PROcess

March 24, 2012

`align` *Align peaks for a specified precision.*

Description

A vector of peaks are expanded to a collection of intervals, 'eps' * m/z of the peak to the left and right of the peak position. They are then aligned using the method of intersection graphs by Gentleman and Vandal.

Usage

```
align(pVec, eps = 0.003)
```

Arguments

`pVec`	A vector of peaks.
`eps`	A user specified precision of peak position.

Value

A vector of aligned peaks.

References

R. Gentleman and A. C. Vandal, Computational Algorithms for Censored-Data Problems Using Intersection Graphs. (2001), J. Comput. Graphic Statist., vol. 10, p403–421.

aveSpec — *Compute mean spectrum of a set of spectra*

Description

Compute mean spectrum of a set of spectra.

Usage

```
aveSpec(nVec)
```

Arguments

nVec	A character vector of file names.

Details

'aveSpec' computes the point-wise mean of intensities of a set of spectra over the whole range of the m/z values.

Value

A matrix with 2 columns, the first being the m/z values and the 2nd being the average of intensities corresponding to the m/z value in the first column of the same row.

Author(s)

Xiaochun Li

Examples

```
testfs <- dir(system.file("Test", package = "PROcess"),
full.names=TRUE)
testAve <- aveSpec(testfs)
```

avesd — *Compute the Average Standard Deviation for a Set of Spectra and a Given Cutoff Point*

Description

Compute the Average Standard Deviation for a Set of Spectra and a Given Cutoff Point

Usage

```
avesd(Ma, cutoff)
```

Arguments

Ma	A matrix with rows the m/z values and columns samples/spectra
cutoff	A number of m/z

Details

For a given cutoff point, the standard deviation of all spectra will be calculated at each m/z above this cutoff point. The resulting sd's are then averaged and returned.

Value

A real number.

`binning` *Binning the Mass Spectrometry Data*

Description

The function 'binning' takes a matrix of spectra and constructs a matrix of intensities of reduced dimensions based on a equally spaced mesh of 'breaks' intervals over the entire m/z range.

Usage

```
binning(Ma, breaks = 400)
```

Arguments

`Ma`	a matrix of spectra by column.
`breaks`	number of bins, or equally-spaced intervals for the image of reduced resolution

Value

A matrix of dimensions c(dim(Ma)[1], 'breaks').

See Also

See Also `gelmap`.

`bslnoff` *Baseline Substraction*

Description

This function estimates the baseline and then removes baseline from the raw spectrum.

Usage

```
bslnoff(f, breaks = 200, qntl = 0, method = c("loess", "approx"), bw = 0.005, pl
```

Arguments

f	a matrix with M/Z values in the first column and intensities in the second column
breaks	number of breaks set to M/Z values for finding the local minima or points below a centain quantile of intensities; breaks -1 equally spaced intervals on the log M/Z scale.
qntl	if 0, find local minima; if >0 find intensities < qntl*100th quantile locally.
method	"loess" or "approx" (linear interpolation).
bw	the bandwidth to be passed to loess.
plot	TRUE or FALSE, if true, it will plot the raw spectrum, theestimated baseline and the baseline substracted spectrum.
...	Further parameters that get passed on to plot.

Value

a matrix of two columns: the first column being the M/Z values same as the input, and the second column being the baseline substracted spectra.

Author(s)

Xiaochun Li

Examples

```
fdat <- system.file("Test", package = "PROcess")
fs <- list.files(fdat, pattern="\\.*csv\\.*", full.names=TRUE)
f1 <- read.files(fs[1])
fcut <- f1[f1[,1]>0,]
bseoff <-bslnoff(fcut,method="loess",plot=TRUE, bw=0.1)
title(basename(fs[1]))
```

---

gelmap *Plot the image of a set of spectra*

Description

This function takes its argument as a intensity matrix of a set of spectra and plots the image as a heatmap.

Usage

```
gelmap(Ma, cols = gray(seq(1, 0, by = -0.01)), at.mz = NULL, at.col = NULL, cexC
```

Arguments

Ma	a matrix of intensities; spectra are arranged column-wise.
cols	a vector of color to represent continuum of intensities.
at.mz	a vector of m/z values where labels are desired.
at.col	a vector of sample indices, useful to label particular samples. If NULL, samples will be labelled from down up by 1:ncol(Ma).
cexCol	text size of the sample labels.

See Also

See Also binning.

`getMzs` *Extract M/Z values from the biomarker dataframe.*

Description

Turn column names of the biomarker dataframe to numeric M/Z values.

Usage

```
getMzs(df)
```

Arguments

`df`	The biomarker dataframe with rows as spectra and columns as biomarkers.

Value

A numeric vector.

`getPeaks` *Peak Detection*

Description

For given threshold criteria, find peaks.

Usage

```
getPeaks(bseoffM, peakinfofile,SoN = 2,span = 81,sm.span=11,
zerothrsh=2, area.w = 0.003, ratio = 0.2)
```

Arguments

`bseoffM`	a matrix holding the baseline-substracted spectra, with row-names as the m/z values and column-names as the spectrum names.
`peakinfofile`	a '.csv' file in the same format as Ciphergen's peak info file, with 5 columns data. More details later.
`SoN`	see isPeak().
`span`	see isPeak().
`sm.span`	see isPeak().
`zerothrsh`	ignore peaks whose intensity values are below zerothrsh.
`area.w`	see isPeak().
`ratio`	see isPeak().

Details

For given threshold criteria, detect peaks and write the following columns of information into 'peakinfofile', spectrum name (Spectrum.Tag), spectrum sequential number (Spectrum.), peak sequential number within a spectrum (Peak.), relative intensity (Intensity) and the m/z value where the relative intensity occurs (Substance.Mass).

Author(s)

Xiaochun Li

See Also

rmBaseline

Examples

```
example(renorm)
peakfile <- paste(tempdir(),"testpeakinfo.csv", sep="/")
getPeaks(rtM, peakfile)
```

---

getPeaks2 *Quantify peaks for individual spectra.*

Description

For a vector of given peak locations, quantify peaks for individual spectra.

Usage

```
getPeaks2(bseoffM, pVec, eps = 0.003)
```

Arguments

bseoffM	A matrix of intensities, with rows the m/z values and columns samples.
pVec	A vector of given peak locations.
eps	Relative precision of peak location.

Details

Each peak is expanded to an interval, eps * m/z to the left and right of the peak. Intensities of individual spectra are quantified by the maxima in this interval.

Value

A matrix of intensities with rows the peaks 'pVec' and column the samples. The m/z values of 'pVec' is stored as the 'rownames' of the returned matrix.

intg *Integration*

Description

This function calculates the integration of y with respect to x.

Usage

```
intg(y, x)
```

Arguments

x	a vector of real values, not necessarily sorted.
y	a vector of function values at x.

Details

(x,y)s are sorted according to ascending x values and then the integration is calculated as sum of the products of average of adjacent y values and the difference of adjacent x values.

Author(s)

Xiaochun Li

Examples

```
x <- seq(0, 1,length=100)
y <- x^2
intg(y, x)
```

is.multiple *Find multiple-charged polypeptides.*

Description

For each of the polypeptides in a vector, find its multiple-charged species in the vector.

Usage

```
is.multiple(v, k = 2, eps = 0.003)
```

Arguments

v	a vector of polypeptides.
k	a vector of integers, that is, multiples of interest.
eps	a user specified precision of peak position.

Details

If abs(v-k*u)/v <= eps, then v is considered to be u with k charges.

Value

A list named with the m/z values of polypeptides who have multiple-charged species; each component is a named vector of polypeptides with number of charges as names.

Examples

```
bmks <- c(2360.25, 2666.34, 3055.72, 3058.04,
3776.94, 3778.24, 3779.53, 4712.37,7559.76, 4587.03,
4589.88, 9155.59, 13298.7)
is.multiple(bmks, k=2:5)
```

---

isPeak *Locate Peaks in a Spectrum*

Description

Find local maxima of a spectrum.

Usage

```
isPeak(f,SoN = 2,span = 81,sm.span=11,plot=FALSE,add = FALSE,
zerothrsh=2,area.w = 0.003, ratio = 0.2, ...)
```

Arguments

f	a matrix of two columns representing a spectrum, with the first column the m/z value and second the intensity
SoN	signal to noise ratio criterion for peak detection
span	parameter for estimating local variance before peak detection; default is 81 points, that is, 40 points to the left and right of a point of which the variance is being estimated.
sm.span	parameter for smoothing the spectrum before peak detection; default is 11 points, that is, 5 points to the left and right of a point being smoothed
plot	logical, plot the smoothed spectrum and peaks?
add	add to the existing raw and baseline-substracted plot?
zerothrsh	ignore peaks whose intensity values are below zerothrsh.
area.w	the neighbourhood of a peak m/z, mz*(1-area.w, 1+area.w).to calculate area of the peak,
ratio	if area/max(area) > ratio, the peak is retained.
...	further arguments that get passed on to plot.

Details

A spectrum is smoothed first using the nearest 'span' neighbours. A larger span 'sm.span' is needed to estimate the local variation so that it is not overestimated due to the peaks nearby. Then potential peaks are identified using Ripley's 'peaks' function with 'span' points.

Peaks that satify the conditions that the (smoothed) signal to noise ratio is greater than 'SoN' and that the smoothed signal is greater than 1.64 * mad(sm) are returned.

Value

A data frame with five components, 'peak','smooth','mz' and 'sigmas', each of length the number of rows in 'f'. 'peak' is logical indicating whethere there is a peak or not (Y/N), 'smooth' the smooth of the spectrum, 'mz' the same as 'f[,1]', 'sigmas' the estimates of local variation and 'area' the area associated with each peak after the first pass.

Author(s)

Xiaochun Li

Examples

```
example(bslnoff)
pkobj <- isPeak(bseoff,span=81,sm.span=11,plot=TRUE)
```

---

lnn *Estimate Signal and Variation of a Spectrum*

Description

Estimate the signal and the variation of a spectrum.

Usage

```
lnn(x, span = 21, sm.span = 11)
```

Arguments

x	a vector of real values.
span	the window widith for estimation of local variation.
sm.span	the window width for estimation of the signal of x.

Details

The signal of a spectrum is estimated by moving average and its local variation is estimated by moving 'mad', possibly in a large window.

Value

A list with two components:

fitted	estimated signal,
sigma	estimated local variation.

peaks *Peak Detection*

Description

Finds the local maxima, local noise and its associated standard deviations in a vector.

Usage

```
peaks(x, span = 3)
noise(x, span = 5)
sigma(x, span = 5)
```

Arguments

x	a vector.
span	a local miximum is defined as an element in a sequence which is greater than all other elements within a window of width 'span' centered at that element. The default value is 3, meaning that a peak is bigger than both of its neighbors. Local noise is definedas an element minus the mean of all elements within a window of width 'span' centered at that element. Local standard deviation of an element is defined as the standard deviation of all elements within a window of width 'span' centered at that element.

Value

a logical vector of the same length as 'series' indicating where the peaks are.

Author(s)

Xiaochun Li

Examples

```
x <- seq(0, 10*pi, by=0.1)
y <- sin(x)*x
plot(x,y, type="l")
is.max <- peaks(y)
points(x[is.max],y[is.max], pch=21, bg="red")
legend(2, 25, legend = "Peaks",pch = 19, col="red", bty = "n")

# can be used for local minima too:
# is.min <- peaks(-y)
# points(x[is.min],y[is.min], pch=21, bg="blue")
```

pk2bmkr *Find Biomarkers.*

Description

Align peaks of spectra in 'peakinfofile' and find biomarkers by a procedure described in Gentleman and Geyer (1994).

Usage

```
pk2bmkr(peakinfofile, bseoffM, bmkfile, eps = 0.003, binary = F,p.fltr = 0.1)
```

Arguments

peakinfofile	a '.csv' file in the same format as Ciphergen's peakinfo file with 5 columns data, Spectrum.Tag, Spectrum., Peak., Intensity and Substance.Mass.
bseoffM	a matrix holding the baseline-substracted spectra, with row-names as the m/z values and column-names as the spectrum names.
bmkfile	a '.csv' file in the same format as Ciphergen's biomarker file, with spectra (samples) as columns, and biomarkers as rows.
eps	expected experimental variation in the m/z values.
binary	output intensity or binary peak presence/absence signals.
p.fltr	a number between 0 and 1. If a proto-biomarker is identified as peak in > p.fltr x 100 percent of spectra, it's kept in 'bmkfile'.

Value

A dataframe with spectra as rows and biomarkers as columns. Spectrum labels and biomarker positions may be in the names of the dataframe.

Author(s)

Xiaochun Li

References

Gentleman, R. and Geyer, C.J. (1994). Maximum likelihood for interval censored data: Consistency and computation. Biometrika, 81:618–623.

See Also

rmBaseline,getPeaks

Examples

```
example(getPeaks)
bmkfile <- paste(tempdir(),"testbiomarker.csv",sep="/")
testBio <- pk2bmkr(peakfile, rtM, bmkfile)

## plot biomarker intensities of the 2 spectra
```

```
mzs <- as.numeric(rownames(rtM))
matplot(mzs, rtM, type="l", xlim=c(1000, 10000))

bks <- getMzs(testBio)
abline(v=bks, col="green")
```

`quality` *Quality Check on a Set of Spectra*

Description

Compute three quality parameters for a set of spectra.

Usage

```
quality(Ma, peakinfofile, cutoff)
```

Arguments

`Ma`	a Matrix where the baseline-subtracted spectra are stored column wise.
`peakinfofile`	a '.csv' file in the same format as Ciphergen's peak info file, with 5 columns data. See Details of `getPeaks`.
`cutoff`	The point in m/z below which spectra are cutoff.

Details

The quality parameters are computed a la fashion de Mani. 1. Estimate noise by moving average with a 5 point window. 2. Estimate the noise envelop by 3 times the standard deviation of noise in a 251 point moving window. 3. Compute the area under the baseline-subtracted curve, area0. 4. Compute the area after subtracting the noise envelop from the baseline-subtracted curve, area1. 5. Parameter 'Quality' is defined as area1/area0. 6. Parameter 'Retain' is defined as the number of points with height above 5 times the noise envelop over total number of points in the spectrum. 7. Detect peaks in each spectrum by `getPeaks` or Ciphergen software. 8. Parameter 'peak' is defined as the number of peaks in a spectrum divided by the mean number of peaks across spectra.

A spectrum is considered to be of poor quality if Quality<0.4, Retain<0.1 and peak<0.5 simultaneously.

Value

A matrix with three named columns, 'Quality','Retain' and 'peak', with spectrum file names as row names.

Author(s)

Xiaochun Li

Examples

```
example(getPeaks)
qualRes <- quality(testM, peakfile, cutoff=1500)
```

read.files *Read a Spectrum from a Comma Deliminated File*

Description

Read a Spectrum from a Comma Deliminated File, maybe compressed.

Usage

```
read.files(fn)
```

Arguments

fn	path to a '.csv' file, possibly compressed.

renorm *Renormalize Spectra*

Description

Renormalize spectra for m/z values greater than 'cutoff'.

Usage

```
renorm(Ma, cutoff)
```

Arguments

Ma	a matrix, with rows the m/z values and the columns the samples.
cutoff	a real value, before which the portion of a spectrum will be ignored.

Details

A sample of spectra will be normalized to have the same AUC, the median of the AUCs of spectra. Each AUC is calculated as the sum of the intensities whose m/z values are greater than 'cutoff'.

Value

A matrix, with rows the m/z values and the columns the samples. Only rows with m/z values greater than 'cutoff' are kept.

Examples

```
example(rmBaseline)
rtM <- renorm(testM, cutoff=1500)
```

`rmBaseline` *Batch Baseline Subtraction.*

Description

Baseline subtraction from each raw spectrum in 'fldr'.

Usage

```
rmBaseline(fldr, bseoffrda = NULL, breaks = 200, qntl = 0,
  method = "loess", bw = 0.1,
  SpecNames = list.files(fldr, pattern = "\\.*csv\\.*"))
```

Arguments

`fldr`	a path to where the raw spectra are stored
`bseoffrda`	optional; name of the file (with extension .rda) where the baseline-substracted spectra, a matrix with row-names as the m/z values and column-names as the spectrum tags, will be saved to.
`breaks`	see bslnoff().
`qntl`	see bslnoff().
`method`	see bslnoff().
`bw`	see bslnoff().
`SpecNames`	a vector of character strings as spectrum names.

Value

A matrix whose columns correspond to baseline-subtracted spectra with row-names as the m/z values and column-names as the spectrum names.

Author(s)

Xiaochun Li

See Also

'bslnoff'.

Examples

```
testdir <- system.file("Test", package = "PROcess")
testM <- rmBaseline(testdir)
```

specZoom *Plotting a Spectrum with Peaks*

Description

Function for plotting an object returned by `isPeak`.

Usage

```
specZoom(pks, xlim = NULL, cols = c("cyan", "red", "black"), ...)
```

Arguments

`pks`	an object (a list) returned by `isPeak`.
`xlim`	a range of m/z values over which a zoomed-in view of the spectrum is desired.
`cols`	a vector of color specification for the smooth (signal), peaks and local noise.
`...`	further arguments that get passed on to `plot`.

Examples

```
example(isPeak)
specZoom(pkobj, xlim=c(5000, 10000))
```

Index