Package 'RGCxGC'

December 14, 2022

Type Package

Title Preprocessing and Multivariate Analysis of Bidimensional Gas Chromatography Data

Version 1.2.0

Date 2022-12-13

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

Description Toolbox for chemometrics analysis of bidimensional gas chromatography data. This package import 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

URL https://github.com/DanielQuiroz97/RGCxGC, https://danielquiroz97.github.io/RGCxGC/

BugReports https://github.com/DanielQuiroz97/RGCxGC/issues

Imports colorRamps, Rdpack, stats, methods, graphics

RdMacros Rdpack

biocViews

Depends R (>= 4.2.0), RNetCDF, ptw

LazyData true

RoxygenNote 7.2.3

Encoding UTF-8

Suggests knitr, rmarkdown, lattice, prettydoc,

VignetteBuilder knitr

NeedsCompilation no

Author Cristian Quiroz-Moreno [aut, cre]
(<https://orcid.org/0000-0002-9069-9147>),
Guilherme L. Alexandrino [aut]
(<https://orcid.org/0000-0002-2007-378X>),
Noroska G.S. Mogollón [aut] (<https://orcid.org/0000-0002-0756-6184>)

Repository CRAN

Date/Publication 2022-12-14 11:30:02 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, but 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 *Two-dimensional baseline correction*

Description

'baseline_corr' provides a two-dimensional baseline correction by using the asymmetric least squares algorithm.

Usage

```
baseline_corr(chromatogram, ...)
```

Arguments

chromatogram	a *raw_GCxGC* object.
...	other parameters passed to baseline.corr function in the pwt package.

Details

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

References

Eilers PH (2004). "Parametric Time Warping." *Analytical Chemistry*, **76**(2), 404–411.

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' perform two-dimensional correlation optimized warping alignment in batch.

Usage

```
batch_2DCOW(reference, sample_chroms, segments, max_warp, add_ref = FALSE)
```

Arguments

reference	a GCxGC chromatogram which will be taken as the reference chromatogram.
sample_chroms	a named list with the sample chromatograms which will be aligned against to the reference chromatogram.
segments	a two integer vector with the number of segments which the first and second dimension will be divided, respectively.
max_warp	a two integer vector with the maximum warping parameter for the first and second dimension
add_ref	a logical indicating if the reference chromatogram will be joined together with the sample chromatograms. By the fault add_ref = F. If add_ref is set to T, the provide reference chromatogram will be included as another sample chromatogram in the downstream analysis.

Details

The first argument is the reference chromatogram which other chromatograms will aligned against to. Then, a named list is needed for the sample_chroms argument. Each chromatogram in this list will be aligned using the reference chromatogram. By default, the reference chromatogram will be not included in the subsequent analysis, such as MPCA. If you would like to add the reference chromatogram, then add_ref = T.

This is an adaptation of two-dimensional COW alignment, firstly implemented in MATLAB. This function takes a set of samples chromatogram to be aligned against to the reference. The argument [segment] will be used to split the whole chromatogram in *n* and *m* parts in the first and the second dimension, respectively. The [max_warp] argument provides the maximum tolerance of the signal transformation for the first and the second dimension, respectively.

Examples

```
# Read Sample chromatogram
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
MTBLS08 <- read_chrom(GB08_fl, mod_time = 5)

# Read reference chromatogram
GB09_fl <- system.file("extdata", "09GB.cdf", package = "RGCxGC")
MTBLS09 <- read_chrom(GB09_fl, mod_time = 5)

# Create a named list
# MTBLS08 will be repeated for exemplification
# Considerer that chromatograms are renamed considering the list names
batch_samples <- list(Chrom1 = MTBLS08, Chrom2 = MTBLS08)


# Perform batch 2DCOW alignment
# Add the reference chromatogram as another sample
batch_alignment <- batch_2DCOW(MTBLS09, batch_samples,
                               c(10, 40), c(1, 10), add_ref = TRUE)
# Exclude the reference chromatogram in the sample chromatogram set
batch_alignment <- batch_2DCOW(MTBLS09, batch_samples, 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 the reference chromatogram.

Details

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

---

dephase_chrom *Method dephase_chrom*

Description

'dephase_chrom' shifts the retention time in the second dimension of the two-dimensional chromatogram. This procedure is usually applied in cases when part of peaks is splited in at the final and beginning of the second dimension. Also, the solvent effect and column bleeding can be removing by dephasing the chromatogram. The dephasing procedure is performing by splitting the chromagram with the relative value provided.

Usage

```
dephase_chrom(Object, rel_dephase)

## S4 method for signature 'GCxGC'
dephase_chrom(Object, rel_dephase)
```

Arguments

Object	a GCxGC class object
rel_dephase	a numeric value from 0 to 100 with the relative dephasing position.

Examples

```
library(colorRamps)
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
GB08 <- read_chrom(GB08_fl, 5L)
plot(GB08, nlevels = 150, color.palette = matlab.like,
     main = "No dephased chromatogram")
GB08_d25 <- dephase_chrom(GB08, 25)
plot(GB08_d25, nlevels = 150, color.palette = matlab.like,
     main = "25% dephased chromatogram")
```

GCxGC-class *Class GCxGC*

Description

Class *GCxGC* defines the superclass of two-dimensional comprehensive gas chromatographic data.

Details

The validity function evaluates if the provided file can be readed as a NetCDF file. The validation function employs the function open.nc to check if the provided file inherits to the NetCDF class.

Slots

name	the name of a NetCDF file where the data will be retrieved from.
mod_time	a integer with the modulation time for the second dimension.

`get_metadata` *Method get_metadata*

Description

'get_metadata' retrieves the metadata contained in a joined_chrom object.

Usage

```
get_metadata(chroms)

## S4 method for signature 'joined_chrom'
get_metadata(chroms)
```

Arguments

`chroms`	a joined_chrom object created by joined_chrom function.

Details

This function accesses to the *groups* slot created by the joined_chrom function. The *Names* are the names of the chromatograms.

Examples

```
data(Myrothecium)
myr_data <- get_metadata(Myrothecium)
myr_data
```

`joined_chrom-class` *Class joined_chrom*

Description

Class *joined_chrom* defines the superclass to gather single chromatogram, as well as batch chromatograms into a single list, prior to multiway principal compoment analysis or unfolding them.

Slots

`chromatograms` a named list with all chromatograms.

`time` the time range of the chromatographic run

`groups` a data.frame containing the experiment metadata with a column named as *Names*.

`mod_time` modulation time of the second dimension

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

Description

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

Usage

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

Arguments

x, y	a GCxGC object, either single or batch chromatograms.
groups	a data.frame containing the metadata. It must have a column named as *Names* to merge with the imported chromatograms.
...	other GCxGC objects 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)
metadata <- data.frame(Names = c("GB08", "GB09"),
                       Type = c("Control", "Treatment"))
join_metadata <- join_chromatograms(GB08, GB09, groups = metadata)
```

make_loadings *Import foreign model loadings*

Description

'make_loading' method takes the loading matrix obtained by a mixOmixs package and fold them into two-dimensional matrix

Usage

```
make_loadings(floadings, time, mod_time, acq_rate)
```

Arguments

`floadings`	a numeric matrix with foreign loadings. With variables in columns and eigenvalues as rows.
`time`	a vector of length two with the time range of the chromatographic run
`mod_time`	the modulation time of the second dimension.
`acq_rate`	the acquisition rate of the mass analyzer. The acquisition rate is printed when read_chrom function is performed

Details

We strongly recommend to use the plsda function in the mixOmics package to perform partial least squares-discriminant analysis. The result of this model is a list containing a loading matrix. The method retrieves a matrix A with *m* and *n* dimensions. Where *m* is the eigenvalues and *n* is the number of loadings which the model returns.

`MPCA-class` *Subclass MPCA subclass for Multiway Principal Component Analysis object*

Description

Subclass MPCA subclass for Multiway Principal Component Analysis object

Slots

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

`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 following native name 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 the large size of chromatograms, these 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`.

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.

Myrothecium *Microbial metabolism kinetics of* Myrothecium sp.

Description

This object contains six chromatograms of the microbial metabolic kinetics. *Myrothecium sp.* were cultured in CMA. Inoculation was made from Petri dishes with the fully-grown fungal cells and sterile distilled water (to wash the surface of the plate). The plate was scraped with a sterile glass handle to obtain the spore suspension. The suspension was liquated and the concentration of 2.4 x 105 spores/mL was determined using a Neubauer chamber and an optical microscope. Then, 50 uL of the suspension was inoculated in a flow chamber into the tubes containing the culture medium. Tubes were kept at 25 celsius degree in a growth chamber with 12h of photoperiod.

A solid-phase microextraction (SPME) assay containing a DVB / CAR / PDMS (Divinylbenzene / Carboxene / Polymethylsiloxane 50/30 mm) fiber was placed into the tube headspace.

A set of columns consisting of HP-5MS 30m x 0.25mm x 0.25 um connected to a Supelcowax 1m x 0.10mm × 0.10um with a 1m x 0.25mm deactivated silica capillary being allocated between them. In these tests, a modulation period of 5s was used.

For GCxGC-QMS data acquisition, GCMSsolution version 5.3 software was used. The temperature program were 60-165 celcius at 3 celcius/min; 165 celcius - 260celcius at 20 celcius/min; 260 celcius (5 min); flow rate was 0.6 mL/min (Helium 5.0 carrier gas); splitless injection mode, ion source temperature 200 celcius, interface temperature 260 celcius; voltage 0,9 kV; mass range 50-380 m/z; acquisition rate 25Hz and electron ionization (70eV).

The original study was developed by Quiroz-Moreno et al. (2020).

Usage

```
data(Myrothecium)
```

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

References

Quiroz-Moreno C, Furlan MF, Belinato JR, Augusto F, Alexandrino GL, Mogollón NGS (2020). "RGCxGC toolbox: An R-package for data processing in comprehensive two-dimensional gas chromatography-mass spectrometry." *Microchemical Journal*, **156**, 104830.

`m_prcomp` *Multiway Principal Component Analysis*

Description

'm_prcomp' Performs a multiway principal components analysis on a given two-dimensional chromatograms and returns the results 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 = FALSE, scale = FALSE, npcs = 3, ...)
```

Arguments

`chrom`	Multiple chromatograms read or batch aligned
`center`	A logical value indicating whether the variables should be shifted to be zero centered. FALSE is set by default.
`scale`	a logical value indicating whether the variables should be scaled to have unit variance before the analysis takes place. The default is FALSE.
`npcs`	an integer indicating how many principals components are desired to maintain. The default is 3 principal components.
`...`	Other arguments passed to `prcomp` function.

Value

MPCA returns a list whit class "MPCA" containing the summary of the analysis, the scores matrix, unfolded loadings, and the metadata if it was provided 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.

Examples

```
data(MTBLS579)

# Perform multiway principal component analysis
MTBLS579_mpca <- m_prcomp(MTBLS579, center = TRUE)
# Print MPCA summary
print(MTBLS579_mpca)
# Retrieve MPCA scores
scores(MTBLS579_mpca)
# Plot bidimensional scores
plot_loading(MTBLS579_mpca)
```

`plot` *Method plot*

Description

'plot' plot the two-dimensional chromatogram as a contour plot.

Usage

```
plot(Object, type = "f", ...)

## S4 method for signature 'GCxGC'
plot(Object, type = "f", ...)
```

Arguments

`Object`	a GCxGC chromatogram, it can be a raw, or preprocessed chromatogram.
`type`	a character indicating the type of chromatogram representation. By default, type = "f" for `filled.contour` function, if type = "c" only contours or isolines will be displayed by using the `contour` function.
`...`	Other parameters passed to `filled.contour` or `filled.contour` function, it depends on the value of the *type* argument.

Details

This plot function employs the built-in contour function. As mentioned in Reichenbach et al. (2004), interpolation is used to 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.

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(chrom_2D, type = "c", nlevels = 50, col = matlab.like(30))
```

---

plot_loading *Plot two-dimensional MPCA loadings*

Description

'plot_loading' plot the loading of the a MPCA object.

Usage

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

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

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 principal component to plot.
`thresh`	numerical value. A threshold to remove low loading values.
`...`	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 it fills the removed variables with zero. Then, the loadings are plotted considering one principle component at a time as a two-dimensional chromatogram.

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)
```

PLSDA-class *Subclass PLSDA*

Description

Class *PLSDA* defines the class to import foreign results of partial least squares-discriminant analysis performed with mixOmics package.

---

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 two-dimensional chromatogram.

Details

After reading a two-dimensional chromatogram, you can perform serveral preprocessing techniques such as smoothing or baseline correction. This action will create an object of a preproc_GCxGC subclass.

print_mpca *Print MPCA summary*

Description

'print_mpca' call the MPCA object to print the summary of this analysis.

Usage

```
print_mpca(Object)

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

Arguments

Object	a MPCA object

Details

The plot function employs the built-in print function and a precomputed MPCA summary to display the explained and cumulative variance for each principal component.

Examples

```
data(MTBLS579)
MTBLS_mpca <- m_prcomp(MTBLS579, center = TRUE)
print_mpca(MTBLS_mpca)
```

projected-class *Class projected*

Description

The *projected* class defines the superclass for projection methods, specially for multiway principal component analysis and discriminant analysis based on partial least squares. The class represents the convergence of in-package results (m_prcomp) and the foreing building model (PLS-DA) procedure.

Slots

`loadings` The eigenvectors of each principal component.

`time` The time range of chromatographic run

`mod_time` modulation time of the second dimension

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 two-dimensional chromatogram.

Details

In the first creation of a *raw_GCxGC* object, the slot for the chromatogram is not created yet. To read and fold the chromatogram use the `read_chrom` function.

Slots

`chromatogram` a numeric matrix.

`time` a vector of two elements with the range of the first dimenstion run time

read_chrom *Read two-dimensional chromatogram*

Description

'read_GCxGC' returns a *raw_GCxGC* with the sample name, the modulation time, the chromatographic time range and the two-dimensional chromatogram.

Usage

```
read_chrom(
  name,
  mod_time,
  sam_rate,
  per_eval = 0.1,
  x_cut = NULL,
  y_cut = NULL,
  verbose = TRUE
)
```

Arguments

`name`	a name of the NetCDF file where the data is allocated.
`mod_time`	a integer, the modulation time of the chromatographic run.
`sam_rate`	a integer, the sampling rate of the equipment. If sam_rate is missing, the sampling rate is calculated by the dividing 1 by the difference of two adjacent scan time.

per_eval	a double between 0 and 1, with the percentage of the run time records to be evaluated if the sampling rate is homogeneous.
x_cut	a vector with two elements representing the retention time range to be mantained in the first dimension.
y_cut	a vector with two elements representing the retention time range to be mantained in the second dimension.
verbose	a logical indicating if the information of chromatogram is printted in the console.

Details

This function reads the NetCDF file and retrieves values in the *total_intensity* array. Then, with the provided sampling rate and modulation time, the chromatogram is folded into a numerical matrix, representing the two-dimensional chromatogram. This function is an adaptation of the presented routine by Skov and Bro (2008).

References

Skov T, Bro R (2008). "Solving fundamental problems in chromatographic analysis." *Analytical and Bioanalytical Chemistry*, **390**(1), 281–285.

Examples

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

---

reference_chrom *Make reference chromatogram*

Description

'reference_chrom' makes a reference chromatogram by calculating a statistic of multiple chromatograms.

Usage

```
reference_chrom(chromatograms, stat = "mean")
```

Arguments

chromatograms	a joined_chrom object.
stat	a character with the name of the mathematical function that pixels will be subjected to. By default, (stat = "mean") the new reference chromatogram will be the result of the provided mathematical function.

Details

The aim of this function is to create a consensus chromatogram to be used as a reference in the peak alignment process. In other words, multiple chromatograms will be subjected to a mathematical function, such as min, max, or mean in order to create a representative chromatogram. Then, the new chromatogram will be used as a template and the other chromatograms will be aligned against it. This function overlap pixels with the same chromatogram index and computes a desired mathematical function for each pixel.

Examples

```
# Read chromatogram 1
GB08_fl <- system.file("extdata", "08GB.cdf", package = "RGCxGC")
MTBLS08 <- read_chrom(GB08_fl, mod_time = 5)

# Read chromatogram 2
GB09_fl <- system.file("extdata", "09GB.cdf", package = "RGCxGC")
MTBLS09 <- read_chrom(GB09_fl, mod_time = 5)

# Join chromatograms
joined <-  join_chromatograms(MTBLS08, MTBLS09)
reference <- reference_chrom(joined, stat = "mean")
plot(reference, main = "Reference chromaogram")
```

---

scores *Method plot_scores*

Description

'scores' exports the scores matrix of a MPCA object.

Usage

```
scores(Object)

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

Arguments

Object	a MPCA object

Details

This function takes the whole MPCA object and retrieves the score matrix.

Examples

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

`set_metadata` *Set the metadata for a joined_chrom*

Description

'set_metadata' fill metadata slot of a joined_chrom object.

Usage

```
set_metadata(Object, metadata)

## S4 method for signature 'joined_chrom,data.frame'
set_metadata(Object, metadata)
```

Arguments

`Object`	a joined_chrom object
`metadata`	a data.frame containing the metadata. It must have a column named as *Names* to merge with the chromatograms.

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)
extra_info <- data.frame(Names = c("GB08", "GB09"),
                         Type = c("Control", "Treatment"))
join_chrom <- join_chromatograms(GB08, GB09)
join_metadata <- set_metadata(join_chrom, metadata = extra_info)
```

twod_cow *Two-dimensional correlation optimized warping alignment*

Description

This is an adaptation of two-dimesional COW alignment, first implemented in MATLAB (Tomasi et al. 2004). This functions takes a sample chromatogram to be aligned against a reference. The argument [segment] will be used to split the whole chromatogram in *n* and *m* parts the first and the second dimension, respectively. The [max_warp] argument provides de maximum tolerance of the signal transformation for the first and the second dimension (Dabao Zhang et al. 2008).

Usage

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

Arguments

sample_chrom	A GCxGC class chromatogram imported by read_chrom function or a preprocessed chromatogram.
ref_chrom	A representative GCxGC chromatogram chosen 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 divided, respectively.
max_warp	A two integer vector with the maximum warping parameter.

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.

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.

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)

GB09_al <- twod_cow(sample_chrom = GB09, ref_chrom = GB08,
                    segments = c(20, 40), max_warp = c(2, 8))
```

unfold_chrom *Unfold two-dimensional chromatograms*

Description

'unfold' converts the two-dimensional chromatograms into a one dimensional vector. Then, all chromatograms are joined into a matrix.

Usage

```
unfold_chrom(Object)
```

Arguments

Object	a batch_2DCOW or joined_chrom objects.

Details

This function takes a single argument, batch_2DCOW or joined_chrom objects and extracts each chromatogram and then it is unfolded into a one-dimensional vector. Then, each one dimensional vector is joined in a single matrix, where each row represent an observation or a chromatogram and each column represent a variable, in our case, each retention time. Also, in order to keep information about chromatographic runs, the retention times for both dimensions are also exported.

Examples

```
data(Myrothecium)
# Unfold 2D chromatogram
chrom_1D <- unfold_chrom(Myrothecium)
# Retrieve retention time for the first dimension
time_1D <- chrom_1D$time
# Retrieve the modulation time
modulation <- chrom_1D$mod_time
```

wsmooth *Two-dimensional smoothing*

Description

'wsmooth' weighted whittaker smoothing.

Usage

```
wsmooth(chromatogram, penalty = 1, lambda = 1, min_int = 0)
```

Arguments

chromatogram	*raw_GCxGC* or *preproc_GCxGC* object with *name* and *mod_time* slots.
penalty	an integer of the order of the penalty. Only penalty of first (penalty = 1) and second (penalty = 2) order are allowed. By default, the smooth function is performed with first penalty order.
lambda	smoothing parameter: larger values lead to more smoothing.
min_int	minimum intensity value. The smoothing routine usually creates low intensity artifacts which can obscure other compounds signals. The min intensity value replace signals bellow the given value with 0. For quadrupole mass detector this artifacts may range from 0-100, while for TOF mass analyzers it can be 0-1e3.

Details

This function takes a raw two-dimensional chromatogram and performs the weighted wittaker smoothing. It smooths the signal with linear or quadratic penalty, depending on the provided 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.

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")) )
# Remove intensities bellow 1.75e5 (too high)
chrom_smooth2 <- wsmooth(chrom_2D, penalty = 1,
                         lambda = 1e1, min_int = 1.75e5)
plot(chrom_smooth2, nlevels = 150,
           color.palette = colorRamps::matlab.like,
           main = expression(paste(lambda,
            "= 10, penalty = 1, min_int = 1.75e5")) )
```

Index