Package 'RGCxGC'

January 20, 2019

Type Package

Title Preprocessing and Multivariate Analysis of Bidimensional Gas Chromatography Data

Version 1.0.0

Date 2019-01-09

Author Cristian Quiroz-Moreno [aut, cre],
Guilherme L. Alexandrino [aut],
Noroska G.S. Mogollón [aut]

Maintainer Cristian Quiroz-Moreno <cristianquirozd1997@gmail.com>

Description Implementation of chemometrics analysis for bidimensional gas chromatography data. This package can handle data for common scientific data format (netCDF) and fold it to 2D chromatogram. Then, it can perform preprocessing and multivariate analysis. In the preprocessing algorithms, baseline correction, smoothing, and peak alignment are available.
While in multivariate analysis, multiway principal component analysis is incorporated.

License MIT + file LICENSE

Imports colorRamps (>= 2.3), Rdpack (>= 0.7), prettydoc (>= 0.2)

RdMacros Rdpack

Depends R (>= 3.5.0), stats, methods, graphics, RNetCDF (>= 1.9-1), ptw (>= 1.9-13)

LazyData true

RoxygenNote 6.1.1

Encoding UTF-8

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2019-01-20 17:30:03 UTC

R topics documented:

`aligned_GCxGC-class` *Subclass aligned_GCxGC*

Description

Subclass *aligned_GCxGC* are contained in *raw_GCxGC* super class. It is not contained in the *prepec_GCxGC* due to raw chromatograms can be aligned without a previous preproccesing technique Although it can improve the performance of the aligment, it is not mandatory.

Details

You can perform the aligment after some preprocessing technique as: baseline correction, or signal smoothing to imporve the performance of the alignment function.

baseline_corr *Bidimensional baseline correction*

Description

'baseline_corr' provides a bidimensional baseline correction using asymetric least squares.

Usage

```
baseline_corr(chromatogram)
```

Arguments

chromatogram	a *raw_GCxGC* object like with provided *name* and *mod_time* slots.

Details

This function takes a raw bidimensional chromatogram and performs the baseline correction with implemented function in `baseline.corr` (Eilers 2004).

References

Eilers PH (2004). "Parametric Time Warping." *Analytical Chemistry*, **76**(2), 404–411. ISSN 00032700, doi: 10.1021/ac034800e.

Examples

```
library(colorRamps)
chrom_name <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
chrom_2D <- read_chrom(chrom_name, 5L)
chrom_bsline <- baseline_corr(chrom_2D)
plot(chrom_bsline, nlevels = 150,
           color.palette = matlab.like)
```

batch_2DCOW *Two Dimensional COW in batch.*

Description

'batch_2DCOW' returns the aligned chromatogram in a named list slot. The first chromatogram is considered as the reference.

Usage

```
batch_2DCOW(chrom_names, mod_time, segments, max_warp)
```

Arguments

chrom_names	The names of the chromatograms to be aligned, the first chromatogram name will be considered as the reference chromatogram.
mod_time	The modulation time of the experiment.
segments	A two integer vector with number of segments which the first and second dimension will be subdivided, respectively.
max_warp	A two intger vector with the maximum warping parameter. *Name* to merge with the chromatograms.

Details

This is an adaptation of bidimesional COW alignment, first implementated in MATLAB. This function takes a sample chromatogram to be aligned to the reference. The argument [segment] will be used to split the whole chromatogram in n parts in the first and the second dimension respectevily. The [max_warp] argument provides de maximum tolerace of the signal transformation as well to the first and the second dimension.

Examples

```
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
GB09_fl <- system.file("extdata", "09GB.cdf", package = "RGCxGC")
chrom_nm <- c(GB08_fl, GB09_fl)

batch_alignment <- batch_2DCOW(chrom_nm, 5L, c(10, 40), c(1, 10))
```

---

batch_2DCOW-class *Subclass batch_2DCOW*

Description

Subclass *batch_2DCOW* are contained in *raw_GCxGC* super class. *batch_2DCOW* contains multiple aligned chromatograms, which the first one is consider as the reference chromatogram.

Details

You can perform the alignment after some preprocessing technic as: baseline correction, or signal smothing to imporve the performance of the aligment function, or with the raw chromatogram.

GCxGC-class *Class GCxGC*

Description

Class *GCxGC* defines the superclass of bidimensional comprehensive gas chromatography

Details

The validity function evaluates if the provied file can be readed in a netCDF format. The validation employs the function `open.nc` to check if the provided file inherits to NetCDF.

Slots

`name` the name of a netCDF file to which the data will be retrieved

`mod_time` A integer with the modulation time for the second dimension. Note the integer should be provide with an *L* at the end of the number.

joined_chrom-class *Class joined_chrom*

Description

Class *joined_chrom* defines the superclass to gather single chromatogram as well batch cromatograms into a singe list previos multiway principal compoment analysis

Slots

`chromatograms` A named list with all chromatograms.

`groups` A data.frame with the metadata. A column *Name* with the same name of the chromatograms

`time` The time range of chromatographic run

`mod_time` modulation time of the second dimension

join_chromatograms *Join two-dimensional chromatograms into a single R object*

Description

'join_chromatograms' save the chromatograms in a named list slot. Also, it saves information like metadata and retention times.

Usage

```
join_chromatograms(x, y, groups, ...)
```

Arguments

x, y	an GCxGC object, either single or batch chromatogram
groups	A data.frame containing the metadata. It must have a column named as *Name* to merge with the imported chromatograms.
...	other GCxGC object to be merged

Examples

```
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
GB09_fl <- system.file("extdata", "09GB.cdf", package = "RGCxGC")
GB08 <- read_chrom(GB08_fl, 5L)
GB09 <- read_chrom(GB09_fl, 5L)
join_gc <- join_chromatograms(GB08, GB09)
```

MPCA-class *Class MPCA*

Description

Class *MPCA* defines the superclass of Multiway Principal Component Analysis

Slots

scores A matrix with the eigenvalues of projected chromatograms into principal components space.

loadings The eigenvectors of each principal component.

summary The summary of the multiway principal component analysis.

groups A data.frame with the experiment metadata. It must have a column

time The time range of chromatographic run

mod_time modulation time of the second dimension *Name* to join with chromatograms.

MTBLS579 *Chromatograms from Dioagnostic Metabolite Biomarkers of Chronic Typhoid Carraige study*

Description

The dataset was retrieved from MetaboLights with the identifier number MTBLS79 https://www.ebi.ac.uk/metabolights/MTBLS579. Two groups from the entire study was downloaded: control and *S. typhi* carriage. The name files of control group are: 08GB, 09GB, and 14GB, which has the native name of 08_GB.cdf, 14_GB.cdf, and 09_GB.cdf in the MetaboLights database. For the *S. typhi* group the names are: 34GB, 24GB, 29GB, which has the native name of 34_GB.cdf, 24_GB.cdf and 29_GB.cdf in MetaboLights database.

Due to large size of chromatograms, this data is a subset of the whole chromatograms from 7 min to 18 min of chromatografic run. If you would like to acces the whole formated chromatograms, please go to https://github.com/DanielQuiroz97/MTBLS579/raw/master/MTBLS579.rda.

The original study was developed by Näsström et al. (2018).

Usage

```
data(MTBLS579)
```

Format

A joined_chrom object containing four slots:

chromatograms A named list with the two-dimensional chromatograms

groups The metadata containing two varaibles and six observations

time The retantion time range of the chromatographic run

mod_time The modulation time

Source

https://www.ebi.ac.uk/metabolights/MTBLS579

References

Näsström E, Jonsson P, Johansson A, Dongol S, Karkey A, Basnyat B, Tran Vu Thieu N, Trinh Van T, Thwaites GE, Antti H, Baker S (2018). "Diagnostic metabolite biomarkers of chronic typhoid carriage." *PLoS Neglected Tropical Diseases*, **12**(1), 1–15. ISSN 19352735, doi: 10.1371/journal.pntd.0006215.

m_prcomp *Multiway Principal Component Analysis*

Description

'MPCA' Performs a multiway principal components analysis on the given bidimensional chromatograms and returns the rusults as object of class MPCA. Before to perform the calculation, each given chromatogramas are unfolded to a single dimension. All chromatograms are merged and principal component analysis is performed with the built-in `prcomp` function. The print method for these objects prints the summary of the analysis. This algorithm was first presented by (Wold et al. 1987).

Usage

```
m_prcomp(chrom, center = TRUE, scale = TRUE, npcs = 3, ...)
```

Arguments

`chrom`	Multiple chromatograms readed or batch aligned
`center`	A logical value indicating whether the variables should be shifted to be zero centered. True is set by default and is strongly seggested not to change to False.
`scale`	a logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place. The default is True to give the same variable importance in chemometrics.
`npcs`	an integer indicating how many principals components are desired to mantain. The default is 3 principal components.
`...`	Other argments passed to prcomp function.

Value

MPCA returns a list whit class "MPCA" containing the summary of the analysis, the scores matrix, and unfolded loadings, and the metadata if it was providen when chromatograms were joined.

References

Wold S, Geladi P, Esbensen K, Öhman J (1987). "Multi-way principal components- and PLS-analysis." *Journal of Chemometrics*, **1**(January 1986), 41–56. ISSN 0886-9383, doi: 10.1002/cem.1180010107.

Examples

```
data(MTBLS579)

MTBLS579_mpca <- m_prcomp(MTBLS579)
print(MTBLS579_mpca)
scores(MTBLS579_mpca)
```

```
plot_loading(MTBLS579_mpca)
```

`plot` *Method plot*

Description

'plot' plot the bidimensional chromatogram as a filled contour plot

Usage

```
plot(Object, ...)

## S4 method for signature 'GCxGC'
plot(Object, ...)
```

Arguments

`Object`	a GCxGC chromatogram, it could be a raw, or preprocessed chromatogram
`...`	Other parameters passes to `filled.contour` function.

Details

This plot function employs the built-in countour function. As mentioned in Reichenbach et al. (2004), interpolation is usedto display non-native GCxGC data.

References

Reichenbach SE, Ni M, Kottapalli V, Visvanathan A (2004). "Information technologies for comprehensive two-dimensional gas chromatography." *Chemometrics and Intelligent Laboratory Systems*, **71**(2), 107–120. ISSN 01697439, doi: 10.1016/j.chemolab.2003.12.009.

Examples

```
library(colorRamps)
chrom_name <-  system.file("extdata", "08GB.cdf", package = "RGCxGC")
chrom_2D <- read_chrom(chrom_name, 5L)
plot(chrom_2D, nlevels = 150, color.palette = matlab.like)
```

plot_loading *Plot two dimensional loadings from MPCA*

Description

'plot_loading' plot the loadings of the previously MPCA performed.

Usage

```
plot_loading(Object, type = "n", pc = 1, ...)

## S4 method for signature 'MPCA'
plot_loading(Object, type = "b", pc = 1, ...)
```

Arguments

Object	a MPCA object
type	the value type of loadings, *p* for positive, *n* for negative, and *b* for negative and positive loading values.
pc	the number of the principal component to plot
...	Other parameters passes to filled.contour function.

Details

This function takes the loadings of MPCA and eval if a certain variable was removed previous compute de MPCA and fill the removed variables with cero. Then, it plots the loadings of a single principal component in two dimensions.

Examples

```
library(colorRamps)
data(MTBLS579)
# MPCA with mean-centered and scaled data
MTBLS579_mpca <- m_prcomp(MTBLS579)
# Negative loadings of the first principal component

plot_loading(MTBLS579_mpca, type = "n", pc = 1,
             color.palette = matlab.like)
# Positive loadings of the first principal component
plot_loading(MTBLS579_mpca, type = "p", pc = 1,
             color.palette = matlab.like)
```

preproc_GCxGC-class *Subclass preproc_GCxGC*

Description

Subclass *preproc_GCxGC* are contained in *raw_GCxGC* super class. It contains a dedicated slot to storage the preprocessed bidimensional chromatogram.

Details

After reading a bidimensional chromatogram, you can perform serveral preprocessing technicas as smothing, or baseline correction. It will create an object of subclass preproc_GCxGC.

print *Print MPCA summary*

Description

'print' call the MPCA object to print the summary analysis

Usage

```
print(Object)

## S4 method for signature 'MPCA'
print(Object)
```

Arguments

Object	a MPCA object

Details

This plot function employs the built-in print function

Examples

```
data(MTBLS579)
MTBLS_mpca <- m_prcomp(MTBLS579)
print(MTBLS_mpca)
```

raw_GCxGC-class *Subclass raw_GCxGC*

Description

Subclass *raw_GCxGC* are contained in *GCxGC* super class. It contains a dedicated slot to storage the folded bidimensional chromatogram.

Details

In the first creation of a *raw_GCxGC* object, the slot for the chromatogram. To read and fold the chromatogram use the function `read_chrom`.

Slots

`chromatogram` a numeric matrix.

`time` a vector of to elements with the range of the first dimenstion retention time

read_chrom *Read bidimensional total ion current chromatogram.*

Description

'read_GCxGC' returns a *raw_GCxGC* with the sample name, the modulation time and the bidimensional chromatogram.

Usage

```
read_chrom(name, mod_time, sam_rate, per_eval = 0.1)
```

Arguments

`name`	A name of the netCDF file to which the data will be retrieved.
`mod_time`	The modulation time of the chromatographic run.
`sam_rate`	the sampling rate of the equipment. If sam_rate is missing, the sampling rate is calculated by the dividing one by the diference of two adjacent scan time.
`per_eval`	An integer with the percentage of the run time to be evaluate, if the sampling rate is homogeneous.

Details

This function reads the netCDF file and retrieve the values in the *total_intensity* variable. Then, with the provided sampling rate and modulation time, it is folded into a numerical matrix (bidimensional chromatogram). This function is an adaptation of the presented routine from Skov and Bro (2008).

For unusual retention times, more than 60 seconds, the chemical equipment converts the measured points per minute in function of sampling rate.

References

Skov T, Bro R (2008). "Solving fundamental problems in chromatographic analysis." *Analytical and Bioanalytical Chemistry*, **390**(1), 281–285. ISSN 16182642, doi: 10.1007/s002160071618z.

Examples

```
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
GB08 <- read_chrom(GB08_fl, 5L)
```

`scores` *Method plot_scores*

Description

'scores' exports the scores matrix of the previously MPCA performed.

Usage

```
scores(Object)

## S4 method for signature 'MPCA'
scores(Object)
```

Arguments

`Object`	a MPCA object

Details

This function takes the scores of MPCA and retrieves the score matrix.

Examples

```
data(MTBLS579)
# MPCA with mean-centered and scaled data
MTBLS579_mpca <- m_prcomp(MTBLS579, center = TRUE, scale = TRUE)
# Export scores matrix
scores(MTBLS579_mpca)
```

twod_cow *Two-dimensional correlation optimized warping alignment*

Description

'twod_cow' returns the bidimensional correlation optimised warping aligmened sample chromatogram versus a reference chromatogram.

Usage

```
twod_cow(sample_chrom, ref_chrom, segments, max_warp)
```

Arguments

sample_chrom	A GCxGC class cromatogram obtained by read_chrom function.
ref_chrom	A representative GCxGC chromatogram choosen to be the template which sample_chrom will be aligned.
segments	A two integer vector with number of segments which the first and second dimension will be subdivided, respectively.
max_warp	A two integer vector with the maximum warping parameter.

Details

This is an adaptation of bidimesional COW aligment first implementated in MATLAB (Tomasi et al. 2004). This functions takes a sample chromatogram to be aligned to reference one. The argument [segment] will be used to split the whole chromatogram in n parts in the first and the second dimension respectevily. The [max_warp] argument provides de maxium tolerace of the signal transformation as well to the firs and the second dimension (Dabao Zhang et al. 2008).

References

Dabao Zhang, Xiaodong Huang, Fred E. Regnier, Min Zhang (2008). "Two-dimensional correlation optimized warping algorithm for aligning GCxGC-MS data." *Analytical Chemistry*, **80**(8), 2664–2671. ISSN 00032700, doi: 10.1021/ac7024317.

Tomasi G, Van Den Berg F, Andersson C (2004). "Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data." *Journal of Chemometrics*, **18**(5), 231–241. ISSN 08869383, doi: 10.1002/cem.859.

Examples

```
library(colorRamps)
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
GB09_fl <- system.file("extdata", "09GB.cdf", package = "RGCxGC")
GB08 <- read_chrom(GB08_fl, 5L)
GB09 <- read_chrom(GB09_fl, 5L)
```

```
GB09_al <- twod_cow(GB09, GB08, c(20, 40), c(2, 8))
```

---

wsmooth *Bidimensional smoothing*

Description

'wsmooth' provides a bidimensional weighted whittaker smoothing of a two dimensional chromatogram

Usage

```
wsmooth(chromatogram, penalty = 1, lambda = 1)
```

Arguments

chromatogram	*raw_GCxGC* or *preproc_GCxGC* object like with provided *name* and *mod_time* slots.
penalty	an integer of the order of the penalty. Only penalty of first (penalty = 1) and second order (penalty = 2) are allowed. By default it is performed with first penalty order.
lambda	smoothing parameter: larger values lead to more smoothing.

Details

This function takes a raw bidimensional chromatogram and performs the weighted wittaker smoothing. It smooths with linear or cuadratic penalty along side the first dimension, based on Whittaker smoother (Eilers 2003).

References

Eilers PH (2003). "A perfect smoother." *Analytical Chemistry*, **75**(14), 3631–3636. ISSN 00032700, doi: 10.1021/ac034173t.

Examples

```
chrom_name <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
chrom_2D <- read_chrom(chrom_name, 5L)
chrom_smooth <- wsmooth(chrom_2D, penalty = 1, lambda = 1e1)
plot(chrom_smooth, nlevels = 150,
          color.palette = colorRamps::matlab.like,
          main = expression(paste(lambda, "= 10, penalty = 1")) )
```

Index