Package 'chromatographR'

May 19, 2022

Type Package

Title Import and Analyze HPLC-DAD/UV Data

Version 0.4.1

Maintainer Ethan Bass <ethanbass@gmail.com>

Description Tools for high-throughput analysis of HPLC-DAD/UV chromatograms (or similar data). Includes functions for preprocessing, alignment, peak-finding and fitting, peak-table construction, data-visualization, etc. Preprocessing and peak-table construction follow the rough formula laid out in alsace (Wehrens, R., Bloemberg, T.G., and Eilers P.H.C., 2015. <doi:10.1093/bioinformatics/btv299>. Alignment of chromatograms is available using parametric time warping (ptw) (Wehrens, R., Bloemberg, T.G., and Eilers P.H.C. 2015. <doi:10.1093/bioinformatics/btv299>) or variable penalty dynamic time warping (VPdtw) (Clifford, D., & Stone, G. 2012. <doi:10.18637/jss.v047.i08>). Peak-finding uses the algorithm by Tom O'Haver <http://terpconnect.umd.edu/~toh/spectrum/PeakFindingandMeasurement.htm>. Peaks are then fitted to a gaussian or exponential-gaussian hybrid peak shape using non-linear least squares (Lan, K. & Jorgenson, J. W. 2001. <doi:10.1016/S0021-9673(01)00594-5>). See the vignette for more details and suggested workflow.

License GPL (>= 2)

URL https://ethanbass.github.io/chromatographR/

BugReports https://github.com/ethanbass/chromatographR/issues

Depends R (>= 3.5.0)

Imports chromConverter, dynamicTreeCut, fastcluster, graphics, grDevices, lattice, methods, minpack.lm, parallel, ptw, pvclust, scales, smoother, stats, utils

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0), VPdtw

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.0

Additional_repositories https://ethanbass.github.io/drat/

NeedsCompilation no

Author Ethan Bass [aut, cre] (<https://orcid.org/0000-0002-6175-6739>)

Repository CRAN

Date/Publication 2022-05-19 07:40:02 UTC

R topics documented:

chromatographR-package *chromatographR*

Description

chromatographR

Details

Package:	chromatographR
Type:	Package
Version:	0.4.1
Date:	2022-05-18
License:	GPL (>= 2)

Author(s)

Ethan Bass

Maintainer: Ethan Bass

attach_metadata *Attach experimental metadata*

Description

Attaches experimental metadata to 'peak_table' object. One of the columns in the supplied metadata must match exactly the row names of the peak table.

Usage

```
attach_metadata(peak_table, metadata, column)
```

Arguments

peak_table	A 'peak_table' object.
metadata	A 'data.frame' containing the sample meta-data.
column	The name of the column containing the sample names.

Value

A `peak_table` object with attached metadata in the `$sample_meta` slot.

Author(s)

Ethan Bass

See Also

get_peaktable normalize_data

Examples

```
data(pk_tab)
path <- system.file("extdata", "Sa_metadata.csv", package = "chromatographR")
meta <- read.csv(path)
pk_tab <- attach_metadata(peak_table = pk_tab, metadata = meta, column="vial")
```

---

attach_ref_spectra *Attach reference spectra*

Description

Gathers reference spectra and attaches them to peak_table object. Reference spectra are defined either as the spectrum with the highest intensity (max.int) or as the spectrum with the highest average correlation to the other spectra in the peak_table (max.cor).

Usage

```
attach_ref_spectra(peak_table, chrom_list, ref = c("max.cor", "max.int"))
```

Arguments

peak_table	Peak table from get_peaktable.
chrom_list	A list of chromatograms in matrix form (timepoints x wavelengths).
ref	What criterion to use to select reference spectra. Current options are maximum correlation (max.cor) or maximum signal intensity (max.int).

Value

A peak_table object with reference spectra attached in the $ref_spectra slot.

Author(s)

Ethan Bass

See Also

get_peaks get_peaktable

Examples

```
data(pk_tab)
pk_tab <- attach_ref_spectra(pk_tab, ref="max.int")
pk_tab <- attach_ref_spectra(pk_tab, ref = "max.cor")
```

`cluster_spectra` *Function to cluster peaks by spectral similarity.*

Description

Function to cluster peaks by spectral similarity. A representative spectrum is selected for each peak in the provided peak table and used to construct a distance matrix based on spectral similarity (pearson correlation) between peaks. Currently, representative spectrum is just selected from the chromatogram with the highest absorbance at lambda max. Hierarchical clustering with bootstrap resampling is performed on the resulting correlation matrix, as implemented in the `pvclust` package. Bootstrap values can be used to select clusters that exceed a certain confidence threshold as defined by alpha.

Usage

```
cluster_spectra(
  peak_table,
  chrom_list,
  peak_no = c(5, 100),
  alpha = 0.95,
  nboot = 1000,
  plot_dend = TRUE,
  plot_spectra = TRUE,
  verbose = TRUE,
  save = TRUE,
  parallel = TRUE,
  max.only = FALSE,
  ...
)
```

Arguments

`peak_table`	Peak table from `get_peaktable`.
`chrom_list`	A list of chromatograms in matrix form (timepoints x wavelengths).
`peak_no`	Minimum and maximum thresholds for the number of peaks a cluster may have.
`alpha`	Confidence threshold for inclusion of cluster.
`nboot`	Number of bootstrap replicates for `pvclust`.
`plot_dend`	Logical. If TRUE, plots dendrogram with bootstrap values.
`plot_spectra`	Logical. If TRUE, plots overlapping spectra for each cluster.
`verbose`	Logical. If TRUE, prints progress report to console.

save	Logical. If TRUE, saves pvclust object to current directory.
parallel	Logical. If TRUE, use parallel processing for pvclust.
max.only	Logical. If TRUE, returns only highest level for nested dendrograms.
...	Additional arguments to pvclust.

Value

Returns S4 "cluster" object with the following components:

peaks	a character vector containing the names of all peaks contained in the given cluster.
pval	a numeric vector of length 1 containing the bootstrap p-value (au) for the given cluster.

Author(s)

Ethan Bass

References

R. Suzuki, H. Shimodaira: Pvclust: an R package for assessing the uncertainty in hierarchical clustering. Bioinformatics, 22-12:1540-1542 (2006). doi:10.1093/bioinformatics/btl117.

Examples

```
data(pk_tab)
data(Sa_warp)
cl <- cluster_spectra(pk_tab, nboot=100, max.only = FALSE, save = FALSE, alpha = .97)
```

---

combine_peaks *Combine peaks in peak table*

Description

Utility function to combine duplicate peaks in peak table, i.e. peaks that were integrated at more than one wavelength or component. Specify tolerance (tol) for retention time matching and minimum spectral correlation (min.cor) for a match.

Usage

```
combine_peaks(peak_table, tol = 0.01, min.cor = 0.9, choose = "max")
```

Arguments

peak_table	Peak table from get_peaktable.
tol	Tolerance for matching retention times.
min.cor	Minimum spectral correlation to confirm a match.
choose	If "max" will retain peak with highest intensity. Otherwise, the first column in the data.frame will be retained.

Value

A peak table similar to the input peak table, but with duplicate columns combined according to the specified criteria.

Author(s)

Ethan Bass

See Also

get_peaks

Examples

```
data(pk_tab)
data(Sa_warp)
pk_tab <- attach_ref_spectra(pk_tab)
combine_peaks(pk_tab, tol = .02, min.cor = .9)
```

---

correct_peaks *Correct peak positions according to a ptw warping model*

Description

Corrects retention time differences using parametric time warping as implemented in ptw.

Usage

```
correct_peaks(peak_list, mod_list)
```

Arguments

peak_list	A nested list of peak tables: the first level is the sample, and the second level is the component. Every component is described by a matrix where every row is one peak, and the columns contain information on retention time, full width at half maximum (FWHM), peak width, height, and area.
mod_list	A list of ptw models.

Details

Once an appropriate warping model has been established, corrected retention times can be predicted for each peak. These are stored in a separate column in the list of peak tables.

Value

The input list of peak tables is returned with extra columns containing the corrected retention time.

Author(s)

Ron Wehrens

See Also

correct_rt

`correct_rt` *Correct retention time*

Description

Corrects retention time differences using parametric time warping, as implemented in ptw, or variable penalty dynamic time warping, as implemented in VPdtw.

Usage

```
correct_rt(
  chrom_list,
  lambdas,
  models = NULL,
  reference = "best",
  alg = c("ptw", "vpdtw"),
  what = c("models", "corrected.values"),
  init.coef = c(0, 1, 0),
  n.traces = NULL,
  n.zeros = 0,
  scale = FALSE,
  trwdth = 200,
  plot = FALSE,
  penalty = 5,
  maxshift = 50,
  verbose = FALSE,
  ...
)
```

Arguments

chrom_list	List of matrices containing concentration profiles.
lambdas	Select wavelengths to use by name.
models	List of models to warp by.
reference	Index of the sample that is to be considered the reference sample.
alg	algorithm to use: parametric time warping(ptw) or variable penalty dynamic time warping vpdtw.
what	What to return: either the 'corrected.values' (useful for visual inspection) or the warping 'models' (for further programmatic use).
init.coef	Starting values for the optimization.
n.traces	Number of traces to use.
n.zeros	Number of zeros to add.
scale	Logical. If true, scale chromatograms before warping.
trwdth	width of the triangle in the WCC criterion.
plot	Logical. Whether to plot alignment.
penalty	Divisor for dilation calculated by dilation. Adjusts penalty for variable penalty dynamic time warping.
maxshift	Integer. Maximum allowable shift for VPdtw.
verbose	Whether to be verbose.
...	Optional arguments for the ptw function. The only argument that cannot be changed is warp.type: this is always equal to "global".

Details

To use variable penalty dynamic time warping, the VPdtw package must be manually installed since it's no longer available from CRAN: `install.packages('VPdtw', repos='https://ethanbass.github.io/drat/')`.

Value

A list of ptw objects or a list of warped absorbance profiles, depending on the value of the what argument.

Note

Adapted from correctRT function in the alsace package by Ron Wehrens.

Author(s)

Ethan Bass

References

- Clifford, D., Stone, G., Montoliu, I., Rezzi, S., Martin, F. P., Guy, P., ... & Kochhar, S. 2009. Alignment using variable penalty dynamic time warping. *Analytical chemistry*, **81(3)**:1000-1007. doi:10.1021/ac802041e.
- Clifford, D., & Stone, G. 2012. Variable Penalty Dynamic Time Warping Code for Aligning Mass Spectrometry Chromatograms in R. *Journal of Statistical Software*, **47(8)**:1-17. doi:10.18637/jss.v047.i08.
- Eilers, P.H.C. 2004. Parametric Time Warping. *Anal. Chem.*, **76**:404-411. doi:10.1021/ac034800e.
- Wehrens, R., Bloemberg, T.G., and Eilers P.H.C. 2015. Fast parametric time warping of peak lists. *Bioinformatics*, **31**:3063-3065. doi:10.1093/bioinformatics/btv299.
- Wehrens, R., Carvalho, E., Fraser, P.D. 2015. Metabolite profiling in LC–DAD using multivariate curve resolution: the alsace package for R. *Metabolomics*, **11**:143-154. doi:10.1007/s1130601406835

See Also

ptw, correct_peaks, VPdtw

Examples

```
data(Sa_pr)
warping.models <- correct_rt(Sa_pr, what = "models", lambdas=c("210"))
warp <- correct_rt(chrom_list = Sa_pr, models = warping.models)
```

---

filter_peaks *Filter peak lists*

Description

Utility function to remove peaks from a peak list, e.g. because their intensity is too low. Currently one can filter on peak height, peak area, and width at half maximum.

Usage

```
filter_peaks(peak_list, min_height, min_area, min_sd, max_sd)
```

Arguments

peak_list	A peak_list object, consisting of a nested list of peak tables, where the first level is the sample, and the second level is the spectral component. Every component is described by a matrix where every row is one peak, and the columns contain information on retention time, full width at half maximum (FWHM), peak width, height, and area.

min_height	Minimum peak height.
min_area	Minimum peak area.
min_sd	Minimal standard deviation.
max_sd	Maximum standard deviation.

Value

A peak list similar, with all rows removed from the peak tables that are not satisfying the criteria.

Author(s)

Ron Wehrens, Ethan Bass

See Also

get_peaks

---

find_peaks *Find peaks in chromatographic profile*

Description

Find peaks in chromatographic profile.

Usage

```
find_peaks(
  y,
  smooth_type = "gaussian",
  smooth_window = 1,
  smooth_width = 0.1,
  slope_thresh = 0,
  amp_thresh = 0,
  bounds = TRUE
)
```

Arguments

y	response (numerical vector)
smooth_type	Type of smoothing. (Defaults to "gaussian").
smooth_window	Window for smoothing. (Defaults to 1).
smooth_width	Width for smoothing. (Defaults to 0.1).
slope_thresh	Minimum threshold for peak slope. (Defaults to 0).
amp_thresh	Minimum threshold for peak amplitude. (Defaults to 0).
bounds	Logical. If TRUE, includes peak boundaries in data.frame. (Defaults to TRUE).

Details

Find peaks with function `find_peaks` by looking for zero-crossings in the smoothed first derivative of a signal that exceed a given slope threshold.

Value

If bounds == TRUE, returns a data.frame containing the center, start, and end of each identified peak. Otherwise, returns a numeric vector of peak centers. All locations are expressed as indices.

Note

The `find_peaks` function is adapted from matlab code in Prof. Tom O'Haver's Pragmatic Introduction to Signal Processing.

Author(s)

Ethan Bass

References

O'Haver, Tom. Pragmatic Introduction to Signal Processing: Applications in scientific measurement. /hrefhttps://terpconnect.umd.edu/~toh/spectrum/ (Accessed January, 2022).

See Also

`fit_peaks`, `get_peaks`

Examples

```
data(Sa_pr)
find_peaks(Sa_pr[[1]][,"220"])
```

`fit_peaks` *Fit chromatographic peaks to an exponential-gaussian hybrid or gaussian profile*

Description

Fit peak parameters using exponential-gaussian hybrid or gaussian function.

Usage

```
fit_peaks(
  y,
  pos = NULL,
  sd.max = 50,
  fit = c("egh", "gaussian", "raw"),
  max.iter = 1000,
  ...
)
```

Arguments

y	response (numerical vector)
pos	Locations of peaks in vector y. If NULL, find_peaks will run automatically to find peak positions.
sd.max	Maximum width (standard deviation) for peaks. Defaults to 50.
fit	Function for peak fitting. (Currently exponential-gaussian hybrid egh, gaussian and raw settings are supported). If raw is selected, trapezoidal integration will be performed on raw data without fitting a peak shape. Defaults to egh.)
max.iter	Maximum number of iterations to use in nonlinear least squares peak-fitting. (Defaults to 1000).
...	Additional arguments to find_peaks.

Details

Peak parameters are calculated using fit_peaks, which fits the data to a gaussian or exponential-gaussian hybrid curve using non-linear least squares estimation as implemented in nlsLM. Area under the fitted curve is estimated using trapezoidal estimation.

Value

Function fit_peaks returns a matrix, whose columns contain the following information:

rt	location of the maximum of the peak (x)
start	start of peak (only included in table if 'bounds==TRUE')
end	end of peak (only included in table if 'bounds==TRUE')
sd	width of the peak (x)
tau	tau parameter (only included in table if 'fit=="egh"')
FWHM	full width at half maximum (x)
height	height of the peak (y)
area	peak area
r.squared	r-squared value for linear fit of model to data.

Again, the first five elements (rt, start, end, sd and FWHM) are expressed as indices, so not in terms of the real retention times. The transformation to "real" time is done in function get_peaks.

Note

The fit_peaks function is adapted from Dr. Robert Morrison's DuffyTools package as well as code published in Ron Wehrens' alsace package.

Author(s)

Ethan Bass

References

Lan, K. & Jorgenson, J. W. 2001. A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks. *Journal of Chromatography A* **915**:1-13. doi:10.1016/S00219673(01)005945.

Naish, P. J. & Hartwell, S. 1988. Exponentially Modified Gaussian functions - A good model for chromatographic peaks in isocratic HPLC? *Chromatographia*, /bold26: 285-296. doi:10.1007/BF02268168.

See Also

find_peaks, get_peaks

Examples

```
data(Sa_pr)
fit_peaks(Sa_pr[[1]][,"220"])
```

---

get_peaks *Get peak list.*

Description

Finds and fits peaks and extracts peak parameters from a list of chromatograms at the specified wavelengths.

Usage

```
get_peaks(
  chrom_list,
  lambdas,
  fit = c("egh", "gaussian", "raw"),
  sd.max = 50,
  max.iter = 100,
  ...
)
```

Arguments

chrom_list	A list of profile matrices, each of the same dimensions (timepoints x wavelengths).
lambdas	Character vector of wavelengths to find peaks at.
fit	What type of fit to use. Current options are exponential-gaussian hybrid (egh), gaussian or raw. The raw setting performs trapezoidal integration directly on the raw data without fitting a peak shape.
sd.max	Maximum width (standard deviation) for peaks. Defaults to 50.
max.iter	Maximum number of iterations for non-linear least squares in fit_peaks.
...	Additional arguments to find_peaks.

Details

Peaks are located by finding zero-crossings in the smoothed first derivative of the specified chromatographic traces (function `find_peaks`). At the given positions, an exponential-gaussian hybrid (or regular gaussian) function is fit to the signal using `fit_peaks`). The area is then calculated using a trapezoidal approximation.

Value

The result is an S3 object of class `peak_list`, containing a nested list of data.frames containing information about the peaks fitted for each chromatogram at each specified wavelength. The data.frame includes information about the retention time (`rt`), `start` and `end` of each peak, as well as the standard deviation (`sd`), `tau` (if `egh` is selected), full width at half maximum (`FWHM`), `height`, `area`, and `r.squared` (coefficient of determination). (*Note:* This last parameter is determined from a linear model of the fitted peak values to the raw data. This approach is not really statistically valid but it can be useful as a rough metric for "goodness-of-fit").

Note

The function is adapted from the getAllPeaks function authored by Ron Wehrens (though the underlying algorithms for peak identification and peak-fitting are not the same).

Author(s)

Ethan Bass

References

Wehrens, R., Carvalho, E., Fraser, P.D. 2015. Metabolite profiling in LC–DAD using multivariate curve resolution: the alsace package for R. *Metabolomics* **11**:143-154. doi:10.1007/s11306014-06835

Examples

```
data(Sa_pr)
pks <- get_peaks(Sa_pr, lambdas = c('210'), sd.max=50, fit="egh")
```

`get_peaktable` *Convert peak list into an ordered peak table.*

Description

Returns a peak_table object. The first slot contains a matrix of intensities, where rows correspond to samples and columns correspond to aligned features. The rest of the slots contain various meta-data about peaks, samples, and experimental settings.

Usage

```
get_peaktable(
  peak_list,
  chrom_list,
  response = c("area", "height"),
  use.cor = FALSE,
  hmax = 0.2,
  plot_it = FALSE,
  ask = plot_it,
  clust = c("rt", "sp.rt"),
  sigma.t = NULL,
  sigma.r = 0.5,
  deepSplit = FALSE,
  verbose = FALSE,
  out = c("data.frame", "matrix")
)
```

Arguments

peak_list	A peak_list object created by get_peaks, containing a nested list of peak tables: the first level is the sample, and the second level is the spectral component. Every component is described by a data.frame where every row is one peak, and the columns contain information on various peak parameters.
chrom_list	A list of chromatographic matrices.
response	Indicates whether peak area or peak height is to be used as intensity measure. Defaults to area setting.
use.cor	Logical. Indicates whether to use corrected retention times (by default) or raw retention times (not advised!).
hmax	Height at which the complete linkage dendrogram will be cut. Can be interpreted as the maximal inter-cluster retention time difference.
plot_it	Logical. If TRUE, for every component a stripplot will be shown indicating the clustering.
ask	Logical. Ask before showing new plot?
clust	Specify whether to perform hierarchical clustering based on spectral similarity and retention time (sp.rt) or retention time alone (rt). Defaults to rt. The sp.rt option is experimental and should be used with caution.
sigma.t	Width of gaussian in retention time distance function. Controls weight given to retention time if sp.rt is selected.
sigma.r	Width of gaussian in spectral similarity function. Controls weight given to spectral correlation if sp.rt is selected.
deepSplit	Logical. Controls sensitivity to cluster splitting. If TRUE, function will return more smaller clusters. See documentation for cutreeDynamic for additional information.
verbose	Logical. Whether to print warning when combining peaks into single time window. Defaults to FALSE.
out	Specify data.frame or matrix as output. Defaults to data.frame.

Details

The function performs a complete linkage clustering of retention times across all samples, and cuts at a height given by the user (which can be understood as the maximal inter-cluster retention time difference) in the simple case based on retention times. Clustering can also incorporate information about spectral similarity using a distance function adapted from Broeckling et al., 2014:

$$latexascii$$

If two peaks from the same sample are assigned to the same cluster, a warning message is printed to the console. These warnings can usually be ignored, but one could also consider reducing the `hmax` variable. However, this may lead to splitting of peaks across multiple clusters. Another option is to filter the peaks by intensity to remove small features.

Value

The function returns a `peak_table` object, consisting of the following elements:

- `tab`: the peak table itself – a data-frame of intensities in a sample x peak configuration.
- `pk_meta`: A data.frame containing peak meta-data (e.g. the spectral component, peak number, and average retention time).
- `sample_meta`: A data.frame of sample meta-data. Must be added using `attach_metadata`).
- `ref_spectra`: A data.frame of reference spectra (in a wavelength x peak configuration). Must be added using `attach_ref_spectra`
- `args`: A vector of arguments given to `get_peaktable` to generate the peak table.

Note

Adapted from getPeakTable function in the alsace package by Ron Wehrens.

Author(s)

Ethan Bass

References

- Broeckling, C. D., F. A. Afsar, S. Neumann, A. Ben-Hur, and J. E. Prenni. 2014. RAM-Clust: A Novel Feature Clustering Method Enables Spectral-Matching-Based Annotation for Metabolomics Data. *Anal. Chem.* **86**:6812-6817. doi:10.1021/ac501530d
- Wehrens, R., Carvalho, E., Fraser, P.D. 2015. Metabolite profiling in LC–DAD using multivariate curve resolution: the alsace package for R. *Metabolomics* **11**:143-154. doi:10.1007/s1130601406835

See Also

`attach_ref_spectra` `attach_metadata`

Examples

```
data(Sa_pr)
pks <- get_peaks(Sa_pr, lambdas = c('210'))
get_peaktable(pks, response = "area")
```

`load_chroms` *Import chromatograms.*

Description

Convenience function to import chromatograms from a list of folders or paths.

Usage

```
load_chroms(
  paths,
  find_files = TRUE,
  format.in = c("csv", "chemstation", "masshunter"),
  sep = ",",
  dat = NULL,
  ...
)
```

Arguments

`paths`	Path(s) to chromatograms or the folders containing the files
`find_files`	Logical. Set to `TRUE` (default) if you are providing the function with a folder or vector of folders containing the files. Otherwise, set to`FALSE`.
`format.in`	Format of files.
`sep`	Argument provided to `read.csv`. Defaults to ",".
`dat`	Optional list of chromatograms. If provided, newly imported chromatograms will be appended to the existing list.
`...`	Additional arguments to `read.csv`.

Details

Chromatograms may be CSVs, ChemStation `.uv` files, or MassHunter `.sp` files. Parsers from the Aston package for python are used to load binary files.

Value

A list of chromatograms in matrix format.

Note

Relies on the file parsers from the Aston package to import ChemStation `.uv` and MassHunter `.sp` files.

Author(s)

Ethan Bass

Examples

```
## Not run:
###  import from single folder
dat <- load_chromes(paths = path)
### import from multiple folders
path = 'foo'
folders <- list.files(path = path, pattern = "EXPORT3D")
dat <- load_chroms(folders)

## End(Not run)
```

`mirror_plot` *Make mirror plot from peak table.*

Description

Plots chromatograms as a mirror plot.

Usage

```
mirror_plot(
  peak_table,
  chrom_list,
  lambdas,
  var,
  subset = NULL,
  print_legend = TRUE,
  legend_txt = NULL,
  legend_pos = "topright",
  legend_size = 1,
  mirror = TRUE,
  xlim = NULL,
  ylim = NULL,
  ...
)
```

Arguments

peak_table	The peak table (output from get_peaktable function).
chrom_list	A list of chromatograms in matrix form (timepoints x wavelengths).
lambdas	The wavelength you wish to plot the traces at.
var	Variable to index chromatograms.
subset	Character vector specifying levels to use (if more than 2 levels are present in var).
print_legend	Logical. Whether to print legend. Defaults to TRUE.
legend_txt	Character vector containing labels for legend.
legend_pos	Legend position.
legend_size	Legend size (cex argument). Default is 1.
mirror	Logical. Whether to plot as mirror or stacked plots. Defaults to TRUE.
xlim	Numerical vector specifying limits for x axis.
ylim	Numerical vector specifying limits for y axis.
...	Additional arguments to matplot function.

Details

Can be used to confirm the identity of a peak or check that a particular column in the peak table represents a single compound. Can also be used to create simple box-plots to examine the distribution of a peak with respect to variables defined in sample metadata.

Value

No return value, called for side effects.

Side effects

If mirror_plot is TRUE, plots a mirror plot comparing two treatments defined by var and subset (if more than two factors are present in var).

Otherwise, if mirror_plot is FALSE, the treatments are plotted in two separate panes.

Author(s)

Ethan Bass

Examples

```
data(Sa_warp)
data(pk_tab)
path <- system.file("extdata", "Sa_metadata.csv", package = "chromatographR")
meta <- read.csv(path)
pk_tab <- attach_metadata(peak_table = pk_tab, metadata = meta, column="vial")
mirror_plot(pk_tab,lambdas=c("210","260"), var="trt", mirror=TRUE, col=c("green","blue"))
```

normalize_data *Normalize peak table or chromatograms*

Description

Normalizes peak table or list of chromatograms by specified column in sample meta-data. Metadata must first be attached to peak_table using attach_metadata.

Usage

```
normalize_data(
  peak_table,
  column,
  chrom_list,
  what = c("peak_table", "chrom_list")
)
```

Arguments

peak_table	A 'peak_table' object
column	The name of the column containing the weights.
chrom_list	List of chromatograms for normalization. The samples must be in same order as the peak_table.
what	'peak_table' or list of chromatograms ('chrom_list').

Value

A peak_table object where the peaks are normalized by the mass of each sample.

Author(s)

Ethan Bass

See Also

get_peaktable attach_metadata

Examples

```
data(pk_tab)
path <- system.file("extdata", "Sa_metadata.csv", package = "chromatographR")
meta <- read.csv(path)
pk_tab <- attach_metadata(peak_table = pk_tab, metadata = meta, column="vial")
norm <- normalize_data(pk_tab, "mass", what = "peak_table")
```

pk_tab *Goldenrod peak table*

Description

Peak table generated from example goldenrod extracts for examples.

Format

A `peak_table` object.

plot.peak_list *Plot fitted peak shapes.*

Description

Visually assess integration accuracy by plotting fitted peaks over trace.

Usage

```
## S3 method for class 'peak_list'
plot(
  x,
  ...,
  chrom_list = NULL,
  index = 1,
  lambda = NULL,
  points = FALSE,
  ticks = FALSE,
  a = 0.5,
  color = NULL,
  cex.points = 0.5
)
```

Arguments

`x`	Peak_list object. Output from the `get_peaks` function.
`...`	Additional arguments to plot function.
`chrom_list`	List of chromatograms (retention time x wavelength matrices)
`index`	Index or name of chromatogram to be plotted.
`lambda`	Wavelength for plotting.
`points`	Logical. If TRUE, plot peak maxima. Defaults to FALSE.
`ticks`	Logical. If TRUE, mark beginning and end of each peak. Defaults to FALSE.
`a`	Alpha parameter controlling the transparency of fitted shapes.
`color`	The color of the fitted shapes.
`cex.points`	Size of points. Defaults to 0.5

Value

No return value, called for side effects.

Side effects

Plots a chromatographic trace from the specified chromatogram (`chr`) at the specified wavelength (`lambda`) with fitted peak shapes from the provided `peak_list` drawn underneath the curve.

Author(s)

Ethan Bass

See Also

`get_peaks`

`plot.peak_table` *Plot spectrum from peak table*

Description

Plots the trace and/or spectrum for a given peak in peak table.

Usage

```
## S3 method for class 'peak_table'
plot(
  x,
  ...,
  loc,
  chrom_list,
  what = "peak",
  chr = "max",
  lambda = "max",
  plot_spectrum = TRUE,
  plot_trace = TRUE,
  box_plot = FALSE,
  vars = NULL,
  spectrum_labels = TRUE,
  scale_spectrum = FALSE,
  export_spectrum = FALSE,
  verbose = TRUE
)
```

Arguments

x	The peak table (output from get_peaktable function).
...	Additional arguments.
loc	The name of the peak or retention time that you wish to plot.
chrom_list	A list of chromatograms in matrix form (timepoints x wavelengths).
what	What to look for. Either peak to extract spectral information for a certain peak, rt to scan by retention time, or click to manually select retention time by clicking on the chromatogram. Defaults to peak.
chr	Numerical index of chromatogram you wish to plot; "max" to plot the chromatogram with the largest signal; or "all" to plot spectra for all chromatograms.
lambda	The wavelength you wish to plot the trace at (if plot_chrom is TRUE and/or the wavelength to be used for the determination of signal abundance.
plot_spectrum	Logical. If TRUE, plots the spectrum of the chosen peak. Defaults to TRUE.
plot_trace	Logical. If TRUE, plots the trace of the chosen peak at lambda. Defaults to TRUE.
box_plot	Logical. If TRUE, plots box plot using categories defined by vars.
vars	Independent variables for boxplot.
spectrum_labels	Logical. If TRUE, plots labels on maxima in spectral plot. Defaults to TRUE.
scale_spectrum	Logical. If TRUE, scales spectrum to unit height. Defaults to FALSE.
export_spectrum	Logical. If TRUE, exports spectrum to console. Defaults to FALSE.
verbose	Logical. If TRUE, prints verbose output to console. Defaults to TRUE.

Details

Can be used to confirm the identity of a peak or check that a particular column in the peak table represents a single compound. Can also be used to create simple box-plots to examine the distribution of a peak with respect to variables defined in sample metadata.

Value

If export_spectrum is TRUE, returns the spectrum as a data.frame with wavelengths as rows and columns encoding the absorbance (or normalized absorbance, if scale_spectrum is TRUE) for the specified sample(s). Otherwise, there is no return value.

Side effects

If plot_trace is TRUE, plots the chromatographic trace of the specified chromatogram (chr), at the specified wavelength (lambda) with a dotted red line to indicate the retention time given by loc. The trace is a single column from the chromatographic matrix.

If plot_spectrum is TRUE, plots the spectrum for the specified chromatogram at the specified retention time. The spectrum is a single row from the chromatographic matrix.

If box_plot is TRUE, produces a boxplot from the specified peak with groups provided by vars.

Author(s)

Ethan Bass

`plot_all_spectra` *Plot all spectra for chosen peak.*

Description

Plot multiple for a given peak in peak table. Wrapper for `plot_spectrum`.

Usage

```
plot_all_spectra(
  peak,
  peak_table,
  chrom_list,
  chrs = "all",
  plot_spectrum = TRUE,
  export_spectrum = TRUE,
  scale_spectrum = TRUE,
  overlapping = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

`peak`	The name of a peak to plot (in character format)
`peak_table`	The peak table (output from `get_peaktable` function)
`chrom_list`	A list of profile matrices, each of the same dimensions (timepoints x components).
`chrs`	Vector of chromatograms to plot.
`plot_spectrum`	Logical. If TRUE, plots the spectrum of the chosen peak.
`export_spectrum`	Logical. If TRUE, exports spectrum to console. Defaults to FALSE.
`scale_spectrum`	Logical. If TRUE, scales spectrum to unit height.
`overlapping`	Logical. If TRUE, plot spectra in single plot.
`verbose`	Logical. If TRUE, prints verbose output to console.
`...`	Additional arguments to plot_spectrum.

Value

If `export_spectrum` is TRUE, returns the spectra as a `data.frame` with wavelengths as rows and one column for each sample in the `chrom_list` encoding the absorbance (or normalized absorbance, if `scale_spectrum` is TRUE) at each wavelength. Otherwise, there is no return value.

Side effects

If plot_spectrum is TRUE, plots the spectra for the specified chromatogram (chr) of the given peak. The spectrum is a single row from the chromatographic matrix.

Author(s)

Ethan Bass

See Also

plot_spectrum

Examples

```
data(Sa_warp)
pks <- get_peaks(Sa_warp, lambda="220")
pk_tab <- get_peaktable(pks)
plot_all_spectra(peak="V13", peak_table = pk_tab, overlapping=TRUE)
```

---

plot_spectrum *Plot spectrum from peak table*

Description

Plots the trace and/or spectrum for a given peak in peak.table object, or plots the spectrum a particular retention time for a given chromatogram.

Usage

```
plot_spectrum(
  loc,
  peak_table,
  chrom_list,
  chr = "max",
  lambda = "max",
  plot_spectrum = TRUE,
  plot_trace = TRUE,
  spectrum_labels = TRUE,
  scale_spectrum = FALSE,
  export_spectrum = FALSE,
  verbose = TRUE,
  what = c("peak", "rt", "click"),
  ...
)
```

Arguments

loc	The name of the peak or retention time for which you wish to extract spectral data.
peak_table	The peak table (output from get_peaktable function).
chrom_list	A list of chromatograms in matrix form (timepoints x wavelengths).
chr	Numerical index of chromatogram you wish to plot, or "max" to automatically plot the chromatogram with the largest signal.
lambda	The wavelength you wish to plot the trace at if plot_trace == TRUE and/or the wavelength to be used for the determination of signal abundance.
plot_spectrum	Logical. If TRUE, plots the spectrum of the chosen peak. Defaults to TRUE.
plot_trace	Logical. If TRUE, plots the trace of the chosen peak at lambda. Defaults to TRUE.
spectrum_labels	Logical. If TRUE, plots labels on maxima in spectral plot. Defaults to TRUE.
scale_spectrum	Logical. If TRUE, scales spectrum to unit height. Defaults to FALSE.
export_spectrum	Logical. If TRUE, exports spectrum to console. Defaults to FALSE.
verbose	Logical. If TRUE, prints verbose output to console. Defaults to TRUE.
what	What to look for. Either "peak" to extract spectral information for a certain peak, "rt" to scan by retention time, or "click" to manually select retention time by clicking on the chromatogram. Defaults to "peak" mode.
...	Additional arguments.

Details

Can be used to confirm the identity of a peak or check that a particular column in the peak table represents a single compound. Retention times can also be selected by clicking on the plotted trace if what == 'click'.

Value

If export_spectrum is TRUE, returns the spectrum as a data.frame with wavelengths as rows and a single column encoding the absorbance (or normalized absorbance, if scale_spectrum is TRUE) at each wavelength. Otherwise, there is no return value.

Side effects

If plot_trace is TRUE, plots the chromatographic trace of the specified chromatogram (chr), at the specified wavelength (lambda) with a dotted red line to indicate the retention time given by loc. The trace is a single column from the chromatographic matrix.

If plot_spectrum is TRUE, plots the spectrum for the specified chromatogram at the specified retention time. The spectrum is a single row from the chromatographic matrix.

Author(s)

Ethan Bass

Examples

`preprocess` *Preprocess time/wavelength data*

Description

Standard pre-processing of response matrices, consisting of a time axis and a spectral axis (e.g. HPLC-DAD/UV data). For smooth data, like UV-VIS data, the size of the matrix can be reduced by interpolation. By default, the data are baseline-corrected in the time direction and smoothed in the spectral dimension.

Usage

```
preprocess(
  X,
  dim1,
  dim2,
  remove.time.baseline = TRUE,
  spec.smooth = TRUE,
  maxI,
  parallel = TRUE,
  interpolate_rows = TRUE,
  interpolate_cols = TRUE,
  mc.cores = 2,
  ...
)
```

Arguments

`X`	A numerical data matrix, or list of data matrices. Missing values are not allowed. If rownames or colnames attributes are used, they should be numerical and signify time points and wavelengths, respectively.
`dim1`	A new, usually shorter, set of time points (numerical). The range of these should not be outside the range of the original time points, otherwise the function stops with an error message.
`dim2`	A new, usually shorter, set of wavelengths (numerical). The range of these should not be outside the range of the original wavelengths, otherwise the function stops with an error message.
`remove.time.baseline`	Logical, indicating whether baseline correction should be done in the time direction, according to `baseline.corr`. Default is TRUE.
`spec.smooth`	Logical, indicating whether smoothing should be done in the spectral direction, according to `smooth.spline`. Default is TRUE.

`maxI`	if given, the maximum intensity in the matrix is set to this value.
`parallel`	Logical, indicating whether to use parallel processing. Defaults to TRUE (unless you're on Windows).
`interpolate_rows`	Logical. Whether to interpolate along dim1. Defaults to TRUE.
`interpolate_cols`	Logical. Whether to interpolate along dim2. Defaults to TRUE.
`mc.cores`	How many cores to use for parallel processing. Defaults to 2.
`...`	Further optional arguments to `baseline.corr`.

Value

The function returns the preprocessed data matrix, with rownames and colnames indicating the time points and wavelengths, respectively.

Note

Adapted from preprocess function in the alsace package by Ron Wehrens.

Author(s)

Ethan Bass

References

- Wehrens, R., Bloemberg, T.G., and Eilers P.H.C. 2015. Fast parametric time warping of peak lists. *Bioinformatics* **31**:3063-3065. doi:10.1093/bioinformatics/btv299.
- Wehrens, R., Carvalho, E., Fraser, P.D. 2015. Metabolite profiling in LC–DAD using multivariate curve resolution: the alsace package for R. *Metabolomics* **11:1**:143-154. doi:10.1007/s1130601406835.

Examples

```
data(Sa)
new.ts <- seq(10,18.66,by=.01) # choose time-points
new.lambdas <- seq(200, 318, by = 2) # choose wavelengths
Sa_pr <- preprocess(Sa[[1]], dim1 = new.ts, dim2 = new.lambdas)
```

Sa — *HPLC-DAD data of goldenrod root extracts.*

Description

Four HPLC-DAD data matrices of *Solidago altissima* roots extracted in 90 percent methanol.

Format

A list of four matrices (time x wavelength).

Sa_pr — *HPLC-DAD data of goldenrod root extracts.*

Description

Pre-processed chromatograms.

Format

Four pre-processed matrices (time x wavelength) to use in examples.

Sa_warp — *HPLC-DAD data of goldenrod root extracts.*

Description

Pre-processed and warped chromatograms.

Format

Four pre-processed and warped matrices (time x wavelength) to use in examples.

scan_chrom *Scan spectrum*

Description

Convenience function to call plot_spectrum with what = "click".

Usage

```
scan_chrom(
  chrom_list,
  lambda,
  chr,
  peak_table = NULL,
  scale_spectrum = FALSE,
  spectrum_labels = TRUE,
  export_spectrum = FALSE,
  ...
)
```

Arguments

chrom_list	A list of chromatograms in matrix form (timepoints x wavelengths).
lambda	The wavelength to plot the trace at.
chr	Numerical index of chromatogram you wish to plot.
peak_table	The peak table (output from get_peaktable function).
scale_spectrum	Logical. If TRUE, scales spectrum to unit height. Defaults to FALSE.
spectrum_labels	Logical. If TRUE, plots labels on maxima in spectral plot. Defaults to TRUE.
export_spectrum	Logical. If TRUE, exports spectrum to console. Defaults to FALSE.
...	Additional arguments.

Value

If export_spectrum is TRUE, returns the spectrum as a data.frame with wavelengths as rows and a single column encoding the absorbance (or normalized absorbance, if scale_spectrum is TRUE) at each wavelength. Otherwise, there is no return value.

Side effects

Plots a chromatographic trace from the specified chromatogram (chr), at the specified wavelength (lambda) with a dotted red line to indicate the user-selected retention time. The trace is a single column from the chromatographic matrix.

If plot_spectrum is TRUE, plots the spectrum for the specified chromatogram at the user-specified retention time. The spectrum is a single

Author(s)

Ethan Bass

Examples

```
data(Sa_pr)
scan_chrom(Sa_pr, lambda="210", chr=2, export_spectrum=TRUE)
```

Index