Package: ALS (via r-universe)

October 18, 2026

Type Package

Title Multivariate Curve Resolution Alternating Least Squares (MCR-ALS)

Version 0.0.7

Author Katharine M. Mullen

Maintainer Katharine Mullen <mullenkate@gmail.com>

Depends nnls (>= 1.1), Iso, R (>= 2.10)

Description Alternating least squares is often used to resolve components contributing to data with a bilinear structure; the basic technique may be extended to alternating constrained least squares. Commonly applied constraints include unimodality, non-negativity, and normalization of components. Several data matrices may be decomposed simultaneously by assuming that one of the two matrices in the bilinear decomposition is shared between datasets.

License GPL (>= 2)

NeedsCompilation no

Repository https://k-m-m.r-universe.dev

Date/Publication 2022-08-25 07:32:58 UTC

RemoteUrl https://github.com/cran/ALS

RemoteRef HEAD

RemoteSha 6e50b2a04af257881bd8602d54daf0f0178407c8

Contents

als 2
matchFactor 5
multiex 6
multiex1 6
plotS 7

Index **9**

als *alternating least squares multivariate curve resolution (MCR-ALS)*

Description

This is an implementation of alternating least squares multivariate curve resolution (MCR-ALS). Given a dataset in matrix form d1, the dataset is decomposed as d1=C %*% t(S) where the columns of C and S represent components contributing to the data in each of the 2-ways that the matrix is resolved. In forming the decomposition, the components in each way many be constrained with e.g., non-negativity, uni-modality, selectivity, normalization of S and closure of C. Note that if more than one dataset is to be analyzed simultaneously, then the matrix S is assumed to be the same for every dataset in the bilinear decomposition of each dataset into matrices C and S.

Usage

```
als(CList, PsiList, S=matrix(), WList=list(),
thresh =.001, maxiter=100, forcemaxiter = FALSE,
optS1st=TRUE, x=1:nrow(CList[[1]]), x2=1:nrow(S),
baseline=FALSE, fixed=vector("list", length(PsiList)),
uniC=FALSE, uniS=FALSE, nonnegC = TRUE, nonnegS = TRUE,
normS=0, closureC=list())
```

Arguments

CList	list with the same length as PsiList where each element is a matrix of dimension m by comp and represents the matrix C for each dataset
PsiList	list of datasets, where each dataset is a matrix of dimension m by n
S	matrix with n rows and comp columns, often representing (mass) spectra
WList	An optional list with the same length as PsiList, where each element is a matrix of dimension m by n giving the weight of that datapoint; note that if closure or normalization constraints are applied, then both are applied after the application of weights.
thresh	numeric value that defaults to .001; if ((oldrss - rss) / oldrss) < thresh then the optimization stops, where oldrss is the residual sum of squares at iteration x-1 and rss is the residual sum of squares at iteration x
maxiter	The maximum number of iterations to perform (where an iteration is optimization of either AList and C)
forcemaxiter	Logical indicating whether maxiter iterations should be performed even if the residual difference drops below thresh.
optS1st	logical indicating whether the first constrained least squares regression should estimate S or CList.
x	optional vector of labels for the rows of C, which are used in the application of unimodality constraints.
x2	optional vector of labels for the rows of S, which are used in the application of unimodality constraints.

baseline	logical indicating whether a baseline component is present; if baseline=TRUE then this component is exempt from constraints unimodality or non-negativity
fixed	list with the same length as PsiList in which each element is a vector of the indices of the components to fix to zero in each dataset
nonnegS	logical indicating whether the components (columns) of the matrix S should be constrained to non-negative values
nonnegC	logical indicating whether the components (columns) of the matrix C should be constrained to non-negative values
uniC	logical indicating whether unimodality constraints should be applied to the columns of C
uniS	logical indicating whether unimodality constraints should be applied to the columns of S
normS	numeric indicating whether the spectra are normalized; if normS>0, the spectra are normalized. If normS==1 the maximum of the spectrum of each component is constrained to be equal to one; if normS > 0 && normS!=1 then the norm of the spectrum of each component is constrained to be equal to one.
closureC	list; if the length is zero, then no closure constraints are applied. If the length is not zero, it should be equal to the number of datasets in the analysis, and contain numeric vectors consisting of the desired value of the sum of each row of the concentration matrix.

Value

A list with components:

CList	A list with the same length as the number of datasets, containing the optimized matrix C at termination scaled by the optimized amplitudes for that dataset from AList.
S	The matrix S given as input.
rss	The residual sum of squares at termination.
resid	A list with the same length as the number of datasets, containing the residual matrix for each dataset
iter	The number of iterations performed before termination.

Note

This function was used to solve problems described in

van Stokkum IHM, Mullen KM, Mihaleva VV. Global analysis of multiple gas chromatography-mass spectrometry (GS/MS) data sets: A method for resolution of co-eluting components with comparison to MCR-ALS. Chemometrics and Intelligent Laboratory Systems 2009; 95(2): 150-163.

in conjunction with the package TIMP. For the code to reproduce the examples in this paper, see examples_chemo.zip included in the inst directory of the package source code. .

References

Garrido M, Rius FX, Larrechi MS. Multivariate curve resolution alternating least squares (MCR-ALS) applied to spectroscopic data from monitoring chemical reactions processes. Journal Analytical and Bioanalytical Chemistry 2008; 390:2059-2066.

Jonsson P, Johansson A, Gullberg J, Trygg J, A J, Grung B, Marklund S, Sjostrom M, Antti H, Moritz T. High-throughput data analysis for detecting and identifying differences between samples in GC/MS-based metabolomic analyses. Analytical Chemistry 2005; 77:5635-5642.

Tauler R. Multivariate curve resolution applied to second order data. Chemometrics and Intelligent Laboratory Systems 1995; 30:133-146.

Tauler R, Smilde A, Kowalski B. Selectivity, local rank, three-way data analysis and ambiguity in multivariate curve resolution. Journal of Chemometrics 1995; 9:31-58.

See Also

matchFactor,multiex,multiex1, plotS

Examples

```
## load 2 matrix datasets into variables d1 and d2
## load starting values for elution profiles
## into variables Cstart1 and Cstart2
## load time labels as x, m/z values as x2
data(multiex)

## starting values for elution profiles
matplot(x,Cstart1,type="l")
matplot(x,Cstart2,type="l",add=TRUE)

## using MCR-ALS, improve estimates for mass spectra S and the two
## matrices of elution profiles
## apply unimodality constraints to the elution profile estimates
## note that the starting estimates for S just contain a dummy matrix

test0 <- als(CList=list(Cstart1,Cstart2),S=matrix(1,nrow=400,ncol=2),
PsiList=list(d1,d2), x=x, x2=x2, uniC=TRUE, normS=0)

## plot the estimated mass spectra
plotS(test0$S,x2)

## the known mass spectra are contained in the variable S
## can compare the matching factor of each estimated spectrum to
## that in S
matchFactor(S[,1],test0$S[,1])
matchFactor(S[,2],test0$S[,2])

## plot the estimated elution profiles
## this shows the relative abundance of the 2nd component is low
matplot(x,test0$CList[[1]],type="l")
matplot(x,test0$CList[[2]],type="l",add=TRUE)
```

`matchFactor` *Matching factor functions to describe similarity of two vectors*

Description

Matching factor functions to describe similarity of two vectors. This function may be useful to match an estimated mass spectrum against mass spectra of known compounds, in order to identify the compound represented by the estimated mass spectrum.

Usage

```
matchFactor(u, s, type="dot")
```

Arguments

`u`	numeric vector of length `n`
`s`	numeric vector of length `n`
`type`	character vector describing the matching factor function to apply; the choices are `"dot"` for the normalized dot product or `"euclid"` for the normalized euclidean distance.

Value

numeric between 0 and 1 representing the matching factor; vectors that are more similar have a larger matching factor. Note that if both `u` and `s` are all zero, we let the matching factor be 1; if one and only one of `u` and `s` are all zero, we let the matching factor be 0.

Author(s)

Katharine M. Mullen

References

Alfassi ZB. On the normalization of a mass spectrum for comparison of two spectra. Journal of the American Society for Mass Spectrometry 2004; 15:385-387.

Stein SE, Scott DR. Optimization and testing of mass spectral library search algorithms for compound identication. Journal of the American Society for Mass Spectrometry 1994; 5:859-866.

See Also

`als`

multiex — *Data inspired by GC mass spectrometry experiments*

Description

Data inspired by GC mass spectrometry experiments.

Usage

```
data("multiex")
```

Format

`d1` and `d2` are matrices of dimension 80 by 400 representing time and m/z resolved data.

`x` and `x2` represent the 80 times and 400 m/z values represented by the data, respectively.

`Cstart1` and `Cstart2` are matrices of dimension 80 by 2, representing starting values for elution profiles.

`S` represents mass spectra known to be represented in the data, as a 400 by 2 matrix.

Examples

```
data("multiex")
## mass spectra in the data
plotS(S,x2)

## starting values for elution profiles
matplot(x,Cstart1,type="l")
matplot(x,Cstart2,type="l",add=TRUE)
```

multiex1 — *Data inspired by GC mass spectrometry experiments*

Description

Data inspired by GC mass spectrometry experiments.

Usage

```
data("multiex1")
```

Format

PsiList is a list of 15 matrices of dimension 81 by 165, each representing time and m/z resolved data.

WList is a list of 15 matrices of dimension 81 by 165, in which each point is a weight to be applied to a given data point.

xm and xm2 represent the 81 times and 165 m/z values represented by each dataset in PsiList, respectively.

AList is a list of length 15, the elements of which represent estimates for the amplitude of each component in each of the 15 datasets.

C1 is a 81 by 2 matrix representing a starting value for the shape of the elution profiles.

Sm represents mass spectra known to be represented in the data, as a 165 by 2 matrix.

See Also

als

Examples

```
data("multiex1")
## mass spectra in the data
plotS(Sm,xm2)
```

---

plotS *Plots a matrix representing mass spectra*

Description

For each column in a matrix representing mass spectra, generates a sub-plot

Usage

```
plotS(S, x2, out="", filename=paste("S.", out, sep = ""),
col=vector(),cex=1, lab="",cex.lab=1)
```

Arguments

S	matrix representing mass spectra of dimension n by comp where comp is the number of spectra
x2	vector of masses that label the rows of S
out	if "", the plot is written to the current device; if "ps" a postscript file is written and if "pdf" then a pdf file is written
filename	character vector specifying the name of the file to write if out=TRUE
col	if length is greater than zero, then the color to plot each spectrum

cex	cex A numerical value giving the amount by which plotting text and symbols should be magnified relative to the default.
lab	cex.lab A character vector used as y-axis label.
cex.lab	cex.lab A numerical value giving magnification to be used for x and y labels relative to the default.

Author(s)

Katharine M. Mullen

See Also

als

Examples

```
## load example mass spectra S and vector of m/z values x2
data(multiex)

plotS(S,x2)
```

Index

∗ **datasets**
multiex, 6
multiex1, 6
∗ **hplot**
matchFactor, 5
plotS, 7
∗ **optimize**
als, 2

AList *(*multiex1*)*, 6
als, 2, *5*, *7*, *8*

C1 *(*multiex1*)*, 6
Cstart1 *(*multiex*)*, 6
Cstart2 *(*multiex*)*, 6

d1 *(*multiex*)*, 6
d2 *(*multiex*)*, 6

matchFactor, *4*, 5
multiex, *4*, 6
multiex1, *4*, 6

plotS, *4*, 7
PsiList *(*multiex1*)*, 6

S *(*multiex*)*, 6
Sm *(*multiex1*)*, 6

WList *(*multiex1*)*, 6

x *(*multiex*)*, 6
x2 *(*multiex*)*, 6
xm *(*multiex1*)*, 6
xm2 *(*multiex1*)*, 6