Package: ChemometricsWithR (via r-universe)

June 29, 2024

Type Package

Title Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences (2nd Edition)

Version 0.2.0

Author Ron Wehrens [aut, cre]

Maintainer Ron Wehrens <ron.wehrens@gmail.com>

Description Functions and scripts used in the book ``Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences", 2nd edition, by Ron Wehrens, Springer (2019).

URL https://github.com/rwehrens/ChemometricsWithR

BugReports https://github.com/rwehrens/ChemometricsWithR/issues

Depends R (>= 2.10)

Imports pls, kohonen, MASS

Suggests fastICA, class, ptw, dtw, signal, cluster, mclust, nnet, e1071, rda, rpart, sfsmisc, boot, ipred, randomForest, ada, leaps, glmnet, subselect, BioMark, rrcov, ALS, alsace, fpc, RColorBrewer

License GPL (>= 2)

LazyLoad yes

Repository https://zeehio.r-universe.dev

RemoteUrl https://github.com/rwehrens/ChemometricsWithR

RemoteRef HEAD

RemoteSha 9d15f50972ffa7fe254567c097eab7cbced586c6

Contents

`ChemometricsWithR-package`

Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences (2nd Edition)

Description

Functions and scripts used in the book "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences", 2nd edition, by Ron Wehrens, Springer (2019).

Details

Package accompanying the second edition of the book "Chemometrics with R". The package will not be hosted on CRAN and therefore no longer has to comply to the size constraints that CRAN has imposed. This simplifies matters considerably for readers and in particular removes the need to install a separate data package.

The scripts in the `demo` directory can be used to reproduce the results and plots of the individual chapters. For Chapter 3, for example, simply run `demo("chapter3.R")`.

Author(s)

Ron Wehrens [aut, cre]

Maintainer: Ron Wehrens <ron.wehrens@gmail.com>

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences", 2nd edition, Springer, Heidelberg, 2019.

AdjRkl *Adjusted Rand Index*

Description

The Adjusted Rand Index is a measure of similarity for two groupings or clusterings. A value of 1 indicates total agreement.

Usage

```
AdjRkl(part1, part2)
```

Arguments

part1	First partitioning.
part2	Second partitioning.

Value

Number.

Author(s)

Ron Wehrens

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

Examples

```
data(wines, package = "kohonen")
wines.dist <- dist(scale(wines))
wines.sl <- hclust(wines.dist, method = "single")
wines.cl <- hclust(wines.dist, method = "complete")

AdjRkl(cutree(wines.sl, 4), cutree(wines.cl, 4))
```

arabidopsis *LC-MS metabolomics data sets from Arabidopsis samples*

Description

LC-MS data from Arabidopsis thaliana samples. The `arabidopsis` data object contains relative intensities of 567 reconstructed metabolites (columns) for 761 samples (rows). A sizeable fraction of intensities are missing, in most cases because the corresponding metabolites are below the detection level. The corresponding meta-information object (`arabidopsis.Y`)contains for every sample batch and sequence information, as well as (coded) information on the genotype and the sample type (study sample or reference sample). Processing of the raw data has been done with Metalign and MSClust programs.

Usage

```
data(arabidopsis)
```

References

"@ArticleWehrens2016, author = Ron Wehrens and Jos.~A.~Hageman and Fred~van~Eeuwijk and Rik~Kooke and P\'adraic~J.~Flood and Erik Wijnker and Joost~J.B.~Keurentjes and Arjen~Lommen and Henri\"ette~D.L.M.~van~Eekelen and Robert~D.~Hall and Roland~Mumm and Ric~C.H.~de~Vos, title = Improved batch correction in untargeted MS-based metabolomics, journal = Metabolomics, year = 2016, volume = 12, DOI = 10.1007/s11306-016-1015-8, pages = 1–12 "

Examples

```
data(arabidopsis)
dim(arabidopsis)
sum(is.na(arabidopsis))
dim(arabidopsis.Y)
```

bdata *HPLC-UV data of two chemical mixtures*

Description

Two chemical mixtures of three compounds have been measured using HPLC-UV. Two of the compounds are known: diazinon and parthion-ethyl, both organophosphorus pesticides. Each data matrix consists of 73 wavelengths and 40 time points. The challenge is to infer the pure spectra of the individual compounds, as well as their time profiles.

Usage

```
data(bdata)
```

Format

A list of four elements. The first two, d1 and d2, are the mixture matrices of the two analytes and one unknown interferent. The last two, sp1 and sp2, contain the pure spectra of the two analytes.

Source

Original matlab data files obtained from http://www.ub.edu/mcr/web_mcr/download_dataHPLC.html (bdataset.zip). No longer available.

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". Springer, 2nd edition, Heidelberg, 2019.

R. Tauler, S. Lacorte and D. Barcelo. "Application of multivariate curve self-modeling curve resolution for the quantitation of trace levels of organophosphorous pesticides in natural waters from interlaboratory studies". J. of Chromatogr. A, 730, 177-183 (1996).

Examples

```
data(bdata)
persp(bdata$d1, phi = 20, theta = 34, expand = .5,
      xlab = "Time", ylab = "Wavelength")
```

Error *Often-used error functions*

Description

Error functions for classification and regression

Usage

```
rms(x, y)
err.rate(x, y)
```

Arguments

x, y	True or predicted values, either numbers or factors.

Value

Function rms returns the root-mean-square error for real-valued x and y vectors. Function err.rate returns the fraction of non-matching cases in x and y (real numbers or factors).

Author(s)

Ron Wehrens

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

`gini` *Gini impurity index for cart objects*

Description

Calculation of the change in the Gini impurity index for classification and regression trees. The function returns changes in the gini index associated with using individual values of `x` as split points. Included for demonstration purposes.

Usage

```
gini(x, clss)
```

Arguments

`x`	Numeric vector of length n.
`clss`	Clss labels, length n.

Value

The change in Gini impurity index, given a vector of possible splits, and a vector of clss labels. Lower values indicate more pure leaves.

Author(s)

Ron Wehrens

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

`lcmsimage` *Display LC-MS data*

Description

LC-MS data are characterised by peak positions in two dimensions, the retention time dimension and the mass-to-charge ratio (m/z) dimension. The plot shows this plane (corresponding to one sample, basically an intensity matrix), with projections to the top and the right.

Usage

```
lcmsimage(z, rt, mz, xlim = range(rt), ylim = range(mz),
          zmin = 0, xlab = "Time (s)", ylab = "m/z",
          ColourScale = c("exponential", "linear"), nBreaks = 12,
          colors = terrain.colors(nBreaks),
          PlotProjections = c("max", "sum", "none"),
          ncolorbarticks = 5,
          colorbarticks = axisTicks(zlim2, log = ColourScale == "exponential",
          nint = ncolorbarticks), ...)
```

Arguments

`z`	the intensity matrix
`rt`	the vector of time points corresponding to the x axis
`mz`	the vector of m/z values corresponding to the y axis
`xlim`, `ylim`	defining which part of the data is shown, by default all
`zmin`	minimum of the intensity range, default zero
`xlab`, `ylab`	axis labels
`ColourScale`	whether to use an exponential (default) or linear color scale
`nBreaks`, `colors`	which and how many colours to use
`PlotProjections`	ways to calculate the projections to the top and right of the figure. Choosing `'none'` here suppresses the projections. Default is `'max'` (corresponding to a silhouette view, only the largest value is shown) but `'sum'` is also possible, which leads to the total-ion chromatogram and the direct injection mass spectrum.
`ncolorbarticks`, `colorbarticks`	control over the tick marks of the color bar
`...`	other arguments for the projection plots

Author(s)

Ron Wehrens, Tom Bloemberg

Examples

```
data(lcms,package = "ptw")
mycols <- terrain.colors(40)
lcmsimage(t(lcms[,,1]),
          rt = seq(2000, 5500, length = 2000),
          mz = seq(550, 599.5, length = 100),
          zmin = 1, colors = mycols, ncolorbarticks = 10)
```

MCR *Functions for Multivariate Curve Resolution*

Description

Multivariate Curve Resolution, or MCR, decomposes a bilinear matrix into its pure components. A classical example is a matrix consisting of a series of spectral measurements on a mixture of chemicals for following the reaction. At every time point, a spectrum is measured that is a linear combination of the pure spectra. The goal of MCR is to resolve the pure spectra and concentration profiles over time.

Usage

```
mcr(x, init, what = c("row", "col"), convergence = 1e-08,
    maxit = 50)
efa(x, ncomp)
```

Arguments

x	Data matrix
init	Initial guess for pure compounds
what	Whether the pure compounds are rows or columns of the data matrix
convergence	Convergence criterion
maxit	Maximal number of iterations
ncomp	Number of pure compounds

Details

MCR uses repeated application of least-squares regression to find pure profiles and spectra. The method is iterative; EFA is a method to provide initial guesses.

Value

Function mcr returns a list containing

C	An estimate of the pure "concentration profiles"
S	An estimate of the pure "spectra"
resids	The residuals of the final decomposition

`rms` Root-mean-square values of the individual iterations

Function `efa` returns a list containing

`pure.compounds:`
A matrix containing `ncomp` pure compounds, usually concentration profiles at specific wavelengths

`forward:` The development of the singular values of the reduced data matrix when increasing the number of columns in the forward direction

`backward:` The development of the singular values of the reduced data matrix when increasing the number of columns in the backwarddirection

Author(s)

Ron Wehrens

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

Examples

```
data(bdata)
D1.efa <- efa(bdata$d1, 3)
matplot(D1.efa$forward, type = "l")
matplot(D1.efa$backward, type = "l")
matplot(D1.efa$pure.comp, type = "l")

D1.mcr.efa <- mcr(bdata$d1, D1.efa$pure.comp, what = "col")
matplot(D1.mcr.efa$C, type = "l", main = "Concentration profiles")
matplot(t(D1.mcr.efa$S), type = "l", main = "Pure spectra")
```

PCA *Principal Component Analysis*

Description

Functions for PCA: creating a PCA object, extracting variances, scores and loadings for individual PCs, projecting new data in the PC space, and reconstruction using a limited number of PCs.

Usage

```
PCA(X, warn = TRUE)
## S3 method for class 'PCA'
summary(object, varperc = 90, pc.select = c(1:5,10), ...)
variances(object, npc = maxpc)
## S3 method for class 'PCA'
scores(object, npc = maxpc, ...)
```

```
## S3 method for class 'PCA'
loadings(object, npc = maxpc, ...)
reconstruct(object, npc = maxpc)
project(object, npc = maxpc, newdata, ldngs)
```

Arguments

X	a matrix, with each row representing an object.
warn	logical, whether or not to give a warning when the data are not mean-centered.
object	an object of class "PCA" (see below).
varperc	variance threshold in the summary function.
...	extra arguments, e.g., for printing the variance table (digits = ...).
pc.select	PCs to be included in the summary function.
npc	the number of PCs to be returned.
newdata	data (with the same number of variables as the original data) that are to be projected into the space of the first npc PCs.
ldngs	loadings to be used; by default the PCA loadings.

Value

Function PCA returns an object of class "PCA" with components

scores	object weights per PC.
loadings	variable weights per PC.
var	variance explained per PC.
totalvar	The total variance in the data set.

Function summary.PCA gives a short summary of the PCA model, stating how many PCs are needed to cover a certain percentage of the total variance, and for selected PCs gives the (cumulative) variance explained.

Function variances returns the variances associated with each PC.

Function scores returns the scores associated with each PC.

Function loadings returns the loadings associated with each PC.

Function reconstruct returns the reconstruction of the original data matrix, based on npc PCs.

Function project projects the new data into the subspace spanned by the given loadings. If argument ldngs is given, arguments pcamod and npc are not needed.

Author(s)

Ron Wehrens

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

See Also

plot.PCA

Examples

```
data(wines, package = "kohonen")
wines.PC <- PCA(scale(wines))
```

PCA.plot *Principal Component Analysis plotting functions*

Description

Plotting functions for PCA: for scores, loadings, scores and loadings simultaneously (a biplot), and variances (a screeplot, where the log of the explained variance is plotted for each PC).

Usage

```
## S3 method for class 'PCA'
scoreplot(object, pc = c(1, 2), pcscores = scores(object),
          show.names = FALSE, xlab, ylab, xlim, ylim, ...)
## S3 method for class 'PCA'
loadingplot(object, pc = c(1, 2), pcloadings = loadings(object),
            scalefactor = 1, add = FALSE, show.names = FALSE,
            xlab, ylab, xlim, ylim, col = "blue", min.length =
            0.01, varnames = NULL, ...)
## S3 method for class 'PCA'
biplot(x, pc = c(1,2),
        show.names = c("none", "scores", "loadings", "both"),
        score.col = 1, loading.col = "blue",
        min.length = .01, varnames = NULL, ...)
screeplot(object, type = c("scree", "percentage"), npc, ...)
```

Arguments

x, object	an object of class "PCA" (see below).
pc	which PCs to show.
pcscores	matrix of scores, by default the scores of the PCA model object.
show.names	show names rather than plotting symbols. For loadingplot and scoreplot a logical (default: FALSE), for biplot one of 'scores', 'loadings', 'both' or 'none' (default).
xlab, ylab, xlim, ylim, col	graphical parameters of the plot.
pcloadings	matrix of loadings, by default the loadings of the PCA model object.
scalefactor	scaling factor for the loadings; used internally, when the loadingplot function is called from within biplot.PCA.

add	logical, whether to add to the existing plot (again, useful when loadingplot is called from within biplot.PCA).
npc	how many PCs to show in the scree plot (starting from 1).
type	show a real screeplot (scree) or show the percentage of variance explained (percentage).
score.col, loading.col	colours of the scores and loadings in a biplot.
min.length	minimal length of loading vectors to be plotted by arrows. Vectors that are too short lead to warning messages, are not interesting, and only clutter the graphic.
varnames	alternative vector of variable names.
...	Graphical arguments passed on to lower-level plotting functions.

Details

Score plots and loading plots show the amount of explained variance at the axis labels only when PCA has been performed at mean-centered data.

Author(s)

Ron Wehrens

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

See Also

PCA

Examples

```
data(wines, package = "kohonen")
wines.PC <- PCA(scale(wines))
wine.classes <- as.integer(vintages)
scoreplot(wines.PC, col = wine.classes, pch = wine.classes)
loadingplot(wines.PC, show.names = TRUE)
biplot(wines.PC, score.col = wine.classes)
screeplot(wines.PC)
```

pick.peaks *Peak-picking function.*

Description

Function to identify local maxima in a vector, typically a spectrum or a chromatogram.

Usage

```
pick.peaks(x, span)
```

Arguments

x	Numerical vector.
span	Neighbourhood, used to define local maxima.

Value

A vector containing positions of local maxima in the input data.

Author(s)

Ron Wehrens

Examples

```
if (require("ptw")) {
data(lcms, package = "ptw")
plot(lcms[1,,1], type = "l", xlim = c(1000, 1500))
abline(v = pick.peaks(lcms[1,,1], 20), col = "blue")
} else {
  cat("Package ptw not available.\nInstall it by typing 'install.packages(\"ptw\")'")
}
```

Prostate2000Raw *Prostate Cancer 2000 Raw Spectra*

Description

A data object of class msSet, consisting of 654 mass spectra (327 spectra in duplicate) from 2000 to 20000 Da, which were generated from patients with prostate cancer, benign prostatic hypertrophy, and normal controls. These spectra are already baseline corrected and normalized. Please see the references for more details.

Since the original package msProstate is orphaned at the end of 2012, the data are included in the ChemometricsWithR package so that the examples in the book are still executable. This manual page has been adapted to reflect this.

References

B.L. Adam, Y. Qu, J.W. Davis, M.D. Ward, M.A. Clements, L.H. Cazares, O.J. Semmes, P.F. Schellhammer, Y. Yasui, Z. Feng, and G.L. Wright, Jr., "Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy men," *Cancer Research*, 62(13):3609–14, 2002.

Y. Qu, B.L. Adam, Y. Yasui, M.D. Ward, L.H. Cazares, P.F. Schellhammer, Z. Feng, O.J. Semmes, and G.L. Wright Jr., "Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients", *Clinical Chemistry*, 48(10):1835–43, 2002.

R. Wehrens, "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

Examples

```
## Examples have been changed from the original man page upon inclusion
## in the ChemometricsWithRData package
data("Prostate2000Raw")

## plot a few spectra, partially
matplot(Prostate2000Raw$mz[1:8000],
        Prostate2000Raw$intensity[1:8000, 1:5], type = "l",
        lty = 1, col = 1:5, xlab = "m/z", ylab = "response")
```

`shootout` *Shootout NIR data*

Description

NIR data from 654 tablets, measured at two different instruments. The data have been divided in training, test and validation sets.

Usage

```
data(shootout)
```

Format

Variable `shootout` is a list containing spectral data of tablets, measured on two instruments, as well as response variables.

Details

For every tablet, three reponse variables are measured: the amount of active ingredient (nominally 200 mg/tablet), the weight and the hardness. Typically, one wants to estimate the amount of active ingredient from the NIR spectra, a straightforward multivariate calibration problem. The goal of the shootout competition was to find the optimal way to transfer a calibration model of the first instrument to the second.

Source

http://www.idrc-chambersburg.org/shootout2002.html

References

R. Wehrens. "Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences". 2nd edition, Springer, Heidelberg, 2019.

Examples

```
data(shootout)
plot(seq(600, 1898, by = 2), shootout$calibrate.1[1,], type = "l",
xlab = "wavelength", ylab = "log(1/T)")
```

Index