Package ‘gcxgclab’

January 22, 2024

Title GCxGC Preprocessing and Analysis

Version 1.0.1

Description
Provides complete detailed preprocessing of two-dimensional gas chromatogram (GCxGC) samples. Baseline correction, smoothing, peak detection, and peak alignment. Also provided are some analysis functions, such as finding extracted ion chromatograms, finding mass spectral data, targeted analysis, and nontargeted analysis with either the 'National Institute of Standards and Technology Mass Spectral Library' or with the mass data. There are also several visualization methods provided for each step of the preprocessing and analysis.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.1

Depends R (>= 4.2.0)

Imports ncdf4 (>= 1.19.0), dplyr (>= 1.0.8), ggplot2 (>= 3.3.5), ptw (>= 1.9.16), stats (>= 4.2.0), utils (>= 4.2.0), nilde (>= 1.1.6), zoo (>= 1.8.11), nls.multstart (>= 1.3.0), Rdpack (>= 2.4.0)

RdMacros Rdpack

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Stephanie Gamble [aut, cre] (<https://orcid.org/0000-0001-7209-0891>),
Mannion Joseph [ctb],
Granger Caroline [ctb],
Battelle Savannah River Alliance [cph],
NNSA, US DOE [fnd]

Maintainer Stephanie Gamble <stephanie.gamble@srnl.doe.gov>

Repository CRAN

Date/Publication 2024-01-22 14:10:06 UTC

R topics documented:

`align` *Reference Batch Align*

Description

`align` aligns peaks from samples to a reference sample's peaks.

Usage

```
align(data_list, THR = 1e+05)
```

Arguments

`data_list` a *list* object. Data extracted from each cdf file, ideally the output from extract_data().

`THR` a *float* object. Threshold for peak intensity. Should be a number between the baseline value and the highest peak intensity. Default is THR = 100000.

Details

This function aligns the peaks from any number of samples. Peaks are aligned to the retention times of the first peak. If aligning to a reference or standard sample, this should be the first in the lists for data frames and for the mass data. The function comp_peaks() is used to find the corresponding peaks. This function will return a new list of TIC data frames and a list of mass data. The first sample's data is unchanged, used as the reference. Then a TIC data frame and mass data for each of the given samples containing the peaks and time coordinates of the aligned peaks. The time coordinates are aligned to the first sample's peaks, the peak height and MS is unchanged.

Value

A *list* object. List of aligned data from each cdf file and a list of peaks that were aligned for each file.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
file2 <- system.file("extdata","sample2.cdf",package="gcxgclab")
file3 <- system.file("extdata","sample3.cdf",package="gcxgclab")
frame1 <- extract_data(file1,mod_t=.5)
frame2 <- extract_data(file2,mod_t=.5)
frame3 <- extract_data(file3,mod_t=.5)
aligned <- align(list(frame1,frame2,frame3))
plot_peak(aligned$Peaks$S1,aligned$S1,title="Reference Sample 1")
plot_peak(aligned$Peaks$S2,aligned$S2,title="Aligned Sample 2")
plot_peak(aligned$Peaks$S3,aligned$S3,title="Aligned Sample 3")
```

`batch_eic` *Finds batch of EICs*

Description

`batch_eic` calculates the mass defect for each ion, then finds each listed EICs of interest.

Usage

```
batch_eic(data, MOIs, tolerance = 5e-04)
```

Arguments

data	a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
MOIs	a *vector* object. A vector containing a list of all masses of interest to be investigated.
tolerance	a *double* object. The tolerance allowed for the MOI. Default is 0.0005.

Details

Extracted Ion Chromatogram (EIC) is a plot of intensity at a chosen m/z value, or range of values, as a function of retention time. This function uses find_eic() to find intensity values at the given mass-to-charge (m/z) values, MOIs, and in a range around MOI given a tolerance. Calculates the mass defect for each ion, then finds the specific EICs of interest. Returns a data frame of time values, mass values, intensity values,and mass defects.

Value

eic_list, *list* object, containing *data.frame* objects. Data frames of time values, mass values, intensity values, and mass defects for each MOI listed in the input csv or txt file.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
mois <- c(92.1397, 93.07058)
eics <- batch_eic(frame, MOIs=mois ,tolerance = 0.005)
for (i in 1:length(eics)){
   print(plot_eic(eics[[i]], title=paste("EIC for MOI",mois[i])))
   print(plot_eic(eics[[i]], title=paste("EIC for MOI",mois[i]), dim=2))
}
```

batch_ms *Finds batch of mass spectra*

Description

batch_ms Finds batch of mass spectra of peaks.

Usage

```
batch_ms(data, t_peaks, tolerance = 5e-04)
```

Arguments

data	a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
t_peaks	a *vector* object. A list of times at which the peaks of interest are located in the overall time index for the sample.
tolerance	a *double* object. The tolerance allowed for the time index. Default is 0.0005.

Details

This function uses find_ms() to find the mass spectra values of a batch list of peaks in intensity values of a GCxGC sample at overall time index values specified in a txt or csv file. It outputs a list of data frames, for each peak, of the mass values and percent intensity values which can then be plotted to product the mass spectra plot.

Value

A *list* object of *data.frame* objects. Each a data frame of the mass values and the percent intensity values.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
mzs <- batch_ms(frame, t_peaks = peaks$'T'[1:5])
for (i in 1:length(mzs)){
   print(plot_ms(mzs[[i]], title=paste('Mass Spectrum of peak', i)))
}
```

---

batch_preprocess *Batch reprocessing*

Description

`batch_preprocess` performs full preprocessing on a batch of data files.

Usage

```
batch_preprocess(
  path = ".",
  mod_t = 10,
  shift = 0,
  lambda = 20,
  gamma = 0.5,
  subtract = NULL,
  THR = 10^5,
  images = FALSE
)
```

Arguments

`path`	a *string* object. The path to the directory containing the cdf files to be batch preprocessed and aligned.
`mod_t`	a *float* object. The modulation time for the GCxGC sample analysis. Default is 10.

`shift` a *float* object. The number of seconds to shift the phase by. Default is 0 to skip shifting.

`lambda` a *float* object. A number (parameter in Whittaker smoothing), suggested between 1 to 10^5. Small lambda is very little smoothing, large lambda is very smooth. Default is lambda = 20.

`gamma` a *float* object. Correction factor between 0 and 1. 0 results in almost no values being subtracted to the baseline, 1 results in almost everything except the peaks to be subtracted to the baseline. Default is 0.5.

`subtract` a *data.frame* object. Data frame containing TIC data from a background sample or blank sample to be subtracted from the sample TIC data.

`THR` a *float* object. Threshold for peak intensity for peak alignment. Should be a number between the baseline value and the highest peak intensity. Default is THR = 100000.

`images` a *boolean* object. An optional input. If TRUE, all images of preprocessing steps will be displayed. Default is FALSE, no images will be displayed.

Details

This function performs full preprocessing on a batch of data files. Extracts data and performs peak alignment and performs smoothing and baseline correction.

Value

A *data.frame* object. A list of pairs of data frames. A TIC data frame and an MS data frame for each file.

Examples

```
folder <- system.file("extdata",package="gcxgclab")
frame_list <- batch_preprocess(folder,mod_t=.5,lambda=10,gamma=0.5,images=TRUE)
```

`bl_corr` *Baseline correction*

Description

`bl_corr` performs baseline correction of the intensity values.

Usage

```
bl_corr(data, gamma = 0.5, subtract = NULL)
```

Arguments

- `data` a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
- `gamma` a *float* object. Correction factor between 0 and 1. 0 results in almost no values being subtracted to the baseline, 1 results in almost everything except the peaks to be subtracted to the baseline. Default is 0.5.
- `subtract` a *list* object. Data extracted from a cdf file, ideally the output from extract_data().

Details

This function performs baseline correction and baseline subtraction for TIC values.

Value

A *data.frame* object. A data frame of the overall time index, the x-axis retention time, the y-axis retention time, and the baseline corrected total intensity values.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
sm_frame <- smooth(frame, lambda=10)
blc_frame <- bl_corr(sm_frame, gamma=0.5)
plot_chr(blc_frame, title='Baseline Corrected')
```

`comp_nist` *Compares MS to NIST MS database*

Description

`comp_nist` compares the MS data from a peak to the NIST MS database.

Usage

```
comp_nist(nistlist, ms, cutoff = 50, title = "Best NIST match")
```

Arguments

- `nistlist` a *list* object, a list of compound MS data from the NIST MS Library database, ideally the output of nist_list().
- `ms` a *data.frame* object, a data frame of the mass values and the percent intensity values, ideally the output of find_ms().
- `cutoff` a *float* object, the low end cutoff for the MS data, determined based on the MS devices used for analysis. Default is 50.
- `title` a *string* object. Title placed at the top of the head-to-tail plot of best NIST Library match. Default title "Best NIST match".

Details

This function takes the MS data from an intensity peak in a sample and compares it to the NIST MS Library database and determines the compound which is the best match to the MS data.

Value

a *data.frame* object, a list of the top 10 best matching compounds from the NIST database, with their compounds, the index in the nistlist, and match percent.

`comp_peaks` *Compare Peaks*

Description

`comp_peaks` compares peaks of two samples.

Usage

```
comp_peaks(ref_peaks, al_peaks)
```

Arguments

`ref_peaks`	a *data.frame* object. A data frame with 4 columns (Time, X, Y, Peak), ideally the output from either top_peaks() or thr_peaks().
`al_peaks`	a *data.frame* object. A data frame with 4 columns (Time, X, Y, Peak), ideally the output from either top_peaks() or thr_peaks().

Details

This function find compares the peaks from two samples and correlates the peaks by determining the peaks closest to each other in the two samples, within a certain reasonable distance. Then returns a data frame with a list of the correlated peaks including each of their time coordinates.

Value

A *data.frame* object. A data frame with 8 columns containing the matched peaks from the two samples, with the time, x, y, and peak values for each.

`extract_data` *Extracts data from cdf file.*

Description

`extract_data` Extracts the data from a cdf file.

Usage

```
extract_data(filename, mod_t = 10, shift_time = TRUE)
```

Arguments

`filename`	a *string* object. The path or file name of the cdf file to be opened.
`mod_t`	a *float* object. The modulation time for the GCxGC sample analysis. Default is 10.
`shift_time`	a *boolean* object. Determines whether the Overall Time Index should be shifted to 0. Default is TRUE.

Details

This function opens the specified cdf file using the implemented function `nc_open` from ncdf4 package, then extracts the data and closes the cdf file using the implemented function `nc_close` from ncdf4 package (Pierce 2021). It then returns a list of two data frames. The first is a dataframe of the TIC data, the output of create_df(). The second is a data frame of the full MS data, the output of mass_data().

Value

A *list* object. A list of the extracted data: scan acquisition time, total intensity, mass values, intensity values, and point count.

References

Pierce D (2021). “Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files.” *CRAN*. `https://cirrus.ucsd.edu/~pierce/ncdf/index.html`.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
plot_chr(frame, title='Raw Data', scale="linear")
plot_chr(frame, title='Log Intensity')
```

find_eic *Finds EICs*

Description

find_eic calculates the mass defect for each ion, then finds the specific EICs of interest.

Usage

```
find_eic(data, MOI, tolerance = 5e-04)
```

Arguments

data	a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
MOI	a *float* object. The mass (m/z) value of interest.
tolerance	a *double* object. The tolerance allowed for the MOI. Default is 0.0005.

Details

Extracted Ion Chromatogram (EIC) is a plot of intensity at a chosen m/z value, or range of values, as a function of retention time. This function finds intensity values at the given mass-to-charge (m/z) values, MOI, and in a range around MOI given a tolerance. Calculates the mass defect for each ion, then finds the specific EICs of interest. Returns a data frame of time values, mass values, intensity values, and mass defects.

Value

eic, a *data.frame* object. A data frame of time values, retention time 1, retention time 2, mass values, intensity values, and mass defects.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
eic <- find_eic(frame, MOI=92.1397,tolerance=0.005)
plot_eic(eic,dim=1,title='EIC for MOI 92.1397')
plot_eic(eic,dim=2,title='EIC for MOI 92.1397')
```

find_ms *Finds MS*

Description

`find_ms` Finds mass spectra of a peak.

Usage

```
find_ms(data, t_peak, tolerance = 5e-04)
```

Arguments

`data`	a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
`t_peak`	a *float* object. The overall time index value for when the peak occurs in the GCxGC sample (the 1D time value).
`tolerance`	a *double* object. The tolerance allowed for the time index. Default is 0.0005.

Details

This function finds the mass spectra values of a peak in the intensity values of a GCxGC sample at a specified overall time index value. Then outputs a data frame of the mass values and percent intensity values which can then be plotted to product the mass spectra plot.

Value

A *data.frame* object. A data frame of the mass values and the percent intensity values.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
mz <- find_ms(frame, t_peak=peaks$'T'[1])
plot_ms(mz)
plot_defect(mz,title="Kendrick Mass Defect, CH_2")
```

gauss *1D Gaussian function*

Description

`gauss` Defines the 1D Gaussian curve function.

Usage

```
gauss(a, b, c, t)
```

Arguments

`a, b, c`	are *float* objects. Parameters in R^1 for the Gaussian function.
`t`	a *float* object. The independent variable in R^1 for the Gaussian function.

Details

This function defines a 1D Gaussian curve function.

Value

A *float* object. The value of the Gaussian function at time t, given the parameters input a,b,c.

gauss2 *2D Gaussian function*

Description

`gauss2` Defines the 2D Gaussian curve function.

Usage

```
gauss2(a, b1, b2, c1, c2, t1, t2)
```

Arguments

`a, b1, b2, c1, c2`	are *float* objects. Parameters in R^1 for the Gaussian function.
`t1, t2`	are *float* objects. The independent variables t=(t1.t2) in R^2 for the Gaussian function.

Details

This function defines a 2D Gaussian curve function.

Value

A *float* object. The value of the Gaussian function at time t=(t1,t2) given the parameters input a,b1,b2,c1,c2.

gauss2_fit *Fitting to 2D Gaussian curve*

Description

gauss2_fit fits data around a peak to a 2D Gaussian curve.

Usage

```
gauss2_fit(TIC_df, peakcoord)
```

Arguments

TIC_df	a *data.frame* object. Data frame with 4 columns (Overall Time Index, RT1, RT2, TIC), ideally the output from create_df(), or the first data frame returned from extract_data(), $TIC_df.
peakcoord	a *vector* object. The two dimensional time retention coordinates of the peak of interest. c(RT1,RT2).

Details

This function fits data around the specified peak to a 2D Gaussian curve, minimized with nonlinear least squares method nls() from "stats" package.

Value

A *list* object with three items. The first *data.frame* object. A data frame with three columns, (time1, time2, guassfit), the time values around the peak, and the intensity values fitted to the optimal Gaussian curve. Second, a *vector* object of the fitted parameters (a,b1,b2,c1,c2). Third, a *double* object, the volume under the fitted Gaussian curve.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
gaussfit2 <- gauss2_fit(frame$TIC_df, peakcoord=c(peaks$'X'[1], peaks$'Y'[1]))
message(paste('Volume under curve =',gaussfit2[[3]],'u^3'))
plot_gauss2(frame$TIC_df, gaussfit2[[1]])
```

gauss_fit *Fitting to Gaussian curve*

Description

gauss_fit fits data around a peak to a Gaussian curve.

Usage

```
gauss_fit(TIC_df, peakcoord)
```

Arguments

TIC_df	a *data.frame* object. Data frame with 4 columns (Overall Time Index, RT1, RT2, TIC), ideally the output from create_df(), or the first data frame returned from extract_data(), $TIC_df.
peakcoord	a *vector* object. The two dimensional time retention coordinates of the peak of interest. c(RT1,RT2).

Details

This function fits data around the specified peak to a Gaussian curve, minimized with nonlinear least squares method nls() from "stats" package.

Value

A *list* object with three items. The first *data.frame* object. A data frame with two columns, (time, guassfit), the time values around the peak, and the intensity values fitted to the optimal Gaussian curve. Second, a *vector* object of the fitted parameters (a,b,c). Third, a *double* object, the area under the fitted Gaussian curve.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
gaussfit <- gauss_fit(frame$TIC_df, peakcoord=c(peaks$'X'[1], peaks$'Y'[1]))
message(paste('Area under curve =',gaussfit[[3]], 'u^2'))
plot_gauss(frame$TIC_df, gaussfit[[1]])
```

mass_list *Creates list of atomic mass data*

Description

`mass_list` creates a list of atomic mass data

Usage

```
mass_list()
```

Details

This function creates a data frame containing the data for the atomic weights for each element in the periodic table (M. and et al. 2012).

Value

A *data.frame* object, with two columns, (elements, mass).

References

M. W, et al. (2012). "The Ame2012 atomic mass evaluation." *Chinese Phys. C*, **36 1603**.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
mz <- find_ms(frame, t_peak=peaks$'T'[1])
masslist <- mass_list()
non_targeted(masslist, mz, THR=0.05)
```

nist_list *Creates list of NIST data*

Description

`nist_list` creates a list of the data from the NIST MS database.

Usage

```
nist_list(nistfile, ...)
```

Arguments

`nistfile`	a *string* object, the file name or path of the MSP file for the NIST MS Library database.
`...`	additional optional *string* objects, the file names or paths of the MSP file for the NIST MS Library if the data base is broken into multiple files.

Details

This function takes the MSP file containing the data from the NIST MS Library database and creates a list of string vectors for each compound in the database.

Value

nistlist, a *list* object, a list of string vectors for each compound in the database.

`non_targeted` *Compares MS to atomic mass data*

Description

`non_targeted` compares the MS data from a peak to atomic mass data.

Usage

```
non_targeted(masslist, ms, THR = 0.1, ...)
```

Arguments

`masslist`	a *list* object, a list of atomic weights, ideally the output of mass_list().
`ms`	a *data.frame* object, a data frame of the mass values and the percent intensity values, ideally the output of find_ms().
`THR`	a *double* object. The threshold of intensity of which to include peaks for mass comparison. Default is 0.1.
`...`	a *vector* object. Any further optional inputs which indicate additional elements to consider in the compound, or restrictions on the number of a certain element in the compound. Should be in the form c('X', a, b) where X = element symbol, a = minimum number of atoms, b = maximum number of atoms. a and b are optional. If no minimum, use a=0, if no maximum, do not include b.

Details

This function takes the MS data from an intensity peak in a sample and compares it to combinations of atomic masses. Then it approximates the makeup of the compound, giving the best matches to the MS data. Note that the default matches will contain only H, N, C, O, F, Cl, Br, I, and Si. The user can input optional parameters to indicate additional elements to be considered or restrictions on the number of any specific element in the matching compounds.

Value

A *list* object, a list of vectors containing strings of the matching compounds.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
mz <- find_ms(frame, t_peak=peaks$'T'[1])
masslist <- mass_list()
non_targeted(masslist, mz, THR=0.05)
```

phase_shift *Phase shift*

Description

`phase_shift` shifts the phase of the chromatogram.

Usage

```
phase_shift(data, shift)
```

Arguments

`data`	a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
`shift`	a *float* object. The number of seconds to shift the phase by.

Details

This function shifts the phase of the chromatogram up or down by the specified number of seconds.

Value

A *data.frame* object. A list of two data frames. A TIC data frame and an MS data frame.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
shifted <- phase_shift(frame, -.2)
plot_chr(shifted, title='Shifted')
```

plot_chr *Plot chromatogram*

Description

`plot_chr` plots TIC data for chromatogram.

Usage

```
plot_chr(data, scale = "log", dim = 2, floor = -1, title = "Intensity")
```

Arguments

`data` a *list* object. Data extracted from a cdf file, ideally the output from extract_data().

`scale` a *string* object. Either 'linear' or 'log'. log refers to logarithm base 10. Default is log scale.

`dim` a *integer* object. The time dimensions of the plot, either 1 or 2. Default is 2.

`floor` a *float* object. The floor value for plotting. Values below floor will be scaled up. Default for linear plotting is 0, default for log plotting is 10^3.

`title` a *string* object. Title placed at the top of the plot. Default title "Intensity".

Details

This function creates a contour plot using of TIC data vs the x and y retention times using `ggplot` from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A contour plot of TIC data plotted in two dimensional retention time.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
plot_chr(frame, title='Raw Data', scale="linear")
plot_chr(frame, title='Log Intensity')
```

plot_defect *Plots the Kendrick Mass Defect of a peak*

Description

plot_defect Plots Kendrick Mass Defect of a peak.

Usage

```
plot_defect(ms, compound_mass = 14.01565, title = "Kendrick Mass Defect")
```

Arguments

ms	a *data.frame* object. A data frame of the mass values and the percent intensity values, ideally the output of find_ms().
compound_mass	a *float* object. The exact mass, using most common ions, of the desired atom group to base the Kendrick mass on. Default is 14.01565, which is the mass for CH_2.
title	a *string* object. Title placed at the top of the plot. Default title "Kendrick Mass Defect".

Details

This function produces a scatter plot of the Kendrick mass defects for mass spectrum data. Plotted using ggplot from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A line plot of the mass spectra data. The mass values vs the percent intensity values as a percent of the highest intensity.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
mz <- find_ms(frame, t_peak=peaks$'T'[1])
plot_ms(mz)
plot_defect(mz,title="Kendrick Mass Defect, CH_2")
```

plot_eic *Plots the EICs*

Description

plot_eic Plots the EICs

Usage

```
plot_eic(eic, title = "EIC", dim = 1)
```

Arguments

eic	a *data.frame* object. A data frame of the times and intensity values of the EIC of interest, ideally the output of find_eic().
title	a *string* object. Title placed at the top of the plot. Default title "EIC".
dim	a *integer* object. The time dimensions of the plot, either 1 or 2. Default is 1.

Details

This function produces a scatter plot of the overall time index vs the intensity values at a given mass of interest using ggplot from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A scatter plot of the overall time index vs the intensity values at a given mass of interest.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
eic <- find_eic(frame, MOI=92.1397,tolerance=0.005)
plot_eic(eic,dim=1,title='EIC for MOI 92.1397')
plot_eic(eic,dim=2,title='EIC for MOI 92.1397')
```

plot_gauss *Plots a peak with the fitted Gaussian curve.*

Description

plot_gauss Plots a peak with the fitted Gaussian curve.

Usage

```
plot_gauss(TIC_df, gauss_return, title = "Peak fit to Gaussian")
```

Arguments

TIC_df	a *data.frame* object. Data frame with 4 columns (Overall Time Index, RT1, RT2, TIC), ideally the output from create_df(), or the first data frame returned from extract_data(), $TIC_df.
gauss_return	a *data.frame* object. The output from guass_fit(). A data frame with two columns, (time, guassfit), the time values around the peak, and the intensity values fitted to the optimal Gaussian curve.
title	a *string* object. Title placed at the top of the plot.

Details

This function plots the points around the peak in blue dots, with a line plot of the Gaussian curve fit to the peak data in red, using ggplot from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A plot of points around the peak with a line plot of the Gaussian curve fit to the peak data.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
gaussfit <- gauss_fit(frame$TIC_df, peakcoord=c(peaks$'X'[1], peaks$'Y'[1]))
message(paste('Area under curve =',gaussfit[[3]], 'u^2'))
plot_gauss(frame$TIC_df, gaussfit[[1]])
```

plot_gauss2 *Plots a 3D peak with the fitted Gaussian curve.*

Description

plot_gauss2 Plots a 3D peak with the fitted Gaussian curve.

Usage

```
plot_gauss2(TIC_df, gauss2_return, title = "Peak fit to Gaussian")
```

Arguments

TIC_df	a *data.frame* object. Data frame with 4 columns (Overall Time Index, RT1, RT2, TIC), ideally the output from create_df(), or the first data frame returned from extract_data(), $TIC_df.
gauss2_return	a *data.frame* object. The output from guass_fit(). A data frame with two columns, (time, guassfit), the time values around the peak, and the intensity values fitted to the optimal Gaussian curve.
title	a *string* object. Title placed at the top of the plot.

Details

This function plots the points around the peak with a contour plot of the Gaussian curve fit to the peak data, using ggplot from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A contour plot of the Gaussian curve fit to the peak data.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
gaussfit2 <- gauss2_fit(frame$TIC_df, peakcoord=c(peaks$'X'[1], peaks$'Y'[1]))
message(paste('Volume under curve =',gaussfit2[[3]],'u^3'))
plot_gauss2(frame$TIC_df, gaussfit2[[1]])
```

plot_ms *Plots the mass spectra of a peak.*

Description

plot_ms Plots the mass spectra of a peak.

Usage

```
plot_ms(ms, title = "Mass Spectrum")
```

Arguments

ms	a *data.frame* object. A data frame of the mass values and the percent intensity values, ideally the output of find_ms().
title	a *string* object. Title placed at the top of the plot. Default title "Mass Spectrum".

Details

This function produces a line plot of the mass spectra data. The mass values vs the percent intensity values as a percent of the highest intensity using ggplot from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A line plot of the mass spectra data. The mass values vs the percent intensity values as a percent of the highest intensity.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
mz <- find_ms(frame, t_peak=peaks$'T'[1])
plot_ms(mz)
```

plot_nist *Plots the mass spectra of a NIST compound.*

Description

`plot_nist` Plots the mass spectra of a NIST compound.

Usage

```
plot_nist(nistlist, k, ms, title = "NIST Mass Spectrum")
```

Arguments

`nistlist`	a *list* object, a list of compound MS data from the NIST MS Library database, ideally the output of nist_list().
`k`	a *integer* object, the index of the NIST compound in the nistlist input.
`ms`	a *data.frame* object, a data frame of the mass values and the percent intensity values, ideally the output of find_ms().
`title`	a *string* object. Title placed at the top of the plot. Default title "Mass Spectrum".

Details

This function produces line plot of the mass spectra data from the sample on top, and the mass spectrum from a NIST compound entry on the bottom. The mass values vs the percent intensity values as a percent of the highest intensity using `ggplot` from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A line plot of the mass spectra data. The mass values vs the percent intensity values as a percent of the highest intensity.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, `https://ggplot2.tidyverse.org`.

plot_peak *Peak Plot*

Description

`plot_peak` plots peaks on a chromatograph plot.

Usage

```
plot_peak(
  peaks,
  data,
  title = "Intensity with Peaks",
  circlecolor = "red",
  circlesize = 5
)
```

Arguments

`peaks`	a *data.frame* object. A data frame with 4 columns (Time, X, Y, Peak), ideally the output from either thr_peaks() or top_peaks().
`data`	a *list* object. Data extracted from a cdf file, ideally the output from extract_data(). Provides the background GCxGC plot, created with plot_chr().
`title`	a *string* object. Title placed at the top of the plot. Default title "Intensity with Peaks".
`circlecolor`	a *string* object. The desired color of the circles which indicate the peaks. Default color red.
`circlesize`	a *double* object. The size of the circles which indicate the peaks. Default size 5.

Details

This function circles the identified peaks in a sample over a chromatograph plot (ideally smoothed) using `ggplot` from ggplot2 package (Wickham 2016).

Value

A *ggplot* object. A plot of the chromatogram heatmap, with identified peaks circled in red.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, `https://ggplot2.tidyverse.org`.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
plot_peak(peaks, frame, title="Top 20 Peaks")
```

---

plot_peakonly *Plot only peaks*

Description

plot_peakonly plots the peaks from a chromatograph.

Usage

```
plot_peakonly(peak_df, title = "Peaks")
```

Arguments

peak_df	a *data.frame* object. A data frame with 4 columns (Time, X, Y, Peak), ideally the output from top_peaks() or thr_peaks().
title	a *string* object. Title placed at the top of the plot. Default title "Peaks".

Details

This function creates a circle plot of the peak intensity vs the x and y retention times using ggplot from ggplot2 package (Wickham 2016). The size of the circle indicates the intensity of the peak.

Value

A *ggplot* object. A circle plot of peak intensity in 2D retention time.

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, https://ggplot2.tidyverse.org.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
plot_peakonly(peaks,title="Top 20 Peaks")
```

preprocess *Preprocessing*

Description

preprocess performs full preprocessing on a data file.

Usage

```
preprocess(
  filename,
  mod_t = 10,
  shift = 0,
  lambda = 20,
  gamma = 0.5,
  subtract = NULL,
  images = FALSE
)
```

Arguments

filename	a *string* object. The file name or path of the cdf file to be opened.
mod_t	a *float* object. The modulation time for the GCxGC sample analysis.Default is 10.
shift	a *float* object. The number of seconds to shift the phase by. Default is 0 to skip shifting.
lambda	a *float* object. A number (parameter in Whittaker smoothing), suggested between 1 to 10^5. Small lambda is very little smoothing, large lambda is very smooth. Default is lambda = 20.
gamma	a *float* object. Correction factor between 0 and 1. 0 results in almost no values being subtracted to the baseline, 1 results in almost everything except the peaks to be subtracted to the baseline. Default is 0.5.
subtract	a *data.frame* object. Data frame containing TIC data from a background sample or blank sample to be subtracted from the sample TIC data.
images	a *boolean* object. An optional input. If TRUE, all images of preprocessing steps will be displayed. Default is FALSE, no images will be displayed.

Details

This function performs full preprocessing on a data file. Extracts data and performs smoothing and baseline correction.

Value

A *data.frame* object. A list of two data frames. A TIC data frame and an MS data frame.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- preprocess(file,mod_t=.5,lambda=10,gamma=0.5,images=TRUE)
```

---

smooth *Smoothing*

Description

smooth performs smoothing of the intensity values.

Usage

```
smooth(data, lambda = 20, dir = "XY")
```

Arguments

data	a *list* object. Data extracted from a cdf file, ideally the output from extract_data().
lambda	a *float* object. A number (parameter in Whittaker smoothing), suggested between 0 to 10^4. Small lambda is very little smoothing, large lambda is very smooth. Default is lambda = 20.
dir	a *string* object. Either "X", "Y", or "XY" to indicate direction of smoothing. "XY" indicates smoothing in both X (horizontal) and Y (vertical) directions. Default "XY".

Details

This function performs smoothing of the intensity values using Whittaker smoothing algorithm whit1 from the ptw package (Eilers 2003).

Value

A *data.frame* object. A list of two data frames. A TIC data frame and an MS data frame.

References

Eilers PH (2003). “A perfect smoother.” *Analytical Chemistry*, **75**, 3631-3636.

Examples

```
file <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file,mod_t=.5)
sm_frame <- smooth(frame, lambda=10)
plot_chr(sm_frame, title='Smoothed')
```

targeted *Targeted Analysis*

Description

targeted performs targeted analysis for a batch of data files, for a list of masses of interest.

Usage

```
targeted(
  data_list,
  MOIs,
  RTs = c(),
  window_size = c(),
  tolerance = 0.005,
  images = FALSE
)
```

Arguments

data_list	a *list* object. Data extracted from each cdf file, ideally the output from extract_data().
MOIs	a *vector* object. A vector containing a list of all masses of interest to be investigated.
RTs	a *vector* object. An optional vector containing a list of retention times of interest for the listed masses of interest. Default values if left empty will be at the retention time of the highest intensity for the corresponding mass.
window_size	a *vector* object. An optional vector containing a list of window sizes corresponding to the retention times. Window will be defined by (RT-window_size, RT+window_size). Default if left empty will be 0.1.
tolerance	a *float* object. The tolerance allowed for the MOI. Default is 0.005.
images	a *boolean* object. An optional input. If TRUE, all images of the found peaks will be displayed. Default is FALSE, no images will be displayed.

Details

This function performs targeted analysis for a batch of data files, for a list of masses of interest.

Value

a *data.frame* object. A data frame containing the areas of the peaks for the indicated MOIs and list of files.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
file2 <- system.file("extdata","sample2.cdf",package="gcxgclab")
file3 <- system.file("extdata","sample3.cdf",package="gcxgclab")
frame1 <- extract_data(file1,mod_t=.5)
frame2 <- extract_data(file2,mod_t=.5)
frame3 <- extract_data(file3,mod_t=.5)
targeted(list(frame1,frame2,frame3),MOIs = c(92.1397, 93.07058),
RTs = c(6.930, 48.594), images=TRUE)
```

---

thr_peaks *Threshold Peaks*

Description

`thr_peaks` finds all peaks above the given threshold.

Usage

```
thr_peaks(TIC_df, THR = 1e+05)
```

Arguments

`TIC_df`	a *data.frame* object. Data frame with 4 columns (Overall Time Index, RT1, RT2, TIC), ideally the output from create_df(), or the first data frame returned from extract_data(), $TIC_df.
`THR`	a *float* object. Threshold for peak intensity. Should be a number between the baseline value and the highest peak intensity. Default suggestion is THR = 100000.

Details

This function finds all peaks in the sample above a given intensity threshold.

Value

A *data.frame* object. A data frame with 4 columns (Time, X, Y, Peak) with all peaks above the given threshold, with their time coordinates.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
thrpeaks <- thr_peaks(frame$TIC_df, 100000)
plot_peak(thrpeaks, frame, title="Peaks Above 100,000")
plot_peakonly(thrpeaks,title="Peaks Above 100,000")
```

top_peaks *Top Peaks*

Description

`top_peaks` finds the top N highest peaks.

Usage

```
top_peaks(TIC_df, N)
```

Arguments

`TIC_df`	a *data.frame* object. Data frame with 4 columns (Overall Time Index, RT1, RT2, TIC), ideally the output from create_df(), or the first data frame returned from extract_data(), $TIC_df.
`N`	*int* object. The number of top peaks to be found in the sample. N should be an integer >=1. Default suggestion is N = 20.

Details

This function finds the top N peaks in intensity in the sample.

Value

A *data.frame* object. A data frame with 4 columns (Time, X, Y, Peak) with the top N peaks, with their time coordinates.

Examples

```
file1 <- system.file("extdata","sample1.cdf",package="gcxgclab")
frame <- extract_data(file1,mod_t=.5)
peaks <- top_peaks(frame$TIC_df, 5)
plot_peak(peaks, frame, title="Top 20 Peaks")
plot_peakonly(peaks,title="Top 20 Peaks")
```

Index