0.00,
                NA,9.89,18.04,44.30,9.04,18.73,
                44.41,15.04,7.95,0.00,10.82,21.78,
                11.50,30.33,6.85,13.92,30.82,6.58,
                19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_lrEMplus <- lrEMplus(X,dl=rep(1,6),ini.cov="multRepl")
X_roblrEMplus <- lrEMplus(X,dl=rep(1,6),ini.cov="multRepl",rob=TRUE,max.iter=4)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrEMplus2 <- lrEMplus(X,dl=mdl,ini.cov="multRepl")

# Non-closed compositional data set
data(LPdataZM) # (in ppm; 0 is nondetect and NA is missing data)

dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdataZM2 <- subset(LPdataZM,select=-c(Cu,Ni,La)) # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

LPdataZM2_lrEMplus <- lrEMplus(LPdataZM2,dl=dl2)
```

`lrSVD` *Log-ratio SVD algorithm*

Description

This function implements an iterative algorithm to impute left-censored data (e.g. values below detection limit, rounded zeros) based on the singular value decomposition (SVD) of a compositional data set. It is particularly indicated for the case in which the data contain more variables than observations.

This function can be also used to impute missing data instead by setting `imp.missing = TRUE` (see `lrSVDplus` to treat censored and missing data simultaneously).

Usage

```
lrSVD(X, label = NULL, dl = NULL, frac = 0.65, ncp = 2,
         imp.missing=FALSE, beta = 0.5, method = c("ridge", "EM"),
         row.w = NULL, coeff.ridge = 1, threshold = 1e-04, seed = NULL,
         nb.init = 1, max.iter = 1000, z.warning = 0.8, z.delete = TRUE,
         ...)
```

Arguments

X	Compositional data set (matrix or data.frame class).
label	Unique label (numeric or character) used to denote zeros/unobserved values in X.
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X. If NULL the column minima are used as thresholds.
frac	Parameter for initial multiplicative simple replacement of left-censored data (see multRepl) (default = 0.65).
ncp	Number of components for low-rank matrix approximation (default = 2).
imp.missing	If TRUE then unobserved data identified by label are treated as missing data (default = FALSE).
beta	Weighting parameter, balance between the two conditions in objective function (default = 0.5).
method	Parameter estimation method for the iterative algorithm (method = "ridge", default).
row.w	row weights (default = NULL, a vector of 1 for uniform row weights).
coeff.ridge	Used when method = "ridge" (default = 1).
threshold	Threshold for assessing convergence (default = 1e-04).
seed	Seed for random initialisation of the algorithm (default seed = NULL, unobserved values initially imputed by the column mean).
nb.init	Number of random initialisations (default = 1).
max.iter	Maximum number of iterations for the algorithm (default = 1000).
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).
...	Further arguments.

Details

This function implements an efficient imputation algorithm particularly suitable for the case of continuous high-dimensional (wide) compositional data sets (more columns than rows), although it is equally applicable to regular data sets. It is based on a low-rank representation of the data set by a principal components (PC) model as derived by singular value decomposition (SVD) of the data matrix, extending recent work on principal component imputation and matrix completion methods to the case of censored compositional data (the code builds on the function imputePCA; see missMDA package for more details). A preliminary imputation by multiplicative replacement (see multRepl) is conducted to initiate the iterative algorithm in log-ratio coordinates. Two steps, estimation of latent PC model loadings and imputation of empty data matrix cells using the model, are iteratively repeated until convergence. Parameter fitting in this context is performed by a regularisation method (ridge regression in this case) or by the expectation-maximisation (EM) algorithm. Regularization

has been shown generally preferable and it is set as default method (note the regularisation parameter `coeff.ridge` set to 1 by default. If it is < 1 the result is closer to EM estimation, whereas for values > 1 it is closer to mean estimation).

An imputed data set is produced on the same scale as the input data set. If `X` is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

In `dl`, any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

Missing data imputation

When `imp.missing = TRUE`, unobserved values are treated as general missing data. For this case, the argument `label` indicates the unique label for missing values and the argument `dl` is ignored.

Value

A `data.frame` object containing the imputed compositional data set expressed in the original scale.

References

Palarea-Albaladejo, J, Antoni Martín-Fernández, JA, Ruiz-Gazen, A, Thomas-Agnan, C. lrSVD: An efficient imputation algorithm for incomplete high-throughput compositional data. Journal of Chemometrics 2022; 36: e3459.

See Also

`zPatterns`, `lrSVD`, `lrDA`, `multRepl`, `multLN`, `multKM`, `cmultRepl`, `lrSVDplus`

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
              38.09,7.62,23.68,9.70,20.91,0.00,
              27.67,7.15,13.05,32.04,6.54,13.55,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_lrSVD <- lrSVD(X,label=0,dl=rep(1,6))

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)
```

```
X_lrSVD2 <- lrSVD(X,label=0,dl=mdl)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata2 <- subset(LPdata,select=-c(Cu,Ni,La))  # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

LPdata2_lrSVD <- lrSVD(LPdata2,label=0,dl=dl2)

# Treating zeros as general missing data for illustration purposes only
LPdata2_miss <- lrSVD(LPdata2,label=0,imp.missing=TRUE)
```

`lrSVDplus` *Log-ratio SVD algorithm (plus)*

Description

This function implements an extended version of the log-ratio SVD algorithm (`lrSVD` function) to simultaneously deal with both zeros (i.e. data below detection limit, rounded zeros) and missing data in compositional data sets.

Note: zeros and missing data must be labelled using 0 and `NA` respectively to use this function.

Usage

```
lrSVDplus(X, dl = NULL, frac = 0.65,
             ncp = 2, beta = 0.5, method = c("ridge", "EM"), row.w = NULL,
             coeff.ridge = 1, threshold = 1e-04, seed = NULL, nb.init = 1,
             max.iter = 1000, z.warning = 0.8, z.delete = TRUE,
             ...)
```

Arguments

`X`	Compositional data set (`matrix` or `data.frame` class).
`dl`	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as `X`. If `NULL` the column minima are used as thresholds.
`frac`	Parameter for initial multiplicative simple replacement of left-censored data (see `multRepl`) (default = 0.65).
`ncp`	Number of components in low-rank matrix approximation (default = 2).
`beta`	Weighting parameter, balance between the two conditions in objective function (default = 0.5).
`method`	Parameter estimation method for the iterative algorithm (`method = "ridge"`, default).
`row.w`	row weights (default = NULL, a vector of 1 for uniform row weights).
`coeff.ridge`	Used when `method = "ridge"` (default = 1).

threshold	Threshold for assessing convergence (default = 1e-04).
seed	Seed for random initialisation of the algorithm (default seed = NULL, unobserved values initially imputed by the column mean).
nb.init	Number of random initialisations (default = 1).
max.iter	Maximum number of iterations for the algorithm (default = 1000).
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).
...	Further arguments.

Details

The procedure starts with an initial imputation of zeros (using simple replacement with frac*dl) and missing values (using geometric mean imputation from observed data). Subsequently, the iterative algorithm is run until convergence (see ?lrSVD for more details).

Value

A data.frame object containing the imputed compositional data set expressed in the original scale.

References

Palarea-Albaladejo, J, Antoni Martín-Fernández, J, Ruiz-Gazen, A, Thomas-Agnan, C. lrSVD: An efficient imputation algorithm for incomplete high-throughput compositional data. Journal of Chemometrics 2022; 36: e3459.

See Also

zPatterns, lrSVD, lrDA, multRepl, multLN, multKM, cmultRepl, lrSVD

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
# (Note that zeros and missing in the same row or column are allowed)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,41.42,0.00,NA,6.80,12.05,
              NA,35.13,7.96,14.28,35.12,7.51,
              10.85,46.40,31.89,10.86,0.00,0.00,
              10.85,16.27,NA,9.16,19.57,44.15,
              38.09,7.62,23.68,9.70,20.91,0.00,
              NA,9.89,18.04,44.30,9.04,18.73,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_lrSVDplus <- lrSVDplus(X,dl=rep(1,6))
```

```
# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_lrSVDplus2 <- lrSVDplus(X,dl=mdl)

# Non-closed compositional data set
data(LPdataZM) # (in ppm; 0 is nondetect and NA is missing data)

dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdataZM2 <- subset(LPdataZM,select=-c(Cu,Ni,La)) # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

LPdataZM2_lrSVDplus <- lrSVDplus(LPdataZM2,dl=dl2)
```

`mdl` *Water data set: matrix of limits of detection*

Description

Matrix of varying limits of detection for the `Water` data set.

Usage

```
data(mdl)
```

Format

A `matrix` with 100 rows and 4 columns.

Details

Three limits of detection (0.75, 1 and 1.25) were considered for Potassium, four for Arsenic (1.5, 3, 4 and 5), two for Sulphate (29 and 35) and no one for Calcium.

Examples

```
data(Water)
data(mdl)
```

multKM *Multiplicative Kaplan-Meier smoothing spline (KMSS) replacement*

Description

This function implements non-parametric multiplicative KMSS imputation of left-censored values (e.g. values below detection limit, rounded zeros) in compositional data sets. It is based on simulation from a smoothing spline fitted to the Kaplan-Meier (KM) estimate of the empirical cumulative distribution function (ECDF) of the data.

Usage

```
multKM(X, label = NULL, dl = NULL, n.draws = 1000, n.knots = NULL,
          z.warning = 0.8, z.delete = TRUE)
```

Arguments

X	Compositional data set (matrix or data.frame class).
label	Unique label (numeric or character) used to denote zeros/unobserved left-censored values in X.
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X. If NULL the column minima are used as thresholds.
n.draws	Number of random draws from the inverse KM ECDF generated to produce an averaged imputed value (n.draws=1000, default).
n.knots	Integer or function giving the number of knots used for fitting a cubic smoothing spline to the KM ECDF (see smooth.spline for default value). It allows for a vector or list of settings per column of X.
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).

Details

This function imputes left-censored compositional values by averaging (geometric mean) n random draws (n.draws argument) from a cubic smoothing spline curve fitting the inverse KM ECDF below the corresponding limit of detection or censoring threshold. It then applies a multiplicative adjustment to preserve the multivariate compositional properties of the samples. It allows for either single (vector form) or multiple (matrix form, same size as X) limits of detection by component. Although note that it is equivalent to simple substitution by the limit of detection for singly censored components. In dl, any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If X is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered.

The level of smoothing of the estimated spline can be controlled by the n.knots argument. The function splineKM can assist in choosing a finer value, although the default setting works generally well.

Value

A data.frame object containing the imputed compositional data set expressed in the original scale.

References

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligent Laboratory Systems 2015; 143: 85-96.

See Also

zPatterns, splineKM, lrEM, lrSVD, lrDA, multRepl, multLN, cmultRepl

Examples

```
data(Water)
data(mdl) # matrix of limits of detection for Water

Water_multKM <- multKM(Water,label=0,dl=mdl)

# Different smoothing degree by component
Water_multKM2 <- multKM(Water,label=0,dl=mdl,n.knots=c(25,50,30,75))

# Easy to use for KM multiple imputation (m = 10)
Water.mi <- vector("list",length=10)
for (m in 1:10){
  Water.mi[[m]] <- multKM(Water,label=0,dl=mdl,n.draws=1)
}
```

---

multLN *Multiplicative lognormal replacement*

Description

This function implements model-based multiplicative lognormal imputation of left-censored values (e.g. values below detection limit, rounded zeros) in compositional data sets.

Usage

```
multLN(X, label = NULL, dl = NULL, rob = FALSE, random = FALSE,
          z.warning = 0.8, z.delete = TRUE)
```

Arguments

X	Compositional data set (matrix or data.frame class).
label	Unique label (numeric or character) used to denote zeros/unobserved left-censored values in X.
dl	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as X. If NULL the column minima are used as thresholds.
rob	Logical value. FALSE provides maximum-likelihood estimates of model parameters (default), TRUE provides robust estimates (see NADA package for details).
random	Logical value. Values imputed using either estimated geometric mean (FALSE, default) or random values (TRUE) below the limit of detection.
z.warning	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default z.warning=0.8).
z.delete	Logical value. If set to TRUE, rows/columns identified by z.warning are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by z.warning (default z.delete=TRUE).

Details

By default, this function imputes left-censored compositional values by the estimated geometric mean of the values below the corresponding limit of detection or censoring threshold and applies a multiplicative adjustment to preserve the multivariate compositional properties of the samples. Alternatively, imputation can be carried out by random values below the limit of detection (random = TRUE) based on a normal distribution on the positive real line (see below).

It depends on package NADA to produce the required model parameter estimates (either maximum likelihood or robust regression on order statistics). It allows for either single (vector form) or multiple (matrix form, same size as X) limits of detection by component. In dl, any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

It produces an imputed data set on the same scale as the input data set. If X is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered. Note that a normal distribution on the positive real line is considered. That is, it is defined with respect to a measure according to own geometry of the positive real line, instead of the standard lognormal based on the Lebesgue measure in real space.

Value

A data.frame object containing the imputed compositional data set expressed in the original scale.

References

Mateu-Figueras G, Pawlowsky-Glahn V, Egozcue JJ. The normal distribution in some constrained sample spaces. SORT 2013; 37: 29-56.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. Analytica Chimica Acta 2013; 764: 32-43.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

zPatterns, lrEM, lrSVD, lrDA, multRepl, multKM, cmultRepl

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
              38.09,7.62,23.68,9.70,20.91,0.00,
              27.67,7.15,13.05,32.04,6.54,13.55,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_multLN <- multLN(X,label=0,dl=rep(1,6))

# Using ROS for parameter estimation
X_multLNrob <- multLN(X,label=0,dl=rep(1,6),rob=TRUE)

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_multLN2 <- multLN(X,label=0,dl=mdl)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)

# Using ML for parameter estimation
LPdata_multLN <- multLN(LPdata,label=0,dl=dl)
# For comparison
LPdata[30:35,1:10]
round(LPdata_multLN[30:35,1:10],1)

# Using random values < dl
LPdata_multRLN <- multLN(LPdata,label=0,dl=dl,random=TRUE)
round(LPdata_multRLN[30:35,1:10],1)

# Two subsets of limits of detection (using e.g. ML parameter estimation)
```

```
data(LPdata)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
 # DLs for first 50 samples of LPdata
dl1 <- matrix(rep(1,50),ncol=1)%*%dl
 # DLs for last 46 samples of LPdata
dl2 <- matrix(rep(1,46),ncol=1)%*%c(1,0.5,0,0,2.5,0,5.5,0.75,0.3,1.5,1,0,0,600,8)

mdl <- rbind(dl1,dl2)
LPdata_multLN2 <- multLN(LPdata,label=0,dl=mdl)
```

`multRepl` *Multiplicative simple replacement*

Description

This function implements non-parametric multiplicative simple imputation of left-censored (e.g. values below detection limit, rounded zeros) and missing data in compositional data sets.

Usage

```
multRepl(X, label = NULL, dl = NULL, frac = 0.65, imp.missing = FALSE, closure = NULL,
            z.warning=0.8, z.delete = TRUE, delta = NULL)
```

Arguments

`X`	Compositional vector (`numeric` class) or data set (`matrix` or `data.frame` class).
`label`	Unique label (`numeric` or `character`) used to denote zeros/unobserved values in `X`.
`dl`	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as `X`. If `NULL` the column minima are used as thresholds.
`frac`	Fraction of the detection limit/threshold used for imputation (default = 0.65, expressed as a proportion).
`imp.missing`	If `TRUE` then unobserved values identified by `label` are treated as missing data (default = `FALSE`).
`closure`	Closure value used to add a residual part for imputation (see below).
`z.warning`	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default `z.warning=0.8`).
`z.delete`	Logical value. If set to `TRUE`, rows/columns identified by `z.warning` are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by `z.warning` (default `z.delete=TRUE`).
`delta`	This argument has been deprecated and replaced by `frac` (see package's NEWS for details).

Details

This function imputes left-censored compositional values by a given fraction `frac` of the corresponding limit of detection and applies a multiplicative adjustment to preserve the multivariate compositional properties of the samples. It allows for either single (`vector` form) or multiple (`matrix` form, same size as `X`) limits of detection by component. In `dl`, any threshold value can be set for non-censored elements (e.g. use 0 if no threshold for a particular column or element of the data matrix).

Missing data imputation: missing data can be imputed by setting `imp.missing = TRUE`. They are replaced by the estimated column geometric mean from observed values. The non-missing parts in the composition are applied multiplicative adjustment. The argument `dl` and `frac` are ignored and `X` is require to be a data matrix in this case.

Note: negative values can be generated when unobserved components are a large portion of the composition, which is more likely for missing data (e.g in major chemical elements) and non-closed compositions. A workaround is to add a residual filling the gap up to the closure/total when possible. This is done internally when a value for `closure` is specified (e.g. `closure=10^6` if ppm or `closure=100` if percentages). The residual is discarded after imputation.

This function produces an imputed data set on the same scale as the input data set. If `X` is not closed to a constant sum, then the results are adjusted to provide a compositionally equivalent data set, expressed in the original scale, which leaves the absolute values of the observed components unaltered. Note that this adjustment only applies to data sets and not when a single composition is entered. In this latter case, the composition is treated as a closed vector.

Value

A `data.frame` object containing the imputed compositional vector or data set expressed in the original scale.

References

Martin-Fernandez JA, Barcelo-Vidal C, Pawlowsky-Glahn V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Mathematical Geology 2003; 35: 253-78.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. Analytica Chimica Acta 2013; 764: 32-43.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

`zPatterns`, `lrEM`, `lrSVD`, `lrDA`, `multLN`, `multKM`, `cmultRepl`

Examples

```
# A compositional vector (NA indicates nondetect)
y <- c(0.6,NA,0.25,0.03,0.12,NA)
dl <- c(0,0.01,0,0,0,0.005)
# Using the default frac = 0.65
```

```
yr <- multRepl(y,label=NA,dl=dl)
round(yr,4)

# Data set closed to 100 (percentages, common dl = 1%)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,26.20,0.00,15.22,6.80,12.05,
              10.76,31.36,7.10,12.74,31.34,6.70,
              10.85,46.40,31.89,10.86,0.00,0.00,
              7.57,11.35,30.24,6.39,13.65,30.80,
              38.09,7.62,23.68,9.70,20.91,0.00,
              27.67,7.15,13.05,32.04,6.54,13.55,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_multRepl <- multRepl(X,label=0,dl=rep(1,6))

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_multRepl2 <- multRepl(X,label=0,dl=mdl)

# Non-closed compositional data set
data(LPdata) # data (ppm/micrograms per gram)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdata_multRepl <- multRepl(LPdata,label=0,dl=dl)

# Two subsets of limits of detection
data(LPdata)
dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
 # DLs for first 50 samples of LPdata
dl1 <- matrix(rep(1,50),ncol=1)%*%dl
 # DLs for last 46 samples of LPdata
dl2 <- matrix(rep(1,46),ncol=1)%*%c(1,0.5,0,0,2.5,0,5.5,0.75,0.3,1.5,1,0,0,600,8)

mdl <- rbind(dl1,dl2)
LPdata_multRepl2 <- multRepl(LPdata,label=0,dl=mdl)

# Data set with missing values closed to 100 (percentages)
X <- matrix(c(10.47,8.58,59.72,19.30,1.93,
              12.13,7.44,62.87,16.37,1.19,
                 NA,7.30,75.91,16.79,NA,
               9.77,7.80,65.68,14.78,1.97,
              10.79,9.55,65.87,12.41,1.38,
              14.54,8.18,64.55,12.73,NA,
              12.28,7.58,66.01,12.93,1.20,
              28.09,22.92,NA,40.11,8.88,
               7.02,6.30,75.65,11.03,NA),byrow=TRUE,ncol=5)
```

```
X_multReplMiss <- multRepl(X,label=NA,imp.missing=TRUE)

# Non-closed compositional data set
data(LPdata) # (in ppm units)
# Treating zeros as missing data for illustration purposes only
LPdata_multReplMiss <- multRepl(LPdata,label=0,imp.missing=TRUE)
# Negative values generated (see e.g. K and Rb in sample #60)

# Workaround: use residual part to fill up the gap to 10^6 for imputation
LPdata_multReplMiss2 <- multRepl(LPdata,label=0,imp.missing=TRUE,closure=10^6)
```

`multReplus` *Multiplicative simple replacement (plus)*

Description

This function implements an extended version of multiplicative simple imputation (`multRepl` function) to simultaneously deal with both zeros (i.e. data below detection limit, rounded zeros) and missing data in compositional data sets.

Note: zeros and missing data must be labelled using 0 and `NA` respectively to use this function.

Usage

```
multReplus(X, dl = NULL, frac = 0.65, closure = NULL,
              z.warning = 0.8, z.delete = TRUE, delta = NULL)
```

Arguments

`X`	Compositional data set (`matrix` or `data.frame` class).
`dl`	Numeric vector or matrix of detection limits/thresholds. These must be given on the same scale as `X`. If `NULL` the column minima are used as thresholds.
`frac`	Fraction of the detection limit/threshold used for imputation (default = 0.65, expressed as a proportion).
`closure`	Closure value used to add a residual part for imputation (see below).
`z.warning`	Threshold used to identify individual rows or columns including an excess of zeros/unobserved values (to be specify in proportions, default `z.warning=0.8`).
`z.delete`	Logical value. If set to `TRUE`, rows/columns identified by `z.warning` are omitted in the imputed data set. Otherwise, the function stops in error when rows/columns are identified by `z.warning` (default `z.delete=TRUE`).
`delta`	This argument has been deprecated and replaced by `frac` (see package's NEWS for details).

Details

The procedure firstly replaces missing data using the estimated geometric mean based on the observed values and then zeros using `frac*dl`. The observed components are applied a multiplicative adjustment to preserve the multivariate compositional properties of the samples.

Note: negative values can be generated when unobserved components are a large portion of the composition, which is more likely for missing data (e.g in major chemical elements) and non-closed compositions. A workaround is to add a residual filling the gap up to the closure/total when possible. This is done internally when a value for `closure` is specified (e.g. `closure=10^6` if ppm or `closure=100` if percentages). The residual is discarded after imputation.

See `?multRepl` for more details.

Value

A `data.frame` object containing the imputed compositional data set expressed in the original scale.

References

Martin-Fernandez JA, Barcelo-Vidal C, Pawlowsky-Glahn V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Mathematical Geology 2003; 35: 253-78.

Palarea-Albaladejo J, Martin-Fernandez JA. Values below detection limit in compositional chemical data. Analytica Chimica Acta 2013; 764: 32-43.

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

`multRepl`, `lrEMplus`, `lrSVDplus`

Examples

```
# Data set closed to 100 (percentages, common dl = 1%)
# (Note that zeros and missing in the same row are allowed)
X <- matrix(c(26.91,8.08,12.59,31.58,6.45,14.39,
              39.73,41.42,0.00,NA,6.80,12.05,
              NA,35.13,7.96,14.28,35.12,7.51,
              10.85,46.40,31.89,10.86,0.00,0.00,
              10.85,16.27,NA,9.16,19.57,44.15,
              38.09,7.62,23.68,9.70,20.91,0.00,
              NA,9.89,18.04,44.30,9.04,18.73,
              44.41,15.04,7.95,0.00,10.82,21.78,
              11.50,30.33,6.85,13.92,30.82,6.58,
              19.04,42.59,0.00,38.37,0.00,0.00),byrow=TRUE,ncol=6)

X_multReplus <- multReplus(X,dl=rep(1,6))

# Multiple limits of detection by component
mdl <- matrix(0,ncol=6,nrow=10)
```

```
mdl[2,] <- rep(1,6)
mdl[4,] <- rep(0.75,6)
mdl[6,] <- rep(0.5,6)
mdl[8,] <- rep(0.5,6)
mdl[10,] <- c(0,0,1,0,0.8,0.7)

X_multReplus2 <- multReplus(X,dl=mdl)

# Non-closed compositional data set
data(LPdataZM) # (in ppm; 0 is nondetect and NA is missing data)

dl <- c(2,1,0,0,2,0,6,1,0.6,1,1,0,0,632,10) # limits of detection (0 for no limit)
LPdataZM2 <- subset(LPdataZM,select=-c(Cu,Ni,La)) # select a subset for illustration purposes
dl2 <- dl[-c(5,7,10)]

## Not run:
LPdataZM2_multReplus <- multReplus(LPdataZM2,dl=dl2)
# Negative values generated (see e.g. K in sample #64)

## End(Not run)

# Workaround: use residual part to fill up the gap to 10^6 for imputation
LPdataZM2_multReplus <- multReplus(LPdataZM2,dl=dl2,closure=10^6)
```

---

perLog *Test of differences in group means for compositional data*

Description

A nonparametric permutation test to assess the hypothesis of equality of means between subsets of observations according to an externally or internally defined factor variable. If any, zero patterns are considered as default internal grouping factor.

Usage

```
perLog(X, groups = NULL, p = "auto", alpha = 0.05, R = 1000,
          posthoc.g = FALSE, posthoc.lr = FALSE, mAdj = "BH")
```

Arguments

X	Compositional data set (matrix or data.frame class).
groups	Factor variable indicating the grouping structure. If NULL (default), any zero patterns in the data will be used as internal grouping factor. Note that if a grouping factor is set by the user, then any zeros in the data must be previously dealt with, e.g. by imputation.
p	Power parameter selected in overall dissimilarity test statistic, either automatically (default = "auto") or manually fixed.

alpha	Significance level parameter (default = 0.05).
R	Number of permutation resamples (default = 1000).
posthoc.g	Logical. If TRUE, performs post-hoc analysis for pairs of groups (default = FALSE).
posthoc.lr	Logical. If TRUE, performs post-hoc analysis for logratios (default = FALSE).
mAdj	Adjustment of p-values for multiple post-hoc testing (see p.adjust). Default is Benjamini and Hochberg's FDR method (default = "BH").

Details

The test relies on the unique pairwise logratios between parts of the given composition. It assesses whether the observed overall dissimilarity is significantly different from that expected under the null hypothesis of equal group means. If so, it can perform post-hoc analyses by pairs of groups and pairwise logratios, evaluating their relative contributions to dissimilarity at group and overall levels. The p-values in post-hoc testing are adjusted for multiple comparisons using the specified method. In the case of internal grouping defined by zero patterns, strings of binary codes are used to label each pattern in the output, with 0 indicating no zero part and 1 indicating zero part.

The power parameter p can be either automatically selected or manually fixed. For automatic selection, a simple conservative strategy is implemented starting with p = 10, as a liberal reference, and then successively setting p = {2, 3, ..., 9} until less significant differences are no longer obtained at the overall and group comparison levels.

Value

A list object of class "perLog.output" containing summaries of results:

disOv	Overall dissimilarity test statistic.
pvalOv	Overall permutation p-value.
p	Power parameter used in overall dissimilarity test statistic.
posthoc.groups	If 'posthoc.g = TRUE' and the main test is significant, results of the post-hoc analysis for pairs of groups.
posthoc.logratios	If 'posthoc.lr = TRUE' and the main test is significant, results of the post-hoc analysis for pairwise logratios.
wts	Welch's t-statistic.
disEl	Pairwise logratio elemental dissimilarity.
rcElBg	Relative contribution of elemental dissimilarity to between-group dissimilarity.
rcElOv	Relative contribution of elemental dissimilarity to overall dissimilarity.
pvalElAdj	Adjusted p-value in post-hoc comparison.
parts0	If 'groups = NULL', list containing original names of zero parts in the respective zero patterns.

References

Štefelová N, Palarea-Albaladejo J, Martín-Fernández JA. A permutation test of differences between externally or internally defined groupings in compositional data sets. Statistical Methods in Medical Research. 2026;0(0). <doi:10.1177/09622802251413737>

See Also

zPatterns

Examples

```
# Load the Water data set
data(Water)
# Visualise zero patterns and select the first three for illustration
tmp <- zPatterns(Water, label = 0)
Water2 <- Water[tmp %in% c(1,2,3),]
# Test overall differences by zero pattern on the selected data set
zPatterns(Water2, label = 0)
perLog(Water2)
```

Pigs *Pigs data set*

Description

Count data set consisting of scan sample behavioural observations of a group of 29 sows during a day from 7:30am to 3:30pm, and recorded every 5 minutes (97 times). Six locations were considered: straw bed (BED), half in the straw bed (HALF.BED), dunging passage (PASSAGE), half in the dunging passage (HALF.PASS), feeder (FEEDER) and half in the feeder (HALF.FEED).

Usage

```
data(Pigs)
```

Format

A data.frame with 29 observations on the following 6 variables.

`BED` a numeric vector

`HALF.BED` a numeric vector

`PASSAGE` a numeric vector

`HALF.PASS` a numeric vector

`FEEDER` a numeric vector

`HALF.FEED` a numeric vector

Source

Data set kindly provided by the Animal Behaviour and Welfare group at Scotland's Rural College (SRUC), Scotland, UK.

Examples

```
data(Pigs)
```

splineKM *Display Kaplan-Meier empirical cumulative distribution function and smoothing spline curve fit*

Description

This function shows the empirical cumulative distribution function (ECDF) for left-censored data as estimated by the Kaplan-Meier (KM) method and a cubic smoothing spline fitted to it (KMSS method, see multKM).

Usage

```
splineKM(x, label = NULL, dl = NULL, n.knots = NULL,
            legend.pos = "bottomright",
            ylab = "ECDF", xlab = "Value",
            col.km = "black", lty.km = 1, lwd.km = 1,
            col.sm = "red", lty.sm = 2, lwd.sm = 2, ...)
```

Arguments

x	Numerical data vector (vector class).
label	Unique label (numeric or character) used to denote left-censored values in x.
dl	Numeric vector of detection limits/thresholds for each element of x (same length as x). These must be given on the same scale as x (use e.g. 0 for detected data).
n.knots	Integer or function giving the number of knots used for fitting a cubic smoothing spline to the KM ECDF (see smooth.spline for default value).
legend.pos	Location of the graph legend. Choose one amongst "bottomleft", "bottomright" (default), "topleft" or "topright".
ylab	Title for y-axis.
xlab	Title for x-axis.
col.km	Plotting color for KM ECDF (see base graphical parameters par).
lty.km	Line type for KM ECDF (see base graphical parameters par).
lwd.km	Line width for KM ECDF (see base graphical parameters par).
col.sm	Plotting color for smoothing spline curve.
lty.sm	Line style for smoothing spline curve.
lwd.sm	Line width for smoothing spline curve.
...	Other graphical parameters.

Value

Graphical output.

Examples

```
data(Water)
data(mdl)

# Examine default spline smoothed KM ECDF fit for Potassium and Sulphate
splineKM(Water[,1],label=0,mdl[,1])
splineKM(Water[,4],label=0,mdl[,4],xlim=c(28,41))

# Reduce to 5 knots for Potassium
splineKM(Water[,1],label=0,mdl[,1],n.knots=5)
```

Water *Water data set*

Description

100 simulated samples of a 4-part groundwater composition in percentage subject to multiple limits of detection by component. The associated matrix of limits of detection is stored in `mdl`.

Usage

```
data(Water)
```

Format

A `data.frame` with 100 observations on the following 4 variables.

`Potassium` a numeric vector

`Arsenic` a numeric vector

`Calcium` a numeric vector

`Sulphate` a numeric vector

Details

Three limits of detection (0.75, 1 and 1.25) were considered for Potassium, four for Arsenic (1.5, 3, 4 and 5), two for Sulphate (29 and 35) and no one for Calcium. In the case of Sulphate, the detection limit equal to 29 is the minimum value registered for that component. All nondetects coded as 0.

Examples

```
data(Water)
zPatterns(Water,label=0)
```

zCompositions *Treatment of Zeros, Left-Censored and Missing Values in Compositional Data Sets*

Description

Following compositional data analysis principles, this package provides simple and friendly tools to explore and impute zeros, left-censored (such as rounded zeros or values below single or multiple limits of detection; a.k.a nondetects) and missing data; including zero pattern/group-wise data analysis and testing procedures.

Details

Package:	zCompositions
Type:	Package
Version:	1.6.0
Date:	2025-07-07
License:	GPL (>= 2)

Author(s)

Javier Palarea-Albaladejo and Josep Antoni Martin-Fernandez

Maintainer: Javier Palarea-Albaladejo <javier.palarea@udg.edu>

References

Martin-Fernandez, J.A., Barcelo-Vidal, C., Pawlowsky-Glahn, V., 2003. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. Mathematical Geology 35 (3): 253-27.

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J., 2012. Model-based replacement of rounded zeros in compositional data: Classical and robust approaches. Computational Statistics and Data Analysis 56: 2688-2704.

Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J., 2015. Bayesian-multiplicative treatment of count zeros in compositional data sets. Statistical Modelling 15 (2): 134-158.

Palarea-Albaladejo, J., Martin-Fernandez, J.A., Gomez-Garcia, J., 2007. A parametric approach for dealing with compositional rounded zeros. Mathematical Geology 39 (7): 625-645.

Palarea-Albaladejo, J., Martin-Fernandez, J.A., 2008. A modified EM alr-algorithm for replacing rounded zeros in compositional data sets. Computers & Geosciences 34 (8): 902-917.

Palarea-Albaladejo, J., Martin-Fernandez, J.A., 2013. Values below detection limit in compositional chemical data. Analytica Chimica Acta 764: 32-43.

Palarea-Albaladejo, J., Martin-Fernandez J.A., Olea, R.A., 2014. A bootstrap estimation scheme for chemical compositional data with nondetects. Journal of Chemometrics 28: 585-599.

Palarea-Albaladejo J. and Martin-Fernandez J.A., 2015. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligent Laboratory Systems 143: 85-96.

Palarea-Albaladejo, J, Antoni Martín-Fernández, J, Ruiz-Gazen, A, Thomas-Agnan, C., 2022. lrSVD: An efficient imputation algorithm for incomplete high-throughput compositional data. Journal of Chemometrics 36: e3459.

See Also

Aitchison, J., 1986. The Statistical Analysis of Compositional Data. Monographs on Statistics and Applied Probability. Chapman and Hall Ltd., London, UK (re-edited in 2003 with additional material).

Filzmoser, P., Hron, K., Templ, M., 2018. Applied Compositional Data Analysis. With Worked Examples in R. Springer, Switzerland.

Filzmoser P., Hron K., Martín-Fernández J.A., Palarea-Albaladejo J. (eds.), 2021. Advances in Compositional Data Analysis. Springer, Cham.

Pawlowsky-Glahn, V., Buccianti, A. (Eds.), 2011. Compositional Data Analysis: Theory and Applications. John Wiley & Sons, Ltd., Chichester,UK.

Pawlowsky-Glahn, V., Egozcue, J.J., Tolosana-Delgado, R., 2015. Modeling and analysis of compositional data. John Wiley & Sons, Ltd., Chichester, UK.

van den Boogaart, K.G., Tolosana-Delgado, R., 2013, Analyzing Compositional Data with R. Springer-Verlag, Berlin, Germany.

`zPatterns` *Find and display patterns of zeros/missing values in a data set*

Description

This function summarises the patterns of zero and/or missing values in a data set and returns a vector of pattern numbers.

Usage

```
zPatterns(X, label = NULL, plot = TRUE,
             axis.labels = c("Component", "Pattern ID"),
             bar.ordered = as.character(c(FALSE,FALSE)),
             bar.colors = c("red3", "red3"), bar.labels = FALSE,
             show.means = FALSE, round.means = 2, cex.means = 1,
             type.means = c("cgm","am"),
             cell.colors = c("dodgerblue", "white"),
             cell.labels = c(label, paste("No", label)), cex.axis = 1.1,
             grid.color = "black", grid.lty = "dotted",
             legend = TRUE, suppress.print = FALSE, ...)
```

Arguments

X	Data set (matrix or data.frame class).
label	Unique label (numeric or character) used to identify zeros/unobserved values in X.
plot	Logical value indicating whether a graphical summary of the patterns is produced or not (default plot=TRUE).
axis.labels	Vector of axis labels for the table of patterns (format c("x-axis","y-axis")).
bar.ordered	Vector of logical values to order table of patterns according to frequencies by patterns, component or both; with the first element referring to the patterns and the second to the components (default c(FALSE,FALSE)).
bar.colors	Colors for the margin barplots (format c("col.top","col.right")).
bar.labels	Logical value indicating if labels showing percentages must be added to the margin barplots (default bar.labels=FALSE).
show.means	Logical value indicating if mean values by pattern are shown on the graphical summary table (default show.means=FALSE).
round.means	When show.means=TRUE, number of decimal places for the mean values shown (2=default).
cex.means	When show.means=TRUE, numeric character expansion factor; character size for the mean values shown (1=default).
type.means	When show.means=TRUE, statistic used for computing the means. Either compositional geometric mean (type.means=cgm, in percentage units, default) or standard arithmetic mean (type.means=am).
cell.colors	Vector of colors for the table cells (format c("col.unobserved","col.observed")).
cell.labels	Labels for the cells (format c("Unobserved","Observed"), default c(label,paste("No",label))).
cex.axis	Axis labels scaling factor relative to default.
grid.color	Color of the grid lines (default "black").
grid.lty	Style of the grid lines (default "dotted", see lty in par).
legend	Logical value indicating if a legend must be included (default legend=TRUE).
suppress.print	Suppress printed feedback (default suppress.print=FALSE).
...	Other graphical parameters.

Value

Vector (factor type) of pattern IDs corresponding to each row of X.

By default, a summary table is printed showing patterns in the data according to label and some summary statistics: number of zero/missing components by pattern (No.Unobs), pattern frequency in percentage, percentage zero/missing values by component (column) and overall percentage of zero/missing values in the data set. The symbols + and - indicate, respectively, zero/missing and observed components within each pattern. A graphical version of the summary table is returned including barplots on the margins displaying percentage zero/missing and compositional geometric means by pattern (if show.means=TRUE; expressed in percentage scale). Common arithmetic means can be also shown for the case of ordinary data (type.means="am"), however this is not recommended for compositional data.

The patterns are assigned ID number and by default arranged in the table in the same order as they are found in the data set. The argument `bar.ordered` can be used to re-arrange the display according to frequencies of patterns, of unobserved values by component or both.

A warning message is shown if zeros or NA values not identified by `label` are present in the data set. These will be ignored for the graphical display and numerical summaries of patterns, which will be only based on `label`.

Check out 'plus' functions to deal with zeros and missing data simultaneously.

References

Palarea-Albaladejo J. and Martin-Fernandez JA. zCompositions – R package for multivariate imputation of left-censored data under a compositional approach. Chemometrics and Intelligence Laboratory Systems 2015; 143: 85-96.

See Also

`lrEM`, `lrEMplus`, `lrDA`, `multRepl`, `multReplus`, `multLN`, `multKM`, `cmultRepl`

Examples

```
data(LPdata)

pattern.ID <- zPatterns(LPdata,label=0)

LPdata[pattern.ID==5,]
LPdata[pattern.ID==7,]
LPdata[pattern.ID==10,]

# Modify cell labels and show percentages along with barplots
pattern.ID <- zPatterns(LPdata,label=0,
              cell.labels=c("Zero","Non-zero"),bar.labels=TRUE)

# Show compositional geometric means (in %) per zero pattern
zPatterns(LPdata,label=0,show.means=TRUE)

# Same but orderer by pattern frequency and incidence of zeros by component
zPatterns(LPdata,label=0,bar.ordered=c(TRUE,TRUE),,bar.labels=TRUE,show.means=TRUE)

# Data set with zeros and missing data (0 = zero; NA = missing) (see lrEMplus function).
data(LPdataZM)

# Show missingness patterns only
zPatterns(LPdataZM,label=NA)

# Show zero patterns only and means by pattern based on available data
# (blanks indicate not enough data available for computation)
zPatterns(LPdataZM,label=0,show.means=TRUE)
```

Index