Package 'subMALDI'

October 5, 2020

Type Package

Title Organization and Processing of MALDI-MS Datasets

Version 1.0-0

Date August 27, 2020

Author [aut, cre] Kristen Yeh, [aut] Wesley Burr

Maintainer Wesley Burr <wesleyburr@trentu.ca>

Description subMALDI is an open framework tool that permits organization, pre-processing (smoothing, baseline correction, peak detection), and normalization of spectral data sets. As a result of the package's open framework, subMALDI data sets are compatible with functions from a wide variety of other R packages, and user-defined functions are easier to implement and test.

Depends R (>= 3.5.0), dplyr (>= 0.8.0), tidyr (>= 1.1.0), ggplot2 (>= 3.3.0), ggpmisc (>= 0.3.3)

Imports signal (>= 0.7-6), MassSpecWavelet (>= 1.48.0)

License GPL (>= 3) + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

RoxygenNote 7.1.1

R topics documented:

`subMALDI-package` *subMALDI: Organization and Processing of MALDI-MS Datasets*

Description

To learn more about subMALDI, start with the vignettes: `browseVignettes(package = "subMALDI")`

Author(s)

Maintainer: Wesley Burr <wesleyburr@trentu.ca> (ORCID)

Authors:

- Kristen Yeh (ORCID)
- Wesley Burr (ORCID)

See Also

Useful links:

- Report bugs at https://github.com/wesleyburr/subMALDI/issues

After1 *Pairwise Spectral Data Frame from Sample*

Description

Raw spectral data in binary dataframe. The first column contains the *m/z* data for the spectrum, while the second contains the intensity data. Spectrum is obtained from a biological sample, after chemical intervention.

Usage

```
data("After1")
```

Format

A data frame with 134674 observations on the following 2 variables.

`mass` A numeric vector; the *m/z* data for the spectrum.

`Intensity` A numeric vector; the intensity data for the spectrum.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(After1)
```

After2 *Pairwise Spectral Data Frame from Sample*

Description

Raw spectral data in binary dataframe. The first column contains the *m/z* data for the spectrum, while the second contains the intensity data. Spectrum is obtained from a biological sample, after chemical intervention.

Usage

```
data("After2")
```

Format

A data frame with 176209 observations on the following 2 variables.

`mass` A numeric vector; the *m/z* data for the spectrum.

`Intensity` A numeric vector; the intensity data for the spectrum.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(After2)
```

`avgSpectra` *Average Spectral Replicates*

Description

Combines spectral replicates either by averaging (method = "mean") or summing (method = "sum) the intensity values across each row representing a mass-to-charge value in `full_mz`. Capable of averaging 1-6 spectra at once.

Usage

```
avgSpectra(dat, method = "mean", spec1, spec2, spec3 = NULL,
            spec4 = NULL, spec5 = NULL, spec6 = NULL)
```

Arguments

`dat`	The mapped spectral data frame, containing `full_mz` in the first column.
`method`	A character string; the method used to combine the spectra. Methods include "sum" and "mean". Default = "mean."
`spec1`	A character string; the name of the column in `dat` containing the intensity data for the first spectrum to be averaged.
`spec2`	A character string; the name of the column in `dat` containing the intensity data for the second spectrum to be averaged. Default = NULL.
`spec3`	A character string; the name of the column in `dat` containing the intensity data for the third spectrum to be averaged. Default = NULL.
`spec4`	A character string; the name of the column in `dat` containing the intensity data for the fourth spectrum to be averaged. Default = NULL.
`spec5`	A character string; the name of the column in `dat` containing the intensity data for the fifth spectrum to be averaged. Default = NULL.
`spec6`	A character string; the name of the column in `dat` containing the intensity data for the sixth spectrum to be averaged. Default = NULL.

Value

Returns a new column in the input data frame containing the averaged intensity data.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Load sample dataset "Master.rda"
data("Master")

## Average blank spectrum 1 and 2 using the method "mean"
ex <- avgSpectra(Master, method = "mean", spec1 = "Blank1", spec2 = "Blank2")

## Average blank spectrum 1 and 2 using the method "sum"
ex <- avgSpectra(Master, method = "sum", spec1 = "Blank1", spec2 = "Blank2")
```

---

baselineCorr *Baseline Correction*

Description

Offers three different methods for baseline correction of raw spectral data. Methods include monotone minimum, linear interpolation, and LOESS curve fitting.

Usage

```
baselineCorr(dat, mass_dat, intensity_dat, method = NULL, n = NULL)
```

Arguments

`dat`	The name of the spectral data frame, containing `m/z` data in the first column and spectral intensity data in subsequent columns.
`mass_dat`	Character string. The name of the column in `dat` containing the *m/z* data for the spectrum that will be corrected.
`intensity_dat`	Character string. The name of the column in `dat` containing the intensity data for the spectrum to be baseline corrected.
`method`	Character string. The method that is to be used to perform the baseline correction. Either `"monotone"` for monotone minimum correction, `"linear"` for linear interpolation, or `"loess"` for LOESS curve fitting.
`n`	Single odd numeric value. If `method = "linear"`, the size of the window that should be used for linear interpolation of the spectral baseline.

Value

Returns a new data frame containing the baseline corrected spectral data.

Methods

monotone_min Identifies valleys in the spectrum by evaluating the slopes between adjacent intensity values. Moving from left to right, if slope of point A is greater than 0, the nearest point B with slope of less than 0 is located. Points in between A and B are recorded as potential peaks. If slope of A < 0, nearest point B with slope > 0 is located. Points between A and B are recorded as valleys. Each intensity value in the spectrum is then subtracted by the intensity of the nearest valley, correcting any irregularities in the baseline of the data.

linear Divides a spectrum into small segments and evaluates the mean of points in each segment as a valley (1). A baseline is then generated by linearly interpolating valleys across all small segments. In each segment, the intensity of each peak is subtracted by the intensity of the valley in that segment.

loess Divides a spectrum into small segments and evaluates the quantile in each segment. The baseline is estimated as follows. If the intensity of point A in a segment is less than the quantile, the intensity of point A is recorded as a baseline predictor for that point. If the intensity of point A is greater than or equal to the quantile in the segment, the quantile for the segment is recorded the baseline predictor for that point. Local polynomial regression fitting is applied to the predictor points. The baseline is then subtracted from the raw spectral signal.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi (1) Yang, C., He, Z. & Yu, W. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. BMC Bioinformatics 10, 4 (2009). https://doi.org/10.1186/1471-2105-10-4

See Also

smoothSpectrum, peakDet

Examples

```
## Load sample dataset "bsline"
data("bsline")

## Correct the baseline using the monotone minimum method
bsline_mono <- baselineCorr(bsline, "mass", "raw",
                      method = "monotone_min")

## Correct the baseline using the linear interpolation method
## Window size of 7
bsline_linear <- baselineCorr(bsline, "mass", "raw",
                      method = "linear", n = 7)
```

Before1 *Pairwise Spectral Data Frame from Sample*

Description

Raw spectral data in binary dataframe. The first column contains the *m/z* data for the spectrum, while the second contains the intensity data. Spectrum is obtained from a biological sample, prior to chemical intervention.

Usage

```
data("Before1")
```

Format

A data frame with 437372 observations on the following 2 variables.

`mass` A numeric vector; the *m/z* data for the spectrum.

`Intensity` A numeric vector; the intensity data for the spectrum.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(Before1)
```

Before2 *Pairwise Spectral Data Frame from Sample*

Description

Raw spectral data in binary dataframe. The first column contains the *m/z* data for the spectrum, while the second contains the intensity data. Spectrum is obtained from a biological sample, prior to chemical intervention.

Usage

```
data("Before2")
```

Format

A data frame with 213593 observations on the following 2 variables.

`mass` A numeric vector; the *m/z* data for the spectrum.

`Intensity` A numeric vector; the intensity data for the spectrum.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(Before2)
```

Blank1 *Pairwise Spectral Data from Matrix Blank*

Description

Raw spectral data in binary dataframe. The first column contains the *m/z* data for the spectrum, while the second contains the intensity data. Spectrum is a MALDI matrix blank.

Usage

```
data("Blank1")
```

Format

A data frame with 105717 observations on the following 2 variables.

`mass` A numeric vector; the *m/z* data for the spectrum.

`Intensity` A numeric vector; the intensity data for the spectrum.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(Blank1)
```

Blank2 *Pairwise Spectral Data from Matrix Blank*

Description

Raw spectral data in binary dataframe. The first column contains the *m/z* data for the spectrum, while the second contains the intensity data. Spectrum is a MALDI matrix blank.

Usage

```
data("Blank2")
```

Format

A data frame with 276704 observations on the following 2 variables.

`mass` A numeric vector; the *m/z* data for the spectrum.

`Intensity` A numeric vector; the intensity data for the spectrum.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(Blank2)
```

bsline *Raw Spectrum with Irregular Baseline*

Description

A spectrum acquired using ESI FT-ICR-MS with an irregular baseline.

Usage

```
data("bsline")
```

Format

A data frame with 23807 observations on the following 2 variables.

`mass` a numeric vector. The *m/z* data of the spectrum.

`raw` A numeric vector. The raw, unfiltered or corrected intensity of the spectrum.

References

https://github.com/wesleyburr/subMaldi/

Examples

```
data(bsline)
```

`createSpecDF` *Create Empty Spectral Data Frame*

Description

Creates an empty data frame to be used for mapping spectral data to a vector of *m/z* values. The vector of *m/z* values is stored in the first column and can be called by "`full_mz`".

Usage

```
createSpecDF(min_mz = 53.76, max_mz = 1100.00, res = 0.0001, dig = 4)
```

Arguments

`min_mz`	Single numeric value; minimum *m/z* value of the observed range. Default = 53.76.
`max_mz`	Single numeric value; upper end of *m/z* range observed in spectra. Default = 1100.
`res`	The resolution of peaks; a single numeric value indicating the step size from the minimum to maximum *m/z* bin. Also known as bin width. Default = 0.0001.
`dig`	Number of decimal places to round the *m/z* vector to; required for *m/z* values to match full *m/z* vector. Default = 4.

Value

Returns a data frame which can be used to map irregularly spaced spectral data to a set range of *m/z* values, contained in the first column of the frame.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Creating an empty spectrum with an m/z range of 500 to 2000 m/z, with a step size of 0.001
spec_df <- createSpecDF(min_mz = 500, max_mz = 2000, res = 0.001, dig = 3)

## Creating an empty spectrum with an m/z range of 100 to 1000,
# with a step size of 0.0001 and sample names
spec_df <- createSpecDF(min_mz = 100, max_mz = 1000, res = 0.0001, dig = 4)
```

find_max *Find Peak Maxima*

Description

Analyzes spectral data and returns a list of the most intense peak in each spectrum, including the *m/z* value associated with the peak. Able to determine the maxima of 1-6 spectra at once.

Usage

```
find_max(dat, mass_dat, spec1, spec2 = NULL, spec3 = NULL,
          spec4 = NULL, spec5 = NULL, spec6 = NULL)
```

Arguments

dat	The name of the spectral data frame, containing m/z data in the first column and spectral intensity data in subsequent columns.
mass_dat	A character string; the name of the column in dat containing the *m/z* data for the spectrum.
spec1	A character string; the name of the column in dat containing the intensity data for the spectrum to be analyzed.
spec2	A character string; the name of the column in dat containing the intensity data for the second spectrum to be analyzed. Default = NULL.
spec3	A character string; the name of the column in dat containing the intensity data for the third spectrum to be analyzed. Default = NULL.
spec4	A character string; the name of the column in dat containing the intensity data for the fourth spectrum to be analyzed. Default = NULL.
spec5	A character string; the name of the column in dat containing the intensity data for the fifth spectrum to be analyzed. Default = NULL.
spec6	A character string; the name of the column in dat containing the intensity data for the sixth spectrum to be analyzed. Default = NULL.

Value

Returns a data frame indidcating the most intense peaks of each input spectrum. Indicates the spectrum the data is from, the *m/z* value associated with the peak, and the intensity of the maxima.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Load sample dataset "Master.rda"
data("Master")

## Find maxima of a single spectrum
find_max(dat = Master, mass_dat = "full_mz", spec1 = "Blank1")

## Find maxima of four spectra
find_max(dat = Master, mass_dat = "full_mz", spec1 = "Blank1",
          spec2 = "Before1", spec3 = "After1", spec4 = "After2")
```

`find_max_set` *Find Most Intense Peak of a Set*

Description

Analyzes spectral data and returns information about the most intense peak in all of the spectral set. Indicates the most intense spectrum, the *m/z* value, and intesnity of the maximum. Allows for analysis of 2-6 spectra at once.

Usage

```
find_max_set(dat, mass_dat, spec1, spec2, spec3 = NULL,
              spec4 = NULL, spec5 = NULL, spec6 = NULL)
```

Arguments

`dat`	The name of the spectral data frame, containing `m/z` data in the first column and spectral intensity data in subsequent columns.
`mass_dat`	A character string; the name of the column in `dat` containing the *m/z* data for the spectrum.
`spec1`	A character string; the name of the column in `dat` containing the intensity data for the first spectrum to be analyzed.
`spec2`	A character string; the name of the column in `dat` containing the intensity data for the second spectrum to be analyzed.
`spec3`	A character string; the name of the column in `dat` containing the intensity data for the third spectrum to be analyzed. Default = NULL.
`spec4`	A character string; the name of the column in `dat` containing the intensity data for the fourth spectrum to be analyzed. Default = NULL.
`spec5`	A character string; the name of the column in `dat` containing the intensity data for the fifth spectrum to be analyzed. Default = NULL.
`spec6`	A character string; the name of the column in `dat` containing the intensity data for the sixth spectrum to be analyzed. Default = NULL.

Value

Returns a data frame indidcating the peak maximum of the spectral set. The sample number, *m/z* value, and intensity data are returned.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Load sample dataset "Master.rda"
data("Master")

## Find the most intense peak of two spectra
find_max_set(dat = Master, mass_dat = "full_mz", spec1 = "Before1", spec2 = "After1")

## Find the most intense peak of six spectra
find_max_set(dat = Master, mass_dat = "full_mz", spec1 = "After1", spec2 = "Blank2",
              spec3 = "Before1", spec4 = "Blank1", spec5 = "After2", spec6 = "Before2")
```

---

mapSpectrum *Map Spectrum to* m/z *Vector*

Description

Fills in the columns of the empty data frame created using `createSpecDF`. Mass to charge values from pairwise spectral data are compared to the `full_mz` vector. All peaks within the `thresh` of one another are binned, and only the maximum intensity of that bin is filled into the mapped spectrum.

Usage

```
mapSpectrum(dat, massCol, intenseCol, dig = 4, thresh = 1e-04, spec_df, colName)
```

Arguments

`dat`	A pairwise data frame containing your spectral data. Should contain two columns: one for *m/z* and one for intensity.
`massCol`	A character string; the name of the *m/z* column in `dat`.
`intenseCol`	A character string; the name of the intensity column in `dat`.
`dig`	Number of decimal places to round the *m/z* data to; must match the same value used in `createSpecDF` in order for the columns to fill. Default = 4.
`thresh`	Single numeric value; all *m/z* values within `thresh` of each other are binned under that with the maximum intensity. Default = 5e-5.
`spec_df`	An empty data frame created using `createSpecDF`.
`colName`	A character string; the name of the column that should be filled with the spectral data.

Value

Returns a vector that is used to fill in the `colName` column of the mapped spectral data frame.

Warning

It is important that the values for `thresh` and `dig` are equal to that of `res` and `dig` used in `createSpecDF()`. Otherwise the data will fail to map.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

See Also

`createSpecDF`

Examples

```
## Load sample dataset "Blank1.rda"
data("lank1")

## Create empty spectral data frame to map to
spec_df <- createSpecDF(min_mz = 53.76, max_mz = 1100, res = 0.0001, dig = 4)

## Map binary spectral data to empty spectral data frame
spec_df <- mapSpectrum(dat = Blank1, massCol = "mass", intenseCol = "Intensity",
            spec_df = spec_df, colName = "Sample", thresh = 1e-04, dig = 4)
```

`Master` *Master Data Frame (High Resolution)*

Description

A mapped spectral data frame created using `createSpecDF()` and `mapSpectrum()`. Contains the baseline corrected (by linear interpolation) intensity data for 6 spectra, with *m/z* data recorded up to 4 decimal places.

Usage

```
data("Master")
```

Format

A data frame with 980506 observations on the following 7 variables.

`full_mz` A numeric vector; the *m/z* data for all spectra in the data frame.

`Blank1` A numeric vector; the intensity data for the first spectrum of the dataset. A MALDI matrix blank spectrum.

`Blank2` A numeric vector; the intensity data for the second spectrum in the dataset. A MALDI matrix blank spectrum.

`Before1` A numeric vector; the intensity data for the third spectrum in the dataset. Spectrum obtained prior to chemical intervention.

`Before2` A numeric vector; the intensity data for the fourth spectrum in the dataset. Spectrum obtained prior to chemical intervention.

`After1` A numeric vector; the intensity data for the fifth spectrum in the dataset. Spectrum obtained after chemical intervention.

`After2` A numeric vector; the intensity data for the sixth spectrum in the dataset. Spectrum obtained after chemical intervention.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(Master)

## Separate samples
blanks <- select(Master, "full_mz", "Blank1", "Blank2")
precon <- select(Master, "full_mz", "Before1", "Before2")
postcon <- select(Master, "full_mz", "After1", "After2")
```

Master2 *Master Data Frame (Low Resolution)*

Description

A mapped spectral data frame created using `createSpecDF()` and `mapSpectrum()`. Contains the baseline corrected (by linear interpolation) intensity data for 6 spectra, with *m/z* data recorded up to 2 decimal places.

Usage

```
data("Master2")
```

Format

A data frame with 91346 observations on the following 7 variables.

`full_mz` A numeric vector; the *m/z* data for all spectra in the data frame.

`Blank1` A numeric vector; the intensity data for the first spectrum of the dataset. A MALDI matrix blank spectrum.

`Blank2` A numeric vector; the intensity data for the second spectrum in the dataset. A MALDI matrix blank spectrum.

`Before1` A numeric vector; the intensity data for the third spectrum in the dataset. Spectrum obtained prior to chemical intervention.

`Before2` A numeric vector; the intensity data for the fourth spectrum in the dataset. Spectrum obtained prior to chemical intervention.

`After1` A numeric vector; the intensity data for the fifth spectrum in the dataset. Spectrum obtained after chemical intervention.

`After2` A numeric vector; the intensity data for the sixth spectrum in the dataset. Spectrum obtained after chemical intervention.

Source

Yeh, K., Stock N. L., Burr, W. & Stotesbury, T. Preliminary analysis of latent fingerprints recovered from underneath bloodstains using Matrix-Assisted Laser Desportion/Ionization Fourier-Transform Ion Cyclotron Resonance Mass Spectrometry Imaging (MALDI FT-ICR MSI). Forensic Chemistry (2020). In press.

References

https://github.com/wesleyburr/subMaldi

Examples

```
data(Master2)
data(Master)

## Separate samples
blanks <- select(Master2, "full_mz", "Blank1", "Blank2")
precon <- select(Master2, "full_mz", "Before1", "Before2")
postcon <- select(Master2, "full_mz", "After1", "After2")
```

---

normSpectra *Normalize Spectral Data*

Description

Normalize spectral data to a common scale using several different methods. Capable of normalizing 1-6 spectra at once.

Usage

```
normSpectra(dat, mass_dat, method = NULL, norm_mz = NULL,
            upper = NULL, lower = NULL, spec1, spec2 = NULL,
            spec3 = NULL, spec4 = NULL, spec5 = NULL, spec6 = NULL,
            showHI = FALSE)
```

Arguments

`dat`	The name of the spectral data frame, containing `m/z` data in the first column and spectral intensity data in subsequent columns.
`mass_dat`	A character string; the name of the column in `dat` containing the *m/z* data for the spectrum.

method	A character string; the normalization method that should be used to process the data. See 'Methods' below for list of methods. Default = NULL.
norm_mz	Numeric. If method = "custom", the *m/z* peak to which the spectral intensity should be normalized to. Value should have the same number of decimal places as the *m/z* data in dat. If method = "custom_imprecise", this value must be given as a character string of numbers.
lower	Numeric. If method = "stdev", the lower *m/z* bound of the noise region of the spectrum.
upper	Numeric. If method = "stdev", the upper *m/z* bound of the noise region of the spectrum.
spec1	A character string; the name of the column in dat containing the intensity data for the spectrum to be analyzed.
spec2	A character string; the name of the column in dat containing the intensity data for the second spectrum to be normalized. Default = NULL.
spec3	A character string; the name of the column in dat containing the intensity data for the third spectrum to be normalized. Default = NULL.
spec4	A character string; the name of the column in dat containing the intensity data for the fourth spectrum to be normalized. Default = NULL.
spec5	A character string; the name of the column in dat containing the intensity data for the fifth spectrum to be normalized. Default = NULL.
spec6	A character string; the name of the column in dat containing the intensity data for the sixth spectrum to be normalized. Default = NULL.
showHI	Logical. To be used with method = "custom". If TRUE, intensity values greater than norm_mz will be kept. If FALSE, any peaks with intensity greater than norm_mz will be truncated to the intensity of norm_mz. Default = FALSE.

Value

Returns a new data frame including the original *m/z* data and normalized intensity data.

Methods

max Normalizes the intensity data of spectra to a scale of 0,1.

max_set Normalizes the intensity data of spectra to a scale of 0,1, where 1 is the single most intense peak of the spectral set.

custom Normalizes the intensity data of each input spectrum to the intensity of the selected *m/z* peak.

custom_imprecise Normalizes the intensity data of each input spectrum to the intensity of the selected *m/z* peak. Allows for less precise normalization; if data contains four decimal places, input *m/z* values can be input to 2 or 3 decimal places.

TIC Evaluates the sum of all intensities (TIC) of each spectrum in a dataset. If the TIC of all spectra are not equal, their intensities are multiplied by a normalization factor.

rel_TIC Evaluates the sum of all intensities (TIC) of each spectrum in a dataset. If the TIC of all spectra are not equal, their intensities are multiplied by a normalization factor. Each peak intensity is then divided by the normalized peak intensity so that each spectrum in the dataset has a TIC of 1.

RMS Normalizes each spectrum by dividing each intensity by the spectrum's RMS.

median Evaluates the median of each spectrum in a dataset after removing 0 values introduced by mapping. If the medians are not equal between spectra, a normalization factor is applied to the spectral intensities until all median intensities in the dataset are equal.

stdev Evaluates the standard deviation of intensity values within the same noisy region (a region lacking peaks) of each spectrum. All intensities in each spectrum are then divided by the spectrum's standard deviation in the noise region. All output spectra should have a standard deviation of 1 in the selected region.

quantile Normalizes the distributions of the values in each spectrum in a set. Sorts the intensity data of each spectrum and evaluates the average intensity for each rank. The intensity values are then replaces with the averaged intensities, rearranged in their original order.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Load sample dataset "Master2.rda"
data("Master2")

## Normalize spectrum "Before1" to its maximum intensity
ex <- normSpectra(dat = Master2, mass_dat = "full_mz",
            method = "max", spec1 = "Before1")

## Normalize the spectra "Before1" and "Before2" to the TIC
ex <- normSpectra(dat = Master2, mass_dat = "full_mz",
            method = "TIC", spec1 = "Before1", spec2 = "Before2")

## Normalize spectrum "After1" to the intensity of the peak at m/z 253.22
ex <- normSpectra(dat = Master2, mass_dat = "full_mz",
            method = "custom", norm_mz = 253.22, spec1 = "After1")
```

`peakDet` *Peak Detection*

Description

Two methods for peak detection in baseline corrected spectral data. Methods include signal-to-noise ratio and slopes of peaks.

Usage

```
peakDet(dat, mass_dat, intensity_dat, method = NULL, n = NULL,
SNR_thresh = NULL)
```

Arguments

dat	The name of the spectral data frame, containing m/z data in the first column and spectral intensity data in subsequent columns.
mass_dat	Character string. The name of the column in dat containing the *m/z* data for the spectrum.
intensity_dat	Character string. The name of the column in dat containing the intensity data for the spectrum.
method	Character string. The method of peak detection. Either "snr" for signal-to-noise ratio, or "slopes" for slopes of peaks.
n	Single numeric value. For both method = "snr" and method = "slopes", the window size used to calculate noise. Noise is defined as the median of the absolute deviation (MAD) of points within a window [1].
SNR_thresh	Single numeric value. When method = "snr", the signal-to-noise ratio (SNR) threshold for discarding peaks. If the SNR of a peak falls below this threshold, it will be discarded.

Value

Returns a new data frame containing only the peaks which have passed the detection criteria.

Methods

snr Each spectrum is divided into segments of size n. Noise is calculated as the median absolute deviation of points within each segment [1]. If the intensity of a peak divided by the noise in that segment is less than the indicated SNR_thresh, the peak is discarded.

slopes Uses the shapes of peaks to remove false peak candidates [1]. First, the left and right endpoints of each peak are identified on the baseline. Next, the slopes of each endpoint are evaluated. If the either the left or right slope are less than a defined threshold, the peak candidate is discarded [1]. The threshold is defined as half of the local noise level, or half of the median absolute deviation in a window of size n.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi (1) Yang, C., He, Z. & Yu, W. Comparison of public peak detection algorithms for MALDI mass spectrometry data analysis. BMC Bioinformatics 10, 4 (2009). https://doi.org/10.1186/1471-2105-10-4

See Also

smoothSpectrum, baselineCorr

Examples

```
## Load sample dataset "bsline"
data("bsline")

## Baseline correct using method "linear"
linear <- baselineCorr(bsline, "mass", "raw",
```

```
                        method = "linear", n = 7)

## Detect peaks using method "snr"
snr <- peakDet(linear, "mz", "baseline", method = "snr",
                  n = 7, SNR_thresh = 3)

## Detect peaks using method "slopes"
slopes <- peakDet(linear, "mz", "baseline", method = "slopes",
                    n = 7)
```

`plotgridSpectra` *Plot and Compare Multiple Spectra in a Grid*

Description

Plots 4 or 6 spectra in a grid format.

Usage

```
plotgridSpectra(dat, mass_dat, spec1, spec2, spec3, spec4, spec5 = NULL,
                spec6 = NULL, colour1 = "steelblue1", colour2 = "lightpink1",
                colour3 = "coral", colour4 = "goldenrod1", colour5 = "gold",
                colour6 = "seagreen3", span = 5, thresh = 0.1, lbls = FALSE,
                lbl.fmt = "%3.4f", columns = 2, min_mz = 0, max_mz = 10000,
               min_I = 0, max_I = NULL, x_ticks = 500, intensity_scale = "free_y")
```

Arguments

`dat`	The name of the spectral data frame, containing the *m/z* data in the first column.
`mass_dat`	A character string; the name of the column in `dat` containing the *m/z* data.
`spec1`	A character string; the name of the column in `dat` containing the intensity data for the first spectrum that is to be plotted.
`spec2`	A character string; the name of the column in `dat` containing the intensity data for the second spectrum to be plotted.
`spec3`	A character string; the name of the column in `dat` containing the intensity data for the third spectrum to be plotted.
`spec4`	A character string; the name of the column in `dat` containing the intensity data for the fourth spectrum to be plotted.
`spec5`	A character string; the name of the column in `dat` containing the intensity data for the fifth spectrum to be plotted. Default = NULL.
`spec6`	A character string; the name of the column in `dat` containing the intensity data for the sixth spectrum to be plotted. Default = NULL.
`colour1`	A character string indicating the colour the first spectrum should be plotted in.
`colour2`	A charater string indicating the colour of the second spectrum in the plot.
`colour3`	A charater string indicating the colour of the third spectrum in the plot.
`colour4`	A charater string indicating the colour of the fourth spectrum in the plot.
`colour5`	A character string indicating the colour of the fifth spectrum in the plot.
`colour6`	A character string indicating the colour of the sixth spectrum in the plot.

span	Single numeric value; the span of peak maximas in between each label. Default = 5 (ignores two peak maxima on either side of each label).
thresh	Single numeric value (0-100); the threshold of peak itensities which should be labelled. Default = 0.1.
lbls	Logical. If lbls = TRUE, labels indicating the *m/z* value of each peak maxima within the indicated span will be included in the output plot. If lbls = FALSE, labels are not shown. Default = FALSE.
lbl.fmt	Character string in the format "%a.bf", where *a* is the number of figures displayed to the left of decimal places in labels, and *b* is the number of figures displayed to the right of decimal places in labels.
columns	Single numeric value; either 2 or 3. The number of columns spectral plots should be displayed in.
min_mz	Single numeric value; minimum *m/z* value of the observed range.
max_mz	Single numeric value; upper end of *m/z* range observed in spectra.
min_I	Single numeric value; minimum intensity value to be plotted.
max_I	Single numeric value; upper end of the intensity range to be plotted.
x_ticks	Single numeric value; the space between ticks on x-axis.
intensity_scale	A character string; the method that should be used for determining the y-axis scales for each spectrum. If method = "free_y", each spectrum will be plotted with its own intensity scale. If method = "fixed", each spectrum will be plotted with the y-axis of the most intense spectrum in the set.

Value

Returns a line plot of the input spectra arranged in two columns.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Load sample dataset "Master.rda"
data("Master")

## Plot four spectra
plotgridSpectra(dat = Master, mass_dat = "full_mz",
                spec1 = "Before1", spec2 = "Before1",
                spec3 = "After1", spec4 = "After2", lbls = TRUE)

## Plot six spectra
plotgridSpectra(dat = Master, mass_dat = "full_mz",
                spec1 = "Blank1", spec2 = "Blank2",
                spec3 = "Before1", spec4 = "Before2", spec5 = "After1",
                spec6 = "After2", lbls = TRUE)
```

plotSpectra *Plot and Compare Multiple Spectra*

Description

Plots multiple spectra on the same *x*-axis scale for easier comparison. Allows for 2 to 6 spectra per plot, stacked on top of one another.

Usage

```
plotSpectra(dat, mass_dat, spec1, spec2 = NULL,
            spec3 = NULL, spec4 = NULL, spec5 = NULL,
            spec6 = NULL, colour1 = "steelblue1",
            colour2 = "lightpink1", colour3 = "coral",
            colour4 = "goldenrod1", colour5 = "gold",
            colour6 = "seagreen3", span = 5, thresh = 0.1,
            lbls = FALSE, lbl.fmt = "%3.4f", min_mz = 0,
            max_mz = 10000, min_I = 0, max_I = NULL,
            x_ticks = 100, intensity_scale = "free_y")
```

Arguments

dat	The name of the spectral data frame, containing the *m/z* data in the first column.
mass_dat	A character string; the name of the column in dat containing the *m/z* data.
spec1	A character string; the name of the column in dat containing the intensity data for the first spectrum that is to be plotted.
spec2	A character string; the name of the column in dat containing the intensity data for the second spectrum to be plotted.
spec3	A character string; the name of the column in dat containing the intensity data for the third spectrum to be plotted.
spec4	A character string; the name of the column in dat containing the intensity data for the fourth spectrum to be plotted.
spec5	A character string; the name of the column in dat containing the intensity data for the fifth spectrum to be plotted.
spec6	A character string; the name of the column in dat containing the intensity data for the sixth spectrum to be plotted.
colour1	A character string indicating the colour the first spectrum should be plotted in.
colour2	A charater string indicating the colour of the second spectrum in the plot.
colour3	A charater string indicating the colour of the third spectrum in the plot.
colour4	A charater string indicating the colour of the fourth spectrum in the plot.
colour5	A charater string indicating the colour of the fifth spectrum in the plot.
colour6	A charater string indicating the colour of the sixth spectrum in the plot.
span	Single numeric value; the span of peak maximas in between each label. Default = 5 (ignores two peak maxima on either side of each label).
thresh	Single numeric value (0-100); the threshold of peak itensities which should be labelled.

lbls	Logical. If lbls = TRUE, labels indicating the *m/z* value of each peak maxima within the indicated span will be included in the output plot. If lbls = FALSE, labels are not shown.
lbl.fmt	Character string in the format "%a.bf", where *a* is the number of figures displayed to the left of decimal places in labels, and *b* is the number of figures displayed to the right of decimal places in labels. Default = "%3.4f".
min_mz	Single numeric value; minimum *m/z* value of the observed range.
max_mz	Single numeric value; upper end of *m/z* range observed in spectra.
min_I	Single numeric value; minimum intensity value of the observed range.
max_I	Single numeric value; upper end of the intensity range observed in spectra.
x_ticks	Single numeric value; the space between ticks on x-axis.
intensity_scale	A character string; the method that should be used for determining the y-axis scales for each spectrum. If method = "free_y", each spectrum will be plotted with its own intensity scale. If method = "fixed", each spectrum will be plotted with the y-axis of the most intense spectrum in the set.

Value

Returns a line plot of the input spectra.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Plotting using the sample dataset "Master.rda"
data("Master")
plotSpectra(dat = Master, mass_dat = "full_mz",
            spec1 = "Blank1", spec2 = "Blank2",
            intensity_scale = "free_y", lbls = TRUE)
```

---

plotSpectrum *Plot a Single Spectrum*

Description

Plots a single spectrum from a mapped spectral data frame.

Usage

```
plotSpectrum(dat, mass_dat, intensity_dat, colour = "black",
             span = 5, thresh = 0.1, min_mz = min(mass_dat),
             max_mz = max(mass_dat),  min_I = 0,
             max_I = max(intensity_dat), lbls = FALSE,
             lbl.fmt = "%3.4f", x_ticks = 500)
```

Arguments

dat	The mapped spectral data frame, containing full_mz in the first column.
mass_dat	The name of the column in dat containing the *m/z*data for the spectrum that is to be plotted. Should be input as dat$mass_column. By default, the mass column in a mapped data frame is "full_mz". If our spectral data frame is called my_spec, then our input for mass_dat = my_spec$full_mz.
intensity_dat	The name of the column in dat containing the intensity data for the spectrum that is to be plotted. Should be input as dat$intensity_column, as above for mass_dat.
colour	A character string; the name of the colour the spectrum should be plotted in.
span	Single numeric value; the distance between peak maxima that are labelled if lbls = TRUE. Default = 5.
thresh	Single numeric value between 0 and 1; the intensity threshold of peaks in the plotted spectrum which should be labelled. Default = 0.1.
min_mz	Single numeric value; minimum *m/z* value of the observed range.
max_mz	Single numeric value; upper end of *m/z* range to be observed in the plotted spectra.
min_I	Single numeric value; minimum intensity value of the spectrum.
max_I	Single numeric value; upper end of the intensity range observed in spectra.
lbls	Logical; should labels indicating the *m/z* of the most intense peaks be included in the plot?
lbl.fmt	Character string in the format "%a.bf", where *a* is the number of figures displayed to the left of decimal places in labels, and *b* is the number of figures displayed to the right of decimal places in labels.
x_ticks	Single numeric value; the increment between ticks on the x-axis.

Value

Returns a line plot of the input spectrum.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

Examples

```
## Plotting using the sample dataset "Before1.rda"
data("Before1")

plotSpectrum(dat = Before1, mass_dat = Before1$mass,
             intensity_dat = Before1$Intensity)
```

readcsvDir *Import .csv Spectra from a Directory*

Description

Imports all `.csv` files in a directory, turns them into a binary data frame containing a "mass" and "Intensity" column, and outputs them as `.rda` files into an output directory. Allows for rapid import of many spectral datasets at once.

Usage

```
readcsvDir(direct, massCol, intenseCol, output)
```

Arguments

`direct`	The path to the directory where the `.csv` files are held.
`massCol`	A character string; the name of the mass column in the `.csv` files.
`intenseCol`	A character string; the name of the intensity column in the `.csv` files.
`output`	The path to the directory where the exported `.rda` files containing the binary spectral data frames should go.

Value

Returns a directory of `.rda` files containing binary spectral data frames into the `output` file path.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

readcsvSpec *Import a Single .csv Spectrum*

Description

Imports a single spectrum in .csv format and turns it into a binary data frame containing a "mass" and "Intensity" column.

Usage

```
readcsvSpec(spec_file, massCol, intenseCol)
```

Arguments

`spec_file`	A character string; the file path to the `.csv` spectrum that is to be imported.
`massCol`	A character string; the name of the mass column in the spectrum's `.csv` file.
`intenseCol`	A character string; the name of the intensity column in the spectrum's `.csv` file.

Value

Returns a binary data frame containing the imported data in a *m/z* column denoted "mass", and an intensity column denoted "Intensity".

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

`rmveEmpty` *Remove Empty Rows*

Description

Data frames created using createSpecDF can have thousands to millions of rows, especially when dealing with high resolution mass spectrometry data. This can be quite taxing to the speed of executing certain functions, especially when it comes to visualizing data. This function removes empty rows to reduce the computational load and increase the ease of use of functions on the data frame. Any row where all elements (but the first, which is `full_mz`) equal 0 are removed from the data frame.

Usage

```
rmveEmpty(dat)
```

Arguments

`dat`	The spectral data frame, containing `full_mz` in the first column, to be trimmed of all-zero rows.

Value

Returns the input data frame without its all-zero rows.

Note

Avoid truncating a dataframe until all samples to be compared have been mapped.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

See Also

`createSpecDF`,`mapSpectrum`

Examples

```
## Load sample dataset "Mastre.rda"
data("Master")

## Select only the spectra "Before1" and "Before2"
ex <- select(Master, "full_mz", "Before1", "Before2")

## Use rmveEmpty(x) on those data frames to reduce computational load for use with
# other functions and packages
ex <- rmveEmpty(dat = ex)
```

`smoothSpectrum` *Smooth Noise in Spectrum*

Description

Offers two different methods for smoothing noise in raw spectral data: a moving average filter and the Savitzky-Golay filter (1).

Usage

```
smoothSpectrum(dat, mass_dat, intensity_dat, method = NULL,
                           p = NULL, n = NULL, m = 0, ts = 1)
```

Arguments

`dat`	The name of the spectral data frame, containing `m/z` data in the first column and spectral intensity data in subsequent columns.
`mass_dat`	A character string; the name of the column in `dat` containing the *m/z* data for the spectrum.
`intensity_dat`	A character string; the name of the column in `dat` containing the intensity data for the spectrum to be smoothed.
`method`	A character string; the method to be used for smoothing. Available methods include a Savitzky-Golay filter (`"sgolay"`) and a moving average filter (`"mov_avg."`)
`p`	Single numeric value. If `method = "sgolay"`, the filter order of smoothing. Default = NULL.
`n`	Single odd numeric value. If `method = "sgolay"`, the length of the smoothing filter. If `method = "mov_avg"`, the window span size. Default = NULL.
`m`	Single numeric value. If `method = "sgolay"`, returns the m-th derivative of the filter coefficients. Default = 0.
`ts`	Single numeric value. If `method = "sgolay"`, the time scaling factor. Default = 1.

Value

Returns a new data frame containing the smoothed spectral data.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi (1) A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least-squares procedures, Anal. Chem. 36 (8) (1964) 1627-1639.

Examples

```
## Load sample dataset "Before1.rda"
data("Before1")

## Testing method "sgolay"
test <- smoothSpectrum(dat = Before1, mass_dat = "mass",
                       intensity_dat = "Intensity",
                       method = "sgolay", p = 4,
                       n = 25, m = 0)

## Testing method "mov_avg"
test <- smoothSpectrum(dat = Before1, mass_dat = Before1$mass,
                       intensity_dat = Before1$Intensity,
                       method = "mov_avg", n = 25)
```

---

subSpectra *Subtract Blank Peaks from Sample Spectra*

Description

This function takes the intensity values in the blank column of the mapped spectral data frame and subtracts them from the intensity values in the sample column. If the intensity value in the blank spectrum is greater than that in the sample spectrum, the intensity of the sample peak is given an intensity of 0.

Usage

```
subSpectra(dat, Blank_Var, Sample, Sub_Sample, showNeg = FALSE)
```

Arguments

dat	The spectral data frame, containing full_mz in the first column and intensity data in the subsequent columns.
Blank_Var	A character string; the name of the blank column that will be subtracted from the sample column.
Sample	A character string; the name of the sample column that the blank data will be subtracted from.
Sub_Sample	A character string; the name of the column to be filled with the subtracted spectrum.
showNeg	Logical; if showNeg = TRUE, then negative values produced by the subtraction will be kept in the dataset. If showNeg = FALSE, then negative values created by the subtraction are set to 0 in the output dataset.

Value

Returns a vector of intensity values that are filled into the `Sub_Sample` column of the mapped data frame at the corresponding rows of `full_mz`.

Author(s)

Kristen Yeh <kristenyeh@trentu.ca> Wesley Burr <wburr@trentu.ca>

References

https://github.com/wesleyburr/subMaldi

See Also

`createSpecDF`, `mapSpectrum`

Examples

```
## Load sample datasets "Master.rda"
data("Master")

## Make subtraction column for new data
Master <- transform(Master, "Subtracted" = 0)

## Subtract Blank1 from Before1
Master <- subSpectra(dat = Master, Blank_Var = "Blank1",
                     Sample = "Before1", Sub_Sample = "Subtracted")
```

Index