Package 'ChemoSpec'

December 2, 2009

Type Package

Title Exploratory Chemometrics for Spectroscopy

Version 1.31

Date 2009-11-30

Author Bryan A. Hanson DePauw University, Greencastle Indiana USA

Maintainer Bryan A. Hanson <hanson@depauw.edu>

Description A collection of functions for plotting spectra (NMR, IR etc) and carrying out various forms of exploratory data analysis, such as HCA and PCA. The design allows comparison of data from samples which fall into groups such as treatment vs. control. Robust methods appropriate for this type of high-dimensional data are available. ChemoSpec is designed to be very user friendly and suitable for people with limited background in R.

License GPL-3

LazyLoad yes

Depends chemometrics, robustbase, RColorBrewer, plyr, pcaPP, mvtnorm, mvoutlier, pls, lattice, grid, rgl

URL http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

R topics documented:

`ChemoSpec-package` *Exploratory Chemometrics for Spectroscopy*

Description

A collection of functions for plotting spectra (NMR, IR etc) and carrying out various forms of exploratory data analysis, such as HCA and PCA. The design permits comparison of data from samples which fall into groups such as treatment vs. control. Robust methods appropriate for this type of high-dimensional data are available. ChemoSpec is designed to be very user friendly and suitable for people with limited background in R. A vignette illustrating typical operations is available.

Details

Package:	ChemoSpec
Type:	Package
Version:	1.31
Date:	2009-11-30
License:	GPL-3
LazyLoad:	yes

Author(s)

Bryan A. Hanson, DePauw University, Greencastle Indiana USA

Maintainer: Bryan A. Hanson <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

`binBuck` *Bin or Bucket a Spectra Object*

Description

This function will bin a `"Spectra"` object by averaging every `bin.ratio` frequency values, and summing the corresponding intensity values. The net effect is a smoothed and smaller data set. If there are gaps in the frequency axis, each data chunk is processed separately. Note: some folks refer to binning as bucketing.

Usage

```
binBuck(spectra, bin.ratio)
```

Arguments

`spectra`	An object of S3 class `"Spectra"` to be binned.
`bin.ratio`	An integer giving the binning ratio, that is, the number of points to be grouped together into one subset of data.

Details

If the frequency range is not divisible by bin.ratio to give a whole number, data points are removed from the beginning of the frequency data until it is, and the number of data points removed is reported at the console. If there are gaps in the data where frequencies have been removed, each data chunk is processed separately. The algorithm forces the requested bin.ratio to be used for each portion of the data.

Value

An object of S3 class `"Spectra"`.

Called by

Top level function, called by user.

Calls

`binData`, `check4Gaps`, `chkSpectra`

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
sumSpectra(CuticleIR)
res <- binBuck(CuticleIR, bin.ratio = 4)
sumSpectra(res)
```

`binData` *Bin or Bucket Data*

Description

This function accepts a vector of x-values and averages them in groups of `bin.ratio` data points. It also accepts a vector of y-values and sums them in groups of `bin.ratio` data points. Both x and y data can be processed in the same call, or they can be processed separately.

Usage

```
binData(x = NULL, y = NULL, bin.ratio = 2)
```

Arguments

`x`	An optional vector of x values to be averaged in groups of `bin.data` points.
`y`	An optional vector of y values to be summed in groups of `bin.data` points.
`bin.ratio`	An integer giving the binning ratio, that is, the number of points to be grouped together into one subset of data.

Details

The x and y values must be contiguous in the sense that there are no gaps in the values (i.e., x[n + 1] - x[n] must be the same for the entire data set; this can be checked with `diff` and is checked internally. If length(x or y) is not divisible by bin.ratio to give a whole number, data points are removed from the beginning of x or y until it is, and the number of data points removed is reported at the console. The algorithm forces the requested bin.ratio to be used.

Value

As appropriate, a data.frame containing the following elements:

`mean.x`	A vector of the averaged x values. Length will be approximately length(x)/bin.ratio, with length(x) adjusted as described above if this does not give a whole number.
`sum.y`	A vector of the summed y values. Length will be approximately length(y)/bin.ratio, with length(y) adjusted as described above if this does not give a whole number.

Called by

`binBuck`

Calls

`isWholeNo`, `check4Gaps`

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
x <- seq(1, 50, 1); y <- rnorm(50)
res <- binData(x, y)
length(res$mean.x) # will be half of the original length
length(res$sum.y)
```

---

check4Gaps *Check for Missing Values (Gaps)*

Description

This function may be used with a Spectra object to see if there are any gaps or discontinuities in the frequency axis. Gaps may arise when unwanted frequencies are removed (e.g, water peaks in 1H NMR, or uninteresting regions in any kind of spectroscopy). As written, it can be used to check for gaps in any appropriate numeric vector. Not normally called directly by the user, but may be (see examples). A plot of the gaps is optional.

Usage

```
check4Gaps(x, y = NULL, tol = 0.01, plot = FALSE, silent = FALSE, ...)
```

Arguments

x	A numeric vector to be checked for gaps.
y	An optional vector of y-values which correspond to the x values. Only needed if plot = TRUE.
tol	A number indicating the tolerance for checking if the step between successive x values are the same. Depending upon how the x values are stored and rounded, you may need to change the value of tol to avoid flagging trivial "gaps".
plot	Logical indicating if a plot of the gaps should be made. If TRUE, y must be provided.
silent	Logical indicating a "no gap" condition (return value is FALSE) should not be reported to the console. Important because check4Gaps is called iteratively by other functions.
...	Other parameters to be passed to the plot routines if plot = TRUE, e.g. xlim.

Details

The basic procedure is to compare x[n + 1] - x[n] for successive values of n. When this value jumps, there is a gap which is flagged. beg.indx and end.indx will always be contiguous as indices must be; it is the x values that jump or have the gap. The indices are provided as they are more convenient in some programming contexts. If not assigned, the result appears at the console.

Value

A data frame giving the data chunks found, with one chunk per line. Also a plot if requested. In the event there are no gaps found, FALSE is returned.

beg.freq	The first frequency value in a given data chunk.
end.freq	The last frequency value in a given data chunk.
size	The length (in frequency units) of the data chunk.
beg.indx	The index of the first frequency value in the data chunk.
eng.indx	The index of the last frequency value in the data chunk.

Called by

sumSpectra, binData, sumSpectra

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
x <- seq(from = 5, to = 12, by = 0.1)
remove <- 40:45; x <- x[-remove]
check4Gaps(x) # really simple
gaps <- check4Gaps(x) # save the result for later use
data(CuticleIR) # has a gap, let's find it and display it
check4Gaps(CuticleIR$freq, CuticleIR$data[1,], plot = TRUE)
```

---

chkSpectra *Verify the Integrity of a Spectra Object*

Description

Utility function to verify that the structure of a "Spectra" object (an instance of an S3 object) is internally consistent. Rather than directly manipulating a "Spectra" object, one should manipulate it via removeFreq or removeSample. Should not see much direct use by users.

Usage

```
chkSpectra(spectra, confirm = FALSE)
```

Arguments

spectra	An object of S3 class "Spectra" to be checked.
confirm	Logical indicating whether or not to write the results to the console, as would be desirable for interactive use.

Details

This function is similar in spirit to validObject in the S4 world. When used at the console, and the object is OK, no message is written unless confirm = TRUE. At the console, if there is a problem, messages are issued regardless of the value of confirm. When used in a function, this function is silent (assuming confirm = FALSE) unless there is a problem.

Value

None; messages will be printed at the console if there is a problem.

Called by

All ChemoSpec functions that modify "Spectra" objects.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
chkSpectra(CuticleIR, confirm = TRUE) # OK
# What's next works, but is the wrong way to manipulate a "Spectra" object.
# Use removeSample instead.
remove <- c(20:40)
CuticleIR$freq <- CuticleIR$freq[-remove]
chkSpectra(CuticleIR, confirm = TRUE) # not OK, you didn't listen to me!
```

---

classPCA *Classical PCA of Spectra Objects*

Description

A wrapper which carries out classical PCA analysis on a "Spectra" object. The data are row- and column-centered, and the user can select various options for scaling.

Usage

```
classPCA(spectra, choice = "noscale")
```

Arguments

spectra	An object of S3 class "Spectra".
choice	A character string indicating the choice of scaling. One of c("noscale", "autoscale", "Pareto").

Details

The scale choice `autoscale` scales the columns by their standard deviation. `Pareto` scales by the square root of the standard deviation.

Value

An object of class `prcomp`, modified to include a list element called `$method`, a character string describing the pre-processing carried out and the type of PCA performed (it appears on plots which you might make).

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. `<hanson@depauw.edu>`

References

`http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html`
K. Varmuza and P. Filzmoser *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, 2009.

See Also

`prcomp` for the underlying function, `robPCA` for analogous robust PCA calculations.
For displaying the results, `plotScree`, `plotScores`, `plotLoadings`, `plot2Loadings`.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
plotScores(CuticleIR, title = "Cuticle IR Spectra", results, pcs = c(1,2),
ellipse = "rob", tol = 0.05)
```

`colLeaf` *Color Dendrogram of a Spectra Object*

Description

Utility function called by `hcaSpectra` and `hcaScores` via `dendrapply`. Not intended for end-users.

Usage

```
colLeaf(n, spectra)
```

Arguments

`n`	The leaf number (an integer).
`spectra`	An object of S3 class `"Spectra"`.

Value

None. The leaf colors of the relevant dendrogram object are modified to correspond to those in `spectra`

Called by

`hcaSpectra`, `hcaScores`

Note

The basic idea was found in the help archives. I can't write this kind of stuff!

Author(s)

Bryan A. Hanson, DePauw University. `<hanson@depauw.edu>`

References

`http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html`

`CuticleIR` *IR Spectra of Plant Cuticles by Genotype and Treatment*

Description

IR spectra obtained by ATR sampling on leaves of *Portulaca oleracea* (common purslane). There are 157 spectra, divided into four groups from a G x E experiment. Two genotypes were studied, golden (G), and tall green (T). Two temperature regimes were employed, experimental (E, 35C) and control (C, 22C). Sample name GC10 means golden phenotype, control treatment, plant no. 10.

Usage

```
data(CuticleIR)
```

Format

List of 7
$ freq : num [1:1242] 501 503 505 507 509 ...
$ data : num [1:157, 1:1242] 0.205 0.247 0.219 0.203 0.234 ...
$ names : chr [1:157] "GC10" "GC11" "GC12" "GC14" ...
$ groups: Factor w/ 4 levels "GC","GE","TC",..: 1 1 1 1 1 1 1 1 1 1 ...
$ colors: chr [1:157] "brown2" "brown2" "brown2" "brown2" ...
$ unit : chr [1:2] "Wavenumbers" "Absorbance"
$ desc : chr "Kelly's Complete IR Data Set, Summer 2009"
- attr(*, "class")= chr "Spectra"

Details

Noisy regions at the extremes of the frequency range have been removed. The region from 1800 - 2500 wavenumbers was also removed as it is uninformative.

Source

Data obtained by Kelly Summers at DePauw University, Summer 2009.

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
chkSpectra(CuticleIR)
sumSpectra(CuticleIR) # this also runs chkSpectra() before doing the summary
```

---

gasNIR *Gasoline NIR Data in Spectra Format*

Description

This is the same data as in the `gasoline` data set that is included with `R`. The data has been reformatted as a `"Spectra"` object suitable for use in `ChemoSpec`. In particular, the octane values in the original data set have been recoded as factors into three groups: low, medium, and high. The samples do not segregate well on PCA.

Usage

```
data(gasNIR)
```

Format

List of 7
$ freq : num [1:401] 900 902 904 906 908 910 912 914 916 918 ...
$ data : num [1:60, 1:401] -0.0502 -0.0442 -0.0469 -0.0467 -0.0509 ...
..- attr(*, "dimnames")=List of 2
.. ..$: chr [1:60] "1" "2" "3" "4" ...
.. ..$: chr [1:401] "900 nm" "902 nm" "904 nm" "906 nm" ...
$ names : chr [1:60] "sample.1" "sample.2" "sample.3" "sample.4" ...
$ groups: Factor w/ 3 levels "L","M","H": 1 1 3 1 2 1 3 2 3 3 ...
$ colors: chr [1:60] "forestgreen" "forestgreen" "dodgerblue4" "forestgreen" ...
$ unit : chr [1:2] "nanometers" "intensity"
$ desc : chr "Kalivas' gasoline data set (converted to Spectra object)"
- attr(*, "class")= chr "Spectra"

Source

Full details of the source can be found by typing `?gasoline` at the console.

Examples

```
data(gasNIR)
hcaSpectra(gasNIR, title = "NIR of Gasoline Samples by Octane Level")
```

`getManyCsv` *Merge CSV Files in a Directory into a Spectra Object*

Description

This function will read all .csv files in a directory, and use the file names to construct group membership and assign colors. All the data is placed into an object of S3 class `"Spectra"`.

Usage

```
getManyCsv(gr.crit = NULL, gr.cols = c("auto"),
freq.unit = "no frequency unit provided",
int.unit = "no intensity unit provided",
descrip = "no description provided",
format = "original", out.file = "mydata")
```

Arguments

`gr.crit`	Group Criteria. A vector of character strings which will be searched for among the file names in order to assign an individual spectrum/sample to group membership.
`gr.cols`	Group Colors. Either the word "auto", in which case colors will be automatically assigned, or a vector of acceptable color names with the same length as `gr.crit`. In the latter case, colors will be assigned one for one, so the first element of `gr.crit` is assigned the first element of `gr.col` and so forth. See details below for some other issues to consider.
`freq.unit`	A character string giving the units of the x-axis (frequency or wavelength).
`int.unit`	A character string giving the units of the y-axis (some sort of intensity).
`descrip`	A character string describing the data set that will be stored. This string is used in some plots so it is recommended that it's length be less than about 40 characters.
`format`	A character string giving the format of the .csv files to be processed. Currently set to `"original"` but not used; intended for future flexibility.
`out.file`	A file name acceptable to the `save` function. The completed object of S3 class `"Spectra"` will be written to this file.

Details

The linking of groups with colors is handled by `groupNcolor`. The user may specify any color name known to R. However, if you plan to use `rggobi` and `GGobi` to view the data later, keep in mind that `ggobi` only uses certain color schemes (although there are many options). In the case of `ChemoSpec`, two particular options have been hard-coded into the function `plotScoresG` for simplicity. If you plan to use `plotScoresG`, you should use choose from one of the two color schemes now if you want all your graphics to use the same scheme. Keep in mind that these colors must be used in order (though you can use the order of argument `gr.crit` to associate a particular group with a particular color:

primary scheme: `c("red3", "dodgerblue4", "forestgreen", "purple4", "orangered", "yellow", "orangered4", "violetred2")`

pastel scheme: `c("seagreen", "brown2", "skyblue2", "hotpink3", "chartreuse3", "darkgoldenrod2", "lightsalmon3", "gray48")`

If you want to see what these colors look like, use `display.brewer.pal(8, "Set1)} or \code{display.brewer.pal(8, "Set2)`. Finally, the difficulty to bear in mind here is that R plots are generally on a white background, so pale colors should be avoided, while `GGobi` plots on a black background, so dark colors should be avoided!

Value

An object of S3 class `"Spectra"` and name "spectra" will be written to `out.file`. You can change it's name later by loading it and assigning it to a new name.

Calls

`groupNcolor`

Called by

Top level function, called by user.

Warning

Files whose names are not matched using gr.crit are still incorporated into the `"Spectra"` object, but they are not assigned a group or color and therefore don't plot, though they do take up space in a plot! I will fix this eventually.

Author(s)

Bryan A. Hanson, DePauw University. `<hanson@depauw.edu>`

References

`http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html`

`groupNcolor` *Assign Group Membership and Colors for a Spectra Object*

Description

A utility function which looks for `gr.crit` in the file names of .csv files and assigns group membership. Also assigns a color to each group. Not intended for users.

Usage

```
groupNcolor(spectra, gr.crit = NULL, gr.cols = c("auto"))
```

Arguments

`spectra`	An object of S3 class `"Spectra"`. Until this function acts on `spectra` it is not quite complete.
`gr.crit`	As per `getManyCsv`
`gr.cols`	As per `getManyCsv`

Value

A *complete* object of S3 class "Spectra". Until this function has done its job, an object of class "Spectra" will not pass checks as the assembly is not complete (see chkSpectra).)

Called by

getManyCsv

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

getManyCsv for details.

---

hcaScores *HCA on PCA scores from a Spectra Object*

Description

A wrapper which performs HCA on the scores from a PCA of a "Spectra" object, color-coding the results as specified in the object.

Usage

```
hcaScores(spectra, pca, title = "no title provided",
scores = c(1:5), method = "complete", ...)
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
title	A character string for the plot title.
scores	A vector of integers specifying which scores to use for the HCA.
method	A character string acceptable as a method in hclust.
...	Additional parameters to be passed to the plotting functions.

Value

None. Side effect is a plot.

Calls

colLeaf

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

hclust for the underlying function. See hcaSpectra for HCA of the entire data set stored in the "Spectra" object.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
hcaScores(CuticleIR,  results, scores = c(1:5), title = "Cuticle IR Spectra",
method = "complete")
```

---

hcaSpectra *Plot HCA Results of a Spectra Object*

Description

A wrapper which carries out HCA and plots a dendrogram colored by the information in a "Spectra" object. All methods for computing the cluster distances are available.

Usage

```
hcaSpectra(spectra, title = "no title provided", method = "complete", ...)
```

Arguments

spectra	An object of S3 class "Spectra".
title	A character string for the plot title.
method	A character string acceptable as a method in hclust.
...	Other parameters to be passed to the plotting functions.

Value

None. The side effect is a plot.

Calls

chkSpectra, colLeaf

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

hclust for the underlying function. hcaScores for similar analysis of PCA scores from a "Spectra" object.

Examples

```
data(CuticleIR)
hcaSpectra(CuticleIR, title = "Cuticle IR Spectra", method = "complete")
```

---

isWholeNo *Determine if a Number is a Whole Number*

Description

This function determines if a given number is a whole number within a given tolerance. Taken from the help page of is.integer.

Usage

```
isWholeNo(x, tol = .Machine$double.eps^0.5)
```

Arguments

x	A number to be tested.
tol	Tolerance for the test.

Value

A logical, indicating the outcome of the test.

Called by

binData

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

is.integer

---

labelExtremes *Label Extreme Values in a Data Set*

Description

A utility function which plots the sample names next to the sample points. The number of samples labeled can be specified by passing it from the calling function. Never called by the user.

Usage

```
labelExtremes(data, names, tol)
```

Arguments

data	A matrix containing the x values of the points/samples in the first column, and the y values in the second.
names	A character vector of sample names. Length must match the number of rows in x.
tol	A number describing the fraction of points to be labeled. tol = 1.0 labels all the points; tol = 0.05 labels the most extreme 5 percent.

Value

None. Annotates the plot with labels.

Called by

plotScoresDecoration, pcaDiag, plot2Loadings

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

normSpectra *Normalize a Spectra Object*

Description

This function takes the y-data of a "Spectra" object and normalizes it by dividing each y-value by the sum of the y-values in a given spectrum.

Usage

```
normSpectra(spectra)
```

Arguments

spectra	An object of S3 class "Spectra" to be normalized.

Value

An object of S3 class "Spectra".

Calls

chkSpectra

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
res <- normSpectra(CuticleIR)
sumSpectra(res)
```

pcaBoot *Cross-Validation of Classical PCA Results for a Spectra Object*

Description

This function carries out classical PCA on the data in a "Spectra" object using a cross-validation method. Nothing more than a wrapper to Peter Filzmoser's pcaCV method with some small plotting changes.

Usage

```
pcaBoot(spectra, pcs, choice = "noscale", repl = 50,
segments = 4, segment.type = c("random", "consecutive", "interleaved"),
length.seg, trace = FALSE, ...)
```

Arguments

spectra	An object of S3 class "Spectra".
choice	A character string indicating the choice of scaling. One of c("noscale", "autoscale", "Pareto").
pcs	As per pcaCV where it is called amax; an integer giving the number of PC scores to include.
repl	As per pcaCV; the number of replicates to perform.
segments	As per pcaCV.
segment.type	As per pcaCV.
length.seg	As per pcaCV.
trace	As per pcaCV.
...	Parameters to be passed to the plotting routines.

Value

A list as described in pcaCV, so the result must be assigned or it will appear at the console. Side effect is a plot.

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html
K. Varmuza and P. Filzmoser *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, 2009.

See Also

pcaCV for the underlying function.

Examples

```
data(CuticleIR)
results <- pcaBoot(CuticleIR, pcs = 5, choice = "noscale")
```

---

pcaDiag *Outlier Diagnostic Plots for PCA of a Spectra Object*

Description

A function to carry diagnostics on the PCA results for a "Spectra" object. Basically a wrapper to Filzmoser's pcaDiagplot which colors everything according to the scheme stored in the "Spectra" object. Works with PCA results of either class "prcomp" or class "princomp". Works with either classical or robust PCA results.

Usage

```
pcaDiag(spectra, pca, pcs = 3, quantile = 0.975,
plot = c("OD", "SD"), ...)
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp or prcomp, modified to include a character string ($method) describing the pre-processing carried out and the type of PCA performed.
pcs	As per pcaDiagplot. The number of principal components to include.
quantile	As per pcaDiagplot. The significance criteria to use as a cutoff.
plot	A character string, indicating whether to plot the score distances or orthogonal distances, or both. Options are c("OD", "SD").
...	Additional parameters to be passed to the plotting functions.

Details

If both plots are desired, the output should be directed to a file rather than the screen. Otherwise, the 2nd plot overwrites the 1st in the active graphics window. Alternatively, just call the function twice, once specifying OD and once specifying SD.

Value

A list is returned as described in pcaDiagplot, so the result must be assigned or it will appear at the console. Side effect is a plot.

Calls

q2rPCA, labelExtremes

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html
K. Varmuza and P. Filzmoser *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, 2009.

See Also

pcaDiagplot in package chemometrics for the underlying function.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
temp <- pcaDiag(CuticleIR, results, pcs = 2, plot = "OD")
```

---

plot2Loadings *Plot PCA Loadings from a Spectra object Against Each Other*

Description

Plots two PCA loadings specified by the user, and labels selected (extreme) points. Typically used to determine which variables (frequencies) are co-varying, although in spectroscopy most peaks are represented by several variables and hence there is a lot of co-varying going on. Also useful to determine which variables are contributing the most to the clustering on a score plot.

Usage

```
plot2Loadings(spectra, pca, title = "no title provided",
loads = c(1, 2), tol = 0.05, ...)
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
title	A character string for the plot title.
loads	A vector of two integers specifying which loading vectors to plot.
tol	A number describing the fraction of points to be labeled. tol = 1.0 labels all the points; tol = 0.05 labels the most extreme 5 percent.
...	Other parameters to be passed to the plotting routines.

Value

None. Side effect is a plot.

Calls

labelExtremes

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

See plotLoadings to plot one loading against the original variable (frequency) axis.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
plot2Loadings(CuticleIR, title = "Cuticle IR Spectra", results,
loads = c(1,2), tol = 0.05)
```

---

plotLoadings *Plot PCA Loadings for a Spectra Object*

Description

Creates a multi-panel plot of loadings along with a reference spectrum.

Usage

```
plotLoadings(spectra, pca, title = "no title provided",
loads = c(1), ref = 1, ...)
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
title	A character string for the plot title.
loads	An integer vector giving the loadings to plot. More than 3 loadings creates a useless plot using the default graphics window.

ref	An integer specifying the reference spectrum to plot, which appears at the bottom of the plot.
...	Additional parameters to be passed to plotting functions.

Value

None. Side effect is a plot.

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

See plot2Loadings to plot two loadings against each other.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
plotLoadings(CuticleIR, results, title = "Cuticle IR Spectra",
loads = c(1), ref = 1)
```

---

plotScores *Plot PCA Scores of a Spectra Object*

Description

Plots the requested PCA scores using the color scheme derived from a "Spectra" object. Options are provided to add confidence ellipses for each group in the object. The ellipses may be robust or classical. Option to label the extreme points provided.

Usage

```
plotScores(spectra, pca, title = "no title provided",
pcs = c(1, 2), ellipse = "none", tol = "none", ...)
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
title	A character string for the plot title.
pcs	A vector of two integers specifying the PCA scores to plot.
ellipse	A character vector specifying the type of ellipses to be plotted. One of c("both", "none", "cls", "rob"). cls specifies classical confidence ellipses, rob specifies robust confidence ellipses.
tol	A number describing the fraction of points to be labeled. tol = 1.0 labels all the points; tol = 0.05 labels the most extreme 5 percent.
...	Additional parameters to be passed to the plotting functions.

Value

None. Side effect is a plot.

Calls

plotScoresDecoration, plotScoresCor, chkSpectra

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

For other ways of displaying the results, plotScree, plotLoadings, plot2Loadings. For a 3D plot of the scores, see plotScores3D.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
plotScores(CuticleIR, title = "Cuticle IR Spectra", results,
pcs = c(1,2), ellipse = "both", tol = 0.05)
```

plotScores3D *3D PCA Score Plot for a Spectra Object*

Description

Creates a basic 3D plot of PCA scores from the analysis of a "Spectra" object, color coded according the to scheme stored in the object.

Usage

```
plotScores3D(spectra, pca, pcs = c(1:3), title = "no title provided",
view = list(y = 34, x = 10, z = 0))
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
title	A character string for the plot title.
pcs	A vector of three integers specifying the PCA scores to plot.
view	A list of viewing transformations to be applied to the data. May contain values for x, y and z axes; keep in mind that the order of the transformations is important. For example, specifying view = list(x = 45, y = 10) produces a different view than view = list(y = 10, x = 45). The list may be as along as you like - the series of transformations representing an accumulation of tweaks to achieve the desired view.

Value

None. Side effect is a plot.

Calls

chkSpectra

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

For a 2D plot of the scores, see plotScores. For more sophisticated 3D plots, use plotScoresG, or plotScoresRGL for simple interactive plot.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
plotScores3D(CuticleIR, results, title = "Cuticle IR Spectra")
```

---

plotScoresCor *Compute Confidence Ellipses*

Description

A utility function which when given a x,y data set computes both classical and robust confidence ellipses. Never called by the user.

Usage

```
plotScoresCor(x, quan = 1/2, alpha = 0.025)
```

Arguments

x	As per cor.plot.
quan	As per cor.plot.
alpha	As per cor.plot.

Value

A list with the following elements (a simpler version of that in the original function cor.plot):

x.cls	The x values for the classical ellipse.
y.cls	The y values for the classical ellipse.
c	The correlation value for the classical ellipse.
x.rob	The x values for the robust ellipse.
y.rob	The y values for the robust ellipse.
r	The correlation value for the robust ellipse.

Called by

plotScores

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

See function cor.plot in package mvoutlier on which this function is based.

---

plotScoresDecoration

Decorate PCA Score Plot of a Spectra Object

Description

Utility function to carry out misc. labeling functions on the PCA score plot of a "Spectra" object. Never called by the user.

Usage

```
plotScoresDecoration(spectra, pca, pcs = c(1, 2), tol = "none")
```

Arguments

spectra	An object of S3 class "Spectra"
pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
pcs	A vector of two integers specifying the PCA scores to plot.
tol	A number describing the fraction of points to be labeled. tol = 1.0 labels all the points; tol = 0.05 labels the most extreme 5 percent.

Value

None. The score plot is decorated.

Calls

labelExtremes

Called by

plotScores

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

plotScoresG *Plot PCA Scores of a Spectra Object using GGobi*

Description

Plots the specified PCs from a PCA analysis of a "Spectra" object using GGobi.

Usage

```
plotScoresG(spectra, pca, pcs = c(1:3), scheme = "primary")
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp.
pcs	A vector of integers specifying the scores to plot. GGobi handles multivariate data, and so it is not necessary to limit the analysis to 3 dimensions.
scheme	The color scheme to use, one of c("primary", "pastel"). See details for more on color choices.

Details

In general, users many specify any valid R color for use in ChemoSpec. However, the color schemes available to GGobi are limited. If you want all your graphics to use the same color scheme, you need to specify a color scheme useable with GGobi at the time the "Spectra" object is created using the function getManyCsv. Two particular color schemes have been hard-coded into the function described here; the choices are:

primary: c("red3", "dodgerblue4", "forestgreen", "purple4", "orangered", "yellow", "orangered4", "violetred2")

pastel: c("seagreen", "brown2", "skyblue2", "hotpink3", "chartreuse3", "darkgoldenrod2", "lightsalmon3", "gray48")

These are very close matches to the color schemes used in GGobi, which in turn come from RColorBrewer. If you want to see what these colors look like, use display.brewer.pal(8, "Set1)} or \code{display.brewer.pal(8, "Set2).

The behavior of this function is to first check to see if a GGobi-compatible color scheme already exists in spectra. If it does, it is re-used with the same mapping of colors to groups as originally specified and stored in spectra. If the color scheme is not GGobi-compatible, color assignments for GGobi window are generated and the mapping of old colors to new colors is reported at the console.

Finally, a difficulty to bear in mind is that R plots are generally on a white background, so pale colors should be avoided, while GGobi plots on a black background (interactively), so dark colors should be avoided! Plus, ggplot2 plots on a grey background by default! So it is not necessarily possible to specify a single color scheme that works with all possible plotting systems in R.

Value

None. Side effect is a plot in an X11 window created by GGobi. The mapping of colors from the Spectra object to the GGobi color schemes is printed at the console, since the original color choices may not be acceptable to GGobi.

Calls

chkSpectra; Requires that GGobi be installed and X11 running.

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html
http://www.ggobi.org/
D. Cook & D. F. Swayne, *Interactive and Dynamic Graphics for Data Analysis with R and GGobi*, Springer 2007.

See Also

For plotting only 3 scores with rudimentary control over the view, see plotScores3D. For plotting 2 scores, see plotScores.

Examples

```
data(CuticleIR)
require(ggobi)
results <- classPCA(CuticleIR, choice = "noscale")
## Not run: plotScoresG(CuticleIR, results)
# GGobi runs interactively using X11
```

---

plotScoresRGL *Interactive 3D Score Plot of a Spectra Object*

Description

This function uses the rgl package to create an interactive plot of PCA scores derived from a "Spectra" object. A title and legend can be added if desired.

Usage

```
plotScoresRGL(spectra, pca, pcs = c(1:3), title = NULL,
t.pos = NULL, leg.pos = NULL, lab.opts = FALSE, ...)
```

Arguments

spectra	An object of S3 class "Spectra".
pca	An object of class prcomp.
pcs	A vector of three integers specifying the PCA scores to plot.
title	A character string for the plot title.

t.pos	A character selection from LETTERS[1:8] (= A through H) indicating the desired location for the title.
leg.pos	A character selection from LETTERS[1:8] (= A through H) indicating the desired location for the legend.
lab.opts	A logical indicating whether or not to display the locations where the title and legend can be placed. These locations are the corners of a cube surrounding the data.
...	Additional parameters to pass downstream, generally to the plotting routines.

Details

If you intend to make a hard copy of your plot, use lab.opts = TRUE until you have found a good view of your data. Then note corners of the cube where the title and legend won't interfere with viewing the data, and use these as arguments for t.pos and leg.pos, and add title. Adjust as necessary, then turn off label display using lab.opts = FALSE. Back at the console, use > rgl.snapshot("file_name.png") to create the hardcopy.

Value

None. Side effect is a plot

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

Other functions in ChemoSpec that plot PCA scores are: plotScores (2D version), plotScores3D (uses lattice graphics) and plotScoresG which uses ggobi and rggobi.

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "autoscale")
plotScoresRGL(CuticleIR, results, title = "Cuticle IR Spectra", leg.pos = "A", t.pos = "E
```

---

plotScree *Scree Plot of PCA Results for a Spectra Object*

Description

Function to draw a scree plot illustrating the importance of the components in a PCA analysis of a "Spectra" object.

Usage

```
plotScree(pca, title = "no title provided", ...)
```

Arguments

pca	An object of class prcomp, modified to include a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on the plot). This is automatically provided if ChemoSpec functions classPCA or robPCA were used to create pca.
title	A character string for the plot title.
...	Additional parameters to be passed to plotting functions.

Details

Technically, if you add $method to the PCA results from other packages, this will plot a scree plot for any PCA results, not just those from "Spectra" objects.

Value

None. Side effect is a plot.

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
results <- classPCA(CuticleIR, choice = "noscale")
plotScree(results, title = "Cuticle IR Spectra")
```

---

plotSpectra *Plot Spectra Object*

Description

Plots the spectra stored in a "Spectra" object. One may choose which spectra to plot, and the x range to plot. Spectra may be plotted offset or stacked. The vertical scale is controlled by a combination of several parameters.

Usage

```
plotSpectra(spectra, title = "no title provided",
which = c(1), yrange = c(0, max(spectra$data)), offset = 0,
amplify = 1, lab.pos = mean(spectra$freq), ...)
```

Arguments

spectra	An object of S3 class "Spectra".
title	A character string for the plot title.
which	An integer vector specifying which spectra to plot, and the order.
yrange	A vector giving the limits of the y axis desired, for instance c(0, 15). This parameter depends upon the range of values in the stored spectra and defaults to the height of the largest peak in the data set. Interacts with the next two arguments, as well as the number of spectra to be plotted as given in which. Trial and error is used to adjust all these arguments to produce the desired plot.
offset	A number specifying the vertical offset between spectra if more than one is plotted. Set to 0.0 for a stacked plot.
amplify	A number specifying an amplification factor to be applied to all spectra. Useful for magnifying spectra so small features show up (though large peaks will then be clipped, unless you zoom on the x axis).
lab.pos	A number giving the location for the identifying label. Generally, pick an area that is clear in all spectra plotted. If no label is desired, give lab.pos outside the plotted x range.
...	Additional parameters to be passed to plotting functions.

Value

None. Side effect is a plot.

Calls

chkSpectra

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
plotSpectra(CuticleIR, "Cuticle IR Spectra",
which = c(10:11, 40:41, 100:101, 140:141, 150:151),
yrange = c(0,10),
offset = 0.8, amplify = 1.0, lab.pos = 2000)
```

q2rPCA *Conversion Between PCA Classes*

Description

Utility to convert objects of S3 class "prcomp" (Q-mode PCA) to objects of S3 class "princomp" (R-mode PCA) or *vice-versa*. Not likely to be called by most users.

Usage

```
q2rPCA(x)
r2qPCA(x)
```

Arguments

x	An object of either class "prcomp" or class "princomp". It will be converted to a form that can be used by functions expecting either class.

Details

In the conversion, the necessary list elements are added; the "old" elements are not deleted (and user added list elements are not affected). To indicate this, the class attribute is updated to include class "conPCA". The new object can then be used by functions expecting either class prcomp or princomp. For details of the structure of prcomp or princomp, see their respective help pages.

Value

A list of class "conPCA". Note that the order of the elements will vary depending upon the direction of conversion.

loadings	The loadings from "princomp", or a copy of the rotations from "prcomp".
scores	The scores from "princomp", or a copy of the x values from "prcomp".
call	The call. Objects of class "prcomp" do not store the original call, so a place holder is used. Otherwise the unchanged call from "princomp".
n.obs	The number of observations from "princomp", or computed from the 1st dimension of x in "prcomp".
class	"conPCA" is pre-pended to the existing class.
sdev	Unchanged from original.
center	Unchanged from original.
scale	Unchanged from original.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

prcomp, princomp

removeFreq *Remove Frequencies from a Spectra Object*

Description

This function removes specified frequencies from a "Spectra" object. For instance, one might want to remove regions lacking any useful information (to reduce the data size), or remove regions with large interfering peaks (e.g. the water peak in 1H NMR).

Usage

```
removeFreq(spectra, rem.freq)
```

Arguments

spectra	An object of S3 class "Spectra" from which to remove selected frequencies.
rem.freq	A list of frequencies to be removed. Some rounding may occur.

Details

To remove the extreme values, some experimentation with the max/min value in rem.freq may be necessary due to rounding. You can check your success by using check4Gaps and looking for an orphaned value or two.

Value

An object of S3 class "Spectra".

Calls

chkSpectra

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
sumSpectra(CuticleIR) # note the frequency range & gaps
res <- removeFreq(CuticleIR, rem.freq = c(2850:3050)) # remove hydrocarbon peaks
sumSpectra(res)
```

`removeSample` *Remove Samples from a Spectra Object*

Description

Removes specified samples from a `"Spectra"` object.

Usage

```
removeSample(spectra, rem.sam)
```

Arguments

`spectra`	An object of S3 class `"Spectra"`
`rem.sam`	Either an integer vector specifying the samples to be removed, or a character vector giving the sample names to be removed.

Details

If `rem.sam` is a character vector, the sample names are grepped for the corresponding values. Remember that the grepping process is greedy, i.e. grepping for "XY" find not only "XY" but also "XYZ".

Value

A modified object of S3 class `"Spectra"`.

Calls

`chkSpectra`

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. `<hanson@depauw.edu>`

References

`http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html`

See Also

`removeFreq` to remove selected frequencies from a `"Spectra"` object.

Examples

```
data(CuticleIR)
new1 <- removeSample(CuticleIR, rem.sam = 20)
# removes the 20th spectrum/sample
new2 <- removeSample(CuticleIR, rem.sam = "GE")
# removes all samples whose name contains "GE"
new3 <- removeSample(CuticleIR, rem.sam = "GE10")
# removes one spectrum/sample with this exact name.
```

---

robPCA *Robust PCA of a Spectra Object*

Description

A wrapper which carries out robust PCA analysis on a "Spectra" object. The data are row- and column-centered, and the user can select various options for scaling.

Usage

```
robPCA(spectra, choice = "noscale")
```

Arguments

spectra	An object of S3 class "Spectra"
choice	A character vector describing the type of scaling to be carried out. One of c("noscale", "mad").

Value

An object of classes "conPCA" and "princomp" (see q2rPCA). It includes a list element called $method, a character string describing the pre-processing carried out and the type of PCA performed (it appears on plots which you might make).

Calls

chkSpectra, r2qPCA

Called by

Top level function, called by user.

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html
K. Varmuza and P. Filzmoser *Introduction to Multivariate Statistical Analysis in Chemometrics*, CRC Press, 2009.

See Also

See PCAgrid on which this function is based. For the classical version, see classPCA.
For displaying the results, plotScree, plotScores, plotLoadings, plotScoresG, plotScores3D

Examples

```
data(CuticleIR)
results <- robPCA(CuticleIR, choice = "mad")
plotScores(CuticleIR, title = "Cuticle IR Spectra",
results, pcs = c(1,2), ellipse = "rob", tol = 0.05)
```

---

specSurvey *Plot Std Dev of Frequencies in a Spectra Object*

Description

Computes the standard deviation of each frequency value for all the spectra in a "Spectra" object and plots it against the frequency. Useful for identifying uninformative spectral regions.

Usage

```
specSurvey(spectra, title = "No title provided", ...)
```

Arguments

spectra	An object of S3 class "Spectra".
title	A character string for the plot title.
...	Additional parameters to be passed to the plotting functions.

Value

None. Side effect is a plot.

Calls

chkSpectra

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
specSurvey(CuticleIR, title = "Cuticle IR Spectra")
```

Spectra *Spectra Objects*

Description

In ChemoSpec, spectral data sets are stored in an S3 class called Spectra, which contains a variety of information in addition to the spectra themselves. Spectra objects are created by getManyCsv or similar functions (no others currently exist).

Structure

The structure of a Spectra object is a list of 7 elements and an attribute as follows:

element	*type*	*description*
$freq	num	A common frequency (or wavelength) axis for all the spectra.
$data	num	The intensities for the spectra. A matrix of dimension no. samples x no. frequency points.
$names	chr	The sample names for the spectra; length must be no. samples.
$groups	Factor	The group classification of the samples; length must be no. samples.
$colors	chr	The colors for each sample; length must be no. samples. Groups and colors correspond.
$unit	chr	Two entries, the first giving the x axis unit, the second the y axis unit.
$desc	chr	A character string describing the data set. This appears on plots and therefore should probably be kept to 40 characters or less.
- attr	chr "Spectra"	The S3 class designation.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

See Also

sumSpectra to summarize a "Spectra" object. chkSpectra to verify the integrity of a "Spectra" object.

sumSpectra *Summarize a Spectra Object*

Description

Provides a summary of a "Spectra" object, essentially a more spectroscopist-friendly version of str().

Usage

```
sumSpectra(spectra, ...)
```

Arguments

spectra	An object of S3 class "Spectra".
...	Arguments to be passed downstream. See details.

Details

Prior to summarizing, chkSpectra is run with confirm = FALSE. If there are problems, warnings are issued to the console and the summary is not done. If issued with plot = TRUE, this is passed via the ... to check4Gaps which then produces a plot showing any gaps in the data. Cool.

Value

None. Results printed at console, perhaps a plot as well.

Calls

chkSpectra, check4Gaps

Called by

Top level function, called by user.

Author(s)

Bryan A. Hanson, DePauw University. <hanson@depauw.edu>

References

http://academic.depauw.edu/~hanson/ChemoSpec/ChemoSpec.html

Examples

```
data(CuticleIR)
sumSpectra(CuticleIR)
```

Index