

Package ‘Metabonomic’

October 13, 2009

Type Package

Title GUI for Metabonomic Analysis

Version 3.3.1

Date 2009-10-13

Author Jose L. Izquierdo <izquierdo@ieb.ucm.es> Instituto de Estudios Biofuncionales(U.C.M.).

Maintainer Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Description Graphical User Interface for the Metabonomic Analysis (Baseline, Normalization, Peak Detection, PCA, PLS, Nearest Neighbourgt, Neural Network) developed to make easy this data analysis.

License GPL (>= 2)

Depends tcltk, tkplot, robustbase, MASS, gpls, pls, splines, class, nnet, AMORE, clusterSim, hddplot, scatterplot3d, relimp, Icents, FTICRMS, waved, Matrix, caTools

Suggests kopls, PROcess

Repository CRAN

Date/Publication 2009-10-13 15:40:46

R topics documented:

Metabonomic-package	2
Baseline.Correction	3
Import.data	4
Manual.cut	5
manual.model	6
Met.Align	6
Met.B.STAT	7
Met.Binning	7
Met.Checkbox1	8
Met.delete	8

Met.KNN	9
Met.kopls	10
Met.LDA	11
Met.Load.Data	12
Met.Metabolites	13
Met.modalDialog	13
Met.modalDialog2	14
Met.modalDialog3	14
Met.modalDialog4	14
Met.model.1	15
Met.NN1	15
Met.NN2	16
Met.Norm	17
Met.Order	18
Met.PCA	18
Met.peak.detection	19
Met.PLS1	20
Met.PLS2	21
Met.RadioBox1	22
Met.RadioBox2	22
Met.redo	22
Met.Save.Data	23
Met.Selection	23
Met.Show	24
Met.spectrum	24
Met.spectrum.2	25
Met.undo	25
Metabonomic	26
msc.biomarkers.fill	27
msc.peaks.align	28
msc.peaks.alignment	29
msc.peaks.clust	31
msc.peaks.find	32
Phase.correction	34
Require	34
showData2	35
TABLE	36
Try	36
Index	37

Metabonomic-package

Metabonomic: GUI for the Metabonomic Analysis

Description

Graphical User Interface for the Metabonomic Analysis (Baseline, Normalization, Peak Detection, PCA, PLS, Nearest Neighbour, Neural Network) developed to make easy this data analysis.

Details

Package: Metabonomic
Type: Package
Version: 1.0-Beta
Date: 2008-06-13
License: What license is it under?

Metabonomic()

Author(s)

Jose Luis Izquierdo (Instituto de Estudios Biofuncionales UCM)

Maintainer: Jose L. Izquierdo (izquierdo@ieb.ucm.es)

Baseline.Correction

Baseline Correction (FTICRMS)

Description

Baseline Correction of raw spectra based in the FTICRMS package.

Usage

Baseline.Correction()

Details

Baseline correction may be a very essential step to obtain high quality NMR spectra in some cases. Rolling baselines can make it difficult to identify peaks, as well as introduce significant errors into any quantitative measurements. This application based in the FTICRMS package is available for the individual baseline correction. It computes an estimated baseline curve for a spectrum by a method of Rocke and Xi . Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

FTICRMS package <http://cran.r-project.org/web/packages/FTICRMS/index.html>

Import.data

Bruker spectra

Description

Import.data loads the Bruker Spectra for the metabonomic analysis.

Usage

```
Import.data()
```

Details

Import.data has been developed to load directly the spectra in the Bruker spectroscopy format. Import.data can also be executed by selecting the 'file/ Import Bruker file' tab of the Metabonomic GUI. The user has to select the raw data (FID file in the Bruker data directory). This application displays the spectrum reference and manages basic operations such as setting the chemical shift of a certain compound (TSP or DSS) to 0 ppm and zero order and first order phase corrections. When the first set of data is loaded, the GUI asks for a new array. After finishing importing all the spectra, the GUI asks for the "info" text file with the information of the different spectra, where the first column holds the names of the spectra and the different characteristics are in the followings columns. Applications to load other commercial data formats will be added soon.

Example of an info file:

Name	Category	Disease	Sex
43FC	Control	Healthy	F
45FC	Control	Healthy	F
04MC	Control	Healthy	M
54MC	Control	Healthy	M
55FC	Control	Healthy	F
55MC	Control	Healthy	M
09FF	Treated	Smoker	F
09MF	Treated	Smoker	M
53Ff	Treated	Smoker	F
53MF	Treated	Smoker	M
10MA	Treated	Asthmatic	M
11FA	Treated	Asthmatic	F
11MA	Treated	Asthmatic	F
42FA	Treated	Asthmatic	F

42MA	Treated	Asthmatic	M
46MA	Treated	Asthmatic	M
19MA	Treated	Asthmatic	M
46FA	Treated	Asthmatic	F
49MA	Treated	Asthmatic	M
50MA	Treated	Asthmatic	M
12FE	Treated	Emphysema	F
12ME	Treated	Emphysema	M
37ME	Treated	Emphysema	M
38FE	Treated	Emphysema	F
41ME	Treated	Emphysema	M
.	.	.	.
.	.	.	.
.	.	.	.

Value

datos	Spectra data frame
memory.Data	Internal value
memory	Internal value

Author(s)

Jose L. Izquierdo (izquierdo@ieb.ucm.es)

Manual.cut

Spectral region selection

Description

This function selects the spectral region to the statistical analysis. Also, Manual.cut can be used to remove the water peak.

Usage

```
Manual.cut ()
```

Details

Launched with the GUI. Beta version.

Value

datos	Spectra data frame
memory.Data	Internal value
memory	Internal value

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

manual.model

Manual Model

Description

manuel.model select the samples to build the diferent statistical models.

Usage

```
manual.model(title)
```

Arguments

title Graphical interface title.

Details

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.Align

Peaks Alignment

Description

The same peaks detected in diferent spectra will be aligned in the same chemical shift position.

Usage

```
Met.Align(datos, Peaks)
```

Arguments

datos Spectra data frame
Peaks Detected Peaks by [Met.peak.detection](#)

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

caMassClass package <http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

Met.B.STAT

Chemical Shift Region Display

Description

This function shows the differences among the different subgroups in a determinate spectral region. The application plots the values of all samples in a determinate chemical shift region and the mean value of these

Usage

```
Met.B.STAT(datos)
```

Arguments

datos Spectra data frame

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.Binning

Binning

Description

The function 'binning' takes a matrix of spectra and constructs a matrix of intensities of reduced dimensions based on an equally spaced mesh of interval breaks.

Usage

```
Met.Binning(datos)
```

Arguments

datos Spectra data frame

Details

The most common method of reducing the influence of shifting peaks is the so-called binning or bucketing method, which reduces the spectrum resolution. Thereby the spectra are integrated within small spectral regions, called "bins" or "buckets". Subsequent data analysis procedures, which are applied to the binned spectra, are not influenced by peak shifts, as long as these shifts remain within the borders of the corresponding bins. After launching the Binning graphical applications (Preprocessing / Binning), the user can select the bin size. The spectra integration is performed by running the "binning" function from the PROcess library. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

caMassClass package <http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

Met.Checkbox1 *Checkbox1*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.delete *Spectra deletion*

Description

This function eliminates unwanted spectra.

Usage

Met.delete()

Details

Launched with the GUI. Beta version.

Value

<code>datos</code>	Spectra data frame
<code>memory.Data</code>	Internal value
<code>memory</code>	Internal value

Author(s)

Jose L. Izquierdo (izquierdo@ieb.ucm.es)

Met.KNN

k-Nearest Neighbour Classification

Description

k-nearest neighbour classification for test set from training set. For each row of the test set, the k nearest (in Euclidean distance) training set vectors are found, and the classification is decided by majority vote, with ties broken at random. If there are ties for the kth nearest vector, all candidates are included in the vote.

Usage

`Met.KNN(datos, externa)`

Arguments

<code>datos</code>	Spectra data frame
<code>externa</code>	Not implemented yet

Details

The k-Nearest Neighbors (KNN) rule for classification is the simplest of all supervised classification approaches. For classification of an unknown object, its distance, usually the Euclidian distance, to all other objects is computed. The minimum distance is selected and the object is assigned to the corresponding class. The KNN graphical interface (Metabonomic Analysis / KNN) allows to choose between random or manual selection of the samples to build the model, number of the neighbors, minimum vote for definite decision or the use or not of all the neighbors. If the use of all the neighbors is selected, all distances equal to the kth largest are included. If not, a random selection of distances equal to the kth is chosen to use exactly k neighbors. To finish, it returns the results of the validation test and the cross validation test. The KNN graphical application makes use of the "knn" function from the class package

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

class package <http://finzi.psych.upenn.edu/R/library/class/html/knn.html>

Met.kopls

Kernel-based orthogonal projections to latent structures (K-OPLS)

Description

An implementation of K-OPLS based on 'kopls' package with e.g. model training, prediction, cross-validation and plot tools.

Usage

```
Met.kopls(datos)
```

Arguments

datos Spectra data frame

Details

The application (Metabonomic Analysis / Partial Least Squares / Kernel-based Orthogonal PLS) is performed using the functions from the "kopls" package. Firstly, the function perform a K-OPLS cross-validation for a set of 'Y'-orthogonal components. The function returns a number of diagnostic parameters which can be used to determine the optimal number of model components. With the optimal parameters selected , the application performs prediction of no-model samples from the K-OPLS model. The function projects the Y-predictive and Y-orthogonal scores components to predict a value of the response matrix Y.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

Rantalainen M, Bylesjo M, Cloarec O, Nicholson JK, Holmes E and Trygg J. *Kernel-based orthogonal projections to latent structures (K-OPLS)*, *J Chemometrics* 2007; 21:376-385.

Description

Linear Discriminant Analysis.

Usage

```
Met.LDA(datos, externa)
```

Arguments

datos	Spectra data frame
externa	Not implemented yet

Details

Linear Discriminant Analysis (LDA) is another common technique for the analysis of metabonomic data. LDA is used to obtain linear discriminant functions, a linear combination of the original variables chosen to maximize the differences of the classes. The linear discriminant function is calculated by the "lda" function from the "MASS" package. The user is guided by the program to perform all the tasks in the proper order. Firstly, a model of LDA is built with part of the samples and the rest of the samples will be utilized to do a validation test. The user can directly choose the samples to make the model or select the number of samples of each class to be chosen in a random selection. Secondly, the user can select the algorithm to calculate the LDA among "moment" for standard estimators of the mean and variance, "mle" for a Maximum likelihood Estimation or "t" for robust estimates based on a t distribution. Finally, the LDA graphical application returns the results of the validation test and different interactive graphics of the LDA model. If the number of different classes is less than three, the interactive graphic will be a plane where the samples used to build the model and the validation samples will be plotted. If the number of different classes is more than two, the samples used to build the model and the validations samples will be plotted in interactive cubes. In these interactive plots, the user can select the angle of rotation, the components showed and other graphical parameters.

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

MASS package <http://finzi.psych.upenn.edu/R/library/MASS/html/lda.html>

Met.Load.Data

*Data spectra importation***Description**

Met.Load.Data loads data spectra txt file for the metabonomic analysis.

Usage

```
Met.Load.Data ( )
```

Details

The NMR spectra for the metabonomic analysis are loaded as a text file by selecting the "file/ Load Data file" tab. The text file, with no header, has in the first column the chemical shift (in ppm) and the intensities of the different spectra are in the following columns. After finishing importing the spectra text file, the GUI asks for an "info" file. This file contains all the information about the different samples, previously written by the user as a text file, where the first column holds the names of the samples and the different characteristics are in the followings columns separated by tabs. A header with the caption of each column is also required.

Example of info file:

Name	Category	Disease	Sex
43FC	Control	Healthy	F
45FC	Control	Healthy	F
04MC	Control	Healthy	M
54MC	Control	Healthy	M
55FC	Control	Healthy	F
55MC	Control	Healthy	M
09FF	Treated	Smoker	F
09MF	Treated	Smoker	M
53Ff	Treated	Smoker	F
53MF	Treated	Smoker	M
10MA	Treated	Asthmatic	M
11FA	Treated	Asthmatic	F
11MA	Treated	Asthmatic	F
42FA	Treated	Asthmatic	F
42MA	Treated	Asthmatic	M
46MA	Treated	Asthmatic	M
19MA	Treated	Asthmatic	M
46FA	Treated	Asthmatic	F
49MA	Treated	Asthmatic	M
50MA	Treated	Asthmatic	M
12FE	Treated	Emphysema	F
12ME	Treated	Emphysema	M
37ME	Treated	Emphysema	M
38FE	Treated	Emphysema	F

41ME	Treated	Emphysema	M
.	.	.	.
.	.	.	.
.	.	.	.

Value

datos	Spectra data frame
memory.Data	Internal value
memory	Internal value

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.Metabolites *Metabolites Selection*

Description

Metabolites Selection

Usage

Met.Metabolites(datos, externa)

Arguments

datos	Spectra data frame
externa	Not implemented yet

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.modalDialog *Met.modalDialog*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

`Met.modalDialog2` *Met.modalDialog2*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

`Met.modalDialog3` *Met.modalDialog3*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

`Met.modalDialog4` *Met.modalDialog4*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

`Met.model.1`*Met.model.1*

Description

Internal function.

Usage

```
Met.model.1(title)
```

Arguments

`title` Graphical display title.

Value

`invisible` NULL

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

`Met.NN1`*Neural Networks*

Description

Fit single-hidden-layer neural network, possibly with skip-layer connections.

Usage

```
Met.NN1(datos, externa)
```

Arguments

`datos` Spectra data frame

`externa` Not implemented yet

Details

Application of artificial neural networks (ANN) for data processing is characterized by analogy with a biological neuron. An ANN consists of a layered network of nodes, each of which performs a simple operation on several inputs to produce a single output. This application (Metabonomic Analysis / Neural Network / Neural Network (Single hidden layer)) makes use of the "nnet" function from the "nnet" R package. This graphical application allows the user to build a single-hidden-layer neural network, selecting the number of units in the hidden layer, the initial random weight and the weight decay. In addition, the user can choose between random or manual selection of the training samples. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo (izquierdo@ieb.ucm.es)

References

nnet package <http://finzi.psych.upenn.edu/R/library/nnet/html/nnet.html>

Met.NN2

Neural Network 2

Description

Creates a feedforward artificial neural network according to the structure established by the AMORE package standard.

Usage

```
Met.NN2(datos, externa)
```

Arguments

datos	Spectra data frame
externa	Not implemented yet

Details

This application creates a feedforward artificial neural network according to the structure established by the "AMORE" package. With this application the user can select the number of layers and the number of neurons in each layer. Moreover, a lot of parameters can be controlled by the user like the learning rate at which every neuron is trained, the momentum for every neuron, the error criterion (Least Mean Square or Least Mean Logarithm Squares), the activation function of the hidden and the output layer (Purelin, Tansig, Sigmoide or Hardlim) and the training method (Adaptative gradient descend or BATCH gradient descend, with or without momentum). With these parameters selected, the algorithm trains the network with the samples manually or randomly selected, and it tests the networks with the other samples. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

AMORE package <http://finzi.psych.upenn.edu/R/library/AMORE/html/newff.html>

Met.Norm

Normalization

Description

Types of variable normalization

Usage

```
Met.Norm(datos)
```

Arguments

datos Spectra data frame

Details

A crucial step in data pre-processing of spectra from metabonomic studies is the so-called normalization. The normalization step tries to account for possible variations in sample concentrations. Normalization can also be necessary due to technical reasons. If spectra are recorded using a different number of scans or if spectra are recorded with different devices, the absolute values of the spectra are different and rendering a joint analysis of spectra without prior normalization is impossible. The normalization graphical application (Preprocessing / Normalization) allows choosing between several types of normalizations using functions from the clusterSim library. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

clusterSim package <http://finzi.psych.upenn.edu/R/library/clusterSim/html/data.Normalization.html>

`Met.Order`*Order Metabolites based on their ability to discriminate*

Description

The F statistic is used to order the metabolites.

Usage

```
Met.Order(datos)
```

Arguments

`datos` Spectra data frame

Details

Launched with the GUI. Beta version.

Value

`ordtotal.a` List of the metabolites ordered

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

hddplot package <http://finzi.psych.upenn.edu/R/library/hddplot/html/orderFeatures.html>

`Met.PCA`*Principal Components Analysis*

Description

Principal Components Analysis

Usage

```
Met.PCA(datos)
```

Arguments

`datos` Spectra data frame

Details

Principal components analysis (PCA) is one of the most common exploratory steps for multivariate data analysis. The most important use of PCA is indeed to represent a multivariate data in a low-dimensional space. The first principal component represents the direction of largest variation in the swarm of points. The second principal component corresponds with the second largest variation, and so on. The "Metabonomic" GUI incorporates a PCA graphical application (Metabonomic Analysis / PCA) to guide the users in the performance of PCA, allowing the selection of the algorithm parameters. In addition, interactive graphics have been developed to change the component showed, graphical parameters, etc. The Principal Components algorithm used is based on the "prcomp" function from the stats library. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo (izquierdo@ieb.ucm.es)

References

stats package <http://finzi.psych.upenn.edu/R/library/stats/html/prcomp.html>

Met.peak.detection *Find Peaks*

Description

Find Peaks in the Spectra Data.

Usage

```
Met.peak.detection(datos)
```

Arguments

datos	Spectra data frame
-------	--------------------

Details

The alignment of peaks is an alternative to binning the spectrum to account for peak shifts. A peak detection graphical application (Preprocessing / Peak Detection) has been developed to control the "msc.peaks.find" function from caMassClass library. The graphical application adjusts the signal-to-noise ratio and the threshold criterion in the peak's detection process and it returns a data frame with the positions and intensities of the detected peaks. Launched with the GUI. Beta version.

Value

Peaks	Data frame with the detected peaks information
-------	--

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

caMassClass package <http://finzi.psych.upenn.edu/R/library/caMassClass/html/msc.peaks.find.html>

See Also

[Met.peak.detection](#)

Met.PLS1

Generalized Partial Least Squares

Description

Partial least squares is a commonly used dimension reduction technique. The code in this function uses the extension proposed by Ding and Gentleman, 2004.

Usage

```
Met.PLS1(datos, externa)
```

Arguments

datos	Spectra data frame
externa	Not implemented yet

Details

The PLS graphical application (Metabonomic Analysis / Partial Least Squares / PLS) has been developed with a PLS algorithm based on the extension of the generalized partial least squares model proposed by Ding and Gentleman. This algorithm is implemented using the "gpls" function from the "gpls" package and it allows one to separate only two classes of samples. The graphical application controls the manual or random selection of the samples to build the model, the selection of all the algorithm parameters as the tolerance to the convergence, the number of iterations allowed or the number of PLS components used. At the end, the results of the validation test will be returned. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

- gpls package <http://finzi.psych.upenn.edu/R/library/gpls/html/gpls.html>
- Ding, B.Y. and Gentleman, R. (2003) .Classification using generalized partial least squares.
- Marx, B.D (1996) .Iteratively reweighted partial least squares estimation for generalized linear regression. Technometrics 38(4): 374-381.

Met .PLS2

Partial Least Squares Regression

Description

Partial Least Squares Regression

Usage

```
Met.PLS2(datos, externa)
```

Arguments

datos	Spectra data frame
externa	Not implemented yet

Details

The application (Metabonomic Analysis / Partial Least Squares / PLS with graphics) is performed using the "plsr" function from the "pls" package . This PLS-DA is more complex and the user is guided by the application to execute all the steps in the proper order. Firstly, the user chooses between manual or random selection of the samples. Secondly, the user selects the PLS algorithm and the validation method. Four PLSR algorithms are available: the kernel algorithm, the wide kernel algorithm, the SIMPLS algorithm and the classical orthogonal scores algorithm. Next, the application creates a PLS model with the maximum number of components and it shows the explained variance and the R2 graphics of this model. With this information, the user can select the optimum number of PLS components to build the model. In addition, the Standard Error of Prediction (SEP) and the Root Mean Standard of Prediction (RMSEP) are plotted in the R console. Finally, the PLS graphical application returns the results of the validation test and different interactive graphics of the PLS model. Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

- pls package <http://finzi.psych.upenn.edu/R/library/pls/html/mvr.html>

Met . RadioBox1 *Met.RadioBox1*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met . RadioBox2 *Met.RadioBox2*

Description

Internal function.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met . redo *Undo/Redo*

Description

Redo function

Usage

Met . redo ()

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

See Also

[Met.undo](#)

Met.Save.Data	<i>Save data as txt file</i>
---------------	------------------------------

Description

Save data as txt file

Usage

```
Met.Save.Data()
```

Details

Launched with the GUI. Beta version

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.Selection	<i>Category Selection</i>
---------------	---------------------------

Description

The data are classified according to the information loaded in the info file.

Usage

```
Met.Selection(datos, externa.inicial)
```

Arguments

datos	Spectra data frame
externa.inicial	Not implemented yet

Details

This application selects the information that will be used in future supervised analyses. Firstly, the GUI asks which characteristic (different columns of the info file) will be used to classify the samples. Afterwards, the user chooses the different types of samples that will be utilized in the multivariate analyses. So far, the program only allows the selection of four different samples. The "Category Selection" application is launched selecting the "file/ Category Selection" tab. Launched with the GUI. Beta version.

Value

datos	Spectra data frame
...	Internal values
...	

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.Show	<i>Show current data</i>
----------	--------------------------

Description

Show current data.

Usage

Met.Show()

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met.spectrum	<i>Plot Spectrum</i>
--------------	----------------------

Description

Met.spectrum plots the original or the current spectra. Also, it can superimpose two spectrum.

Usage

Met.spectrum(xCoords)

Arguments

xCoords	data spectra
---------	--------------

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met . spectrum . 2 *Plot Spectrum*

Description

Met.spectrum plots the original or the current spectra. Also, it can superimpose two spectrum.

Usage

Met . spectrum . 2 (xCoords)

Arguments

xCoords data spectra

Details

This graphical display (Spectrum/...) has been created for the visualization and overlapping of the spectra. With these applications, the user can focus the interesting areas with a zoom tool, can superimpose different spectra, can increase or decrease the spectra intensity and can change other graphical parameters. Moreover, the user can click with the cross cursor in the spectrum and a new window pops-up showing the chemical shift and the intensity of this selected resonance. This display can be launched for the original or for the current spectra Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Met . undo *Undo/Redo*

Description

Undo function

Usage

Met . undo ()

Details

Launched with the GUI. Beta version.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

See Also

[Met.redo](#)

Metabonomic

GUI for the Metabonomic Analysis

Description

Graphical User Interface for the Metabonomic Analysis (Baseline, Normalization, Peak Detection, PCA, PLS, Nearest Neighbour, Neural Network) developed to make easy this data analysis.

Usage

Metabonomic ()

Details

The "Metabonomic" GUI was designed using the R-Tcl/Tk interface that allows using the TK toolkit, replacing Tcl code with R function calls. The purpose was to allow an easy interaction with the R functions as well as a comprehensive metabonomic analysis. The software offers several graphic outputs and the different plots are created using a combination of different Tcl/Tk interfaces. The program has been created with R version 2.8.0 under Windows operating system. To use the "Metabonomic" GUI, the "Metabonomic" package and all the required packages have to be downloaded and installed in the R console. The required PROcess package can be found in the Bioconductor Project Site . Next, the "Metabonomic" package has to be loaded using the Package installer or writing ">require (Metabonomic)" if the package is already in the computer. The program is started by writing "> Metabonomic()" in the R console, which generates the main user interface . The main GUI has an input console to launch any R applications and two different output consoles for the warnings and for the output messages. The GUI also has a button line, where the following buttons are available: (a) undo, (b) redo, (c) current data display, (d) launch the commands written in the input console, (e) erase the input console, (f) stop any running process and (g) shutdown the GUI and return to R console. Finally, the main GUI has a main menu with different tabs: File, Script, Edit, Preprocessing, Metabonomic Analysis and Spectrum. From the Script tab, the following functions are available: (a) "Load a Script" opens a script into the input console, (b) "Save Script" saves the commands written in the input console as an R script file and (c) "Launch the Script" runs the commands written in the input console.

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

`msc.biomarkers.fill`*Fill Empty Spaces in Biomarker Matrix*

Description

Fill empty spaces (NA's) in biomarker matrix created by `msc.peaks.align`. Transcription of `msc.biomarkers.fill` function of `caMassClass`

Usage

```
msc.biomarkers.fill(X, Bmrks, BinBounds, FillType = 0.9)
```

Arguments

X	Spectrum data either in matrix format [nFeatures x nSamples] or in 3D array format [nFeatures x nSamples x nCopies]. Row names (<code>rownames(X)</code>) store M/Z mass of each row.
Bmrks	biomarker matrix containing one sample per column and one biomarker per row
BinBounds	position (mass) of left-most and right-most peak in each bin
FillType	how to fill empty spaces in biomarker data? * if $0 \leq \text{FillType} \leq 1$ than fill spaces with <code>quantile(probs=FillType)</code> . For example: if <code>FillType=1/2</code> than medium will be used, if <code>FillType=1</code> than maximum value will be used, if <code>FillType=0.9</code> than maximum will be used after discarding 10% * if <code>FillType < 0</code> than empty spaces will not be filled and NA's will remain * if <code>FillType == 2</code> than X value closest to the center of the bin will be used * if <code>FillType == 3</code> empty spaces will be set to zero

Details

Transcription of `msc.biomarkers.fill` function of `caMassClass`

Value

Data in the same format and size as `Bmrks`

Author(s)

Jarek Tuszynski (SAIC) jaroslaw.w.tuszynski@saic.com

References

`caMassClass` <http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

msc.peaks.align *Align Peaks of Mass Spectra into a "Biomarker" Matrix*

Description

Align peaks from multiple protein mass spectra (SELDI) samples into a single "biomarker" matrix
 Transcription of msc.peaks.align function of caMassClass

Usage

```
msc.peaks.align(Peaks, SampFrac = 0.3, BinSize = c(0.002, 0.008), ...)
```

Arguments

Peaks	Peak information. Could have two formats: a filename where to find the data, or the data itself. In the first case, Peaks is string containing path to a file saved by msc.peaks.find, getPeaks (from PROcess package), or by other software. In the second case, it is a data-frame in the same format as returned by msc.peaks.find. A third way to pass the same input data is through use of S, M, H and Tag variables (described below) used by msc.peaks.alignment function.n.
SampFrac	After peak alignment, bins with fewer peaks than SampFrac*nSamp are removed.
BinSize	Upper and lower bound of bin-sizes, based on expected experimental variation in the mass (m/z) values. Size of any bin is measured as (R-L)/mean(R,L) where L and R are masses (m/z values) of left and right boundaries. All resulting bin sizes will all be between BinSize[1] and BinSize[2]. Since SELDI data is often assumed to have +- 3
...	Two additional parameters that can be passed to msc.peaks.clust are mostly for expert users fine-tuning the code: * tol - gaps bigger than tol*max(gap) are assumed to be the same size as the largest gap. See details. * verbose - boolean flag turns debugging printouts on.

Details

Two interfaces were provided to the same function:

* msc.peaks.alignment is a lower level function with more detailed inputs and outputs. Possibly easier to customize for other purposes than processing SELDI data. * msc.peaks.align is a higher level function with simpler interface customized for processing SELDI data.

This function aligns peaks from different samples into bins in such a way as to satisfy constraints in following order:

* bin sizes are in between BinSize[1] and BinSize[2] * no two peaks from the same sample are present in the same bin * bins are split in such a way as to minimize bin size and maximize spaces between bins * if there are multiple, equally good, ways to split a bin than bin is split in such a way as to minimize number of repeats on each smaller sub-bin

The algorithm used does the following:

* Store mass and sample number of each peak into an array * Concatenate arrays from all samples and sort them according to mass * Group sets of peaks into subsets (bins). Each subset will consist of peaks from different spectra that have similar mass. That is done by putting all peaks into a single bin and recursively going through the following steps:

- o Check size of the current bin: if it is too small than we are done, if it is too big than it will be split and if it is already in the desired range than it will be split only if multiple peaks from the same sample are present.
- o If bin needs to be split than find the biggest gap between peaks
- o If multiple gaps were found with the same size as the largest gap (or within tol tolerance from it) than minimizes number of multiple peaks from the same sample after cut
- o Divide the bin into two sub-bins: to the left and to the right of the biggest gap
- o Recursively repeat the above four steps for both sub-bins

 * Store peaks into 2D array (bins by samples) * Remove bins with fewer peaks than SampFrac*nSamp

The algorithm for peak alignment is described as recursive algorithm but the actual implementation uses internal stack, instead in order to increase speed.

Value

Bmrks	Biomarker matrix containing one sample per column and one biomarker per row. If a given sample does not have a peak in some bin than NA is inserted.
BinBounds	Mass of left-most and right-most peak in the bin

Author(s)

Jarek Tuszynski (SAIC) jaroslaw.w.tuszynski@saic.com

References

caMassClass <http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

msc.peaks.alignment

Align Peaks of Mass Spectra into a "Biomarker" Matrix

Description

Align peaks from multiple protein mass spectra (SELDI) samples into a single "biomarker" matrix
Transcription of msc.peaks.align function of caMassClass

Usage

```
msc.peaks.alignment(S, M, H, Tag = 0, SampFrac = 0.3, BinSize = c(0.002, 0.008), ..
```

Arguments

S	Peak sample number. Unique number of the sample the peak belongs to. Likely to come from Peaks\$Spectrum.
M	Peak center mass. Position of the peak on the x-axis. Likely to come from Peaks\$Substance.Mass.
H	Peak height. Likely to come from Peaks\$Intensity.
Tag	Peak sample name. Unique name of the sample the peak belongs to. Likely to come from Peaks\$Spectrum.Tag. Optional since is used only to set column-names of output data.
SampFrac	After peak alignment, bins with fewer peaks than SampFrac*nSamp are removed.
BinSize	Upper and lower bound of bin-sizes, based on expected experimental variation in the mass (m/z) values. Size of any bin is measured as (R-L)/mean(R,L) where L and R are masses (m/z values) of left and right boundaries. All resulting bin sizes will all be between BinSize[1] and BinSize[2]. Since SELDI data is often assumed to have +- 3
...	Two additional parameters that can be passed to msc.peaks.clust are mostly for expert users fine-tuning the code: * tol - gaps bigger than tol*max(gap) are assumed to be the same size as the largest gap. See details. * verbose - boolean flag turns debugging printouts on.

Details

Two interfaces were provided to the same function:

* msc.peaks.alignment is a lower level function with more detailed inputs and outputs. Possibly easier to customize for other purposes than processing SELDI data. * msc.peaks.align is a higher level function with simpler interface customized for processing SELDI data.

This function aligns peaks from different samples into bins in such a way as to satisfy constraints in following order:

* bin sizes are in between BinSize[1] and BinSize[2] * no two peaks from the same sample are present in the same bin * bins are split in such a way as to minimize bin size and maximize spaces between bins * if there are multiple, equally good, ways to split a bin than bin is split in such a way as to minimize number of repeats on each smaller sub-bin

The algorithm used does the following:

* Store mass and sample number of each peak into an array * Concatenate arrays from all samples and sort them according to mass * Group sets of peaks into subsets (bins). Each subset will consist of peaks from different spectra that have similar mass. That is done by putting all peaks into a single bin and recursively going through the following steps: o Check size of the current bin: if it is too small than we are done, if it is too big than it will be split and if it is already in the desired range than it will be split only if multiple peaks from the same sample are present. o If bin needs to be split than find the biggest gap between peaks o If multiple gaps were found with the same size as the largest gap (or within tol tolerance from it) than minimizes number of multiple peaks from the same sample after cut o Divide the bin into two sub-bins: to the left and to the right of the biggest gap o Recursively repeat the above four steps for both sub-bins * Store peaks into 2D array (bins by samples) * Remove bins with fewer peaks than SampFrac*nSamp

The algorithm for peak alignment is described as recursive algorithm but the actual implementation uses internal stack, instead in order to increase speed.

Value

Bmrks	Biomarker matrix containing one sample per column and one biomarker per row. If a given sample does not have a peak in some bin than NA is inserted.
BinBounds	Mass of left-most and right-most peak in the bin

Author(s)

Jarek Tuszynski (SAIC) jaroslaw.w.tuszynski@saic.com

References

<http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

msc.peaks.clust *Clusters Peaks of Mass Spectra*

Description

Clusters peaks from multiple protein mass spectra (SELDI) samples. Transcription of msc.peaks.clust function of caMassClass

Usage

```
msc.peaks.clust(dM, S, BinSize = c(0, sum(dM)), tol = 0.97, verbose = FALSE)
```

Arguments

dM	Distance between sorted peak positions (masses, m/z).
S	Peak sample number, used to identify the spectrum the peak come from.
BinSize	Upper and lower bound of bin-sizes, based on expected experimental variation in the mass (m/z) values. Size of any bin is measured as (R-L)/mean(R,L) where L and R are masses (m/z values) of left and right boundaries. All resulting bin sizes will be between BinSize[1] and BinSize[2]. Default is c(0,sum(dM)) which ensures that no BinSizes is not being used.
tol	gaps bigger than tol*max(gap) are assumed to be the same size as the largest gap. See details.
verbose	boolean flag turns debugging printouts on.

Details

This is a low level function used by `msc.peaks.alignment` and not intended to be directly used by many users. However it might be useful for other code developers. It clusters peaks from different samples into bins in such a way as to satisfy constraints in following order:

* bin sizes are in between `BinSize[1]` and `BinSize[2]` * no two peaks from the same sample are present in the same bin * bins are split in such a way as to minimize bin size and maximize spaces between bins * if there are multiple, equally good, ways to split a bin than bin is split in such a way as to minimize number of repeats on each smaller sub-bin

Value

The output is binary array of the same size as `dM` and `S` where left boundaries of each clusters-bin (biomarker) are marked

Author(s)

Jarek Tuszynski (SAIC) jaroslaw.w.tuszynski@saic.com

References

<http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

`msc.peaks.find` *Find Peaks of Mass Spectra*

Description

Find Peaks in a Batch of Protein Mass Spectra (SELDI) Data. Transcription of `msc.peaks.find` function of `caMassClass`

Usage

```
msc.peaks.find(X, SNR = 2, span = c(81, 11), zerothresh = 0.9)
```

Arguments

<code>X</code>	Spectrum data either in matrix format [<code>nFeatures</code> x <code>nSamples</code>] or in 3D array format [<code>nFeatures</code> x <code>nSamples</code> x <code>nCopies</code>]. Row names (<code>rownames(X)</code>) store M/Z mass of each row
<code>SNR</code>	signal to noise ratio (z-score) criterion for peak detection. Similar to <code>SoN</code> variable in <code>isPeak</code> from <code>PROcess</code> package.
<code>span</code>	two moving window widths. Smaller one will be used for smoothing and local maxima finding. Larger one will be used for local variance estimation. Similar to <code>span</code> and <code>sm.span</code> variables in <code>isPeak</code> from <code>PROcess</code> package.

`zerothresh` Intensity threshold criterion for peak detection. Positive numbers in range [0,1), like default 0.9, will be used to calculate a single threshold used for all samples using `quantile(X,zerothresh)` equation. Negative numbers in range (-1, 0) will be used to calculate threshold for each single sample *i* using `quantile(X[i],-zerothresh)`. Similar to `zerothrsh` variable in `isPeak` from `PROcess` package.

Details

Peak finding is done using the following algorithm: $x = X[j,]$ $\text{thresh} = \text{if}(\text{zerothresh} \geq 0) \text{quantile}(X, \text{zerothresh}) \text{ else } \text{quantile}(x, -\text{zerothresh})$ $\text{sig} = \text{runmean}(x, \text{span}[2])$ $\text{rMax} = \text{runmax}(x, \text{span}[2])$ $\text{rAvr} = \text{runmed}(x, \text{span}[1])$ $\text{rStd} = \text{runmad}(x, \text{span}[1], \text{center}=\text{rAvr})$ $\text{peak} = (\text{rMax} == x) \ \& \ (\text{sig} > \text{thresh}) \ \& \ (\text{sig} - \text{rAvr} > \text{SNR} * \text{rStd})$

What means that a peak have to meet the following criteria to be classified as a peak:

- * be a local maxima in `span[2]` neighborhood
- * smoothed sample (`sig`) is above user defined threshold `zerothresh`
- * locally calculated z-score (see <http://mathworld.wolfram.com/z-Score.html>) of the signal is above user defined signal-to-noise ratio

It is very similar to the `isPeak` and `getPeaks` functions from `PROcess` library (ver 1.3.2) written by Xiaochun Li. For example `getPeaks(X, PeakFile, SoN=SNR, span=span[1], sm.span=span[2], zerothrsh=zerothresh, area.w=0.003, ratio=0)` would give very similar results as `msc.peaks.find` the differences include: speed (`msc.peaks.find` uses much faster C-level code), different use of signal-to-noise-ratio variable, and `msc.peaks.find` does not do or use area calculations.

Value

A data frame, in the same format as data saved in `peakinfofile`, have five components:

<code>Spectrum.Tag</code>	sample name of each peak
<code>Spectrum.</code>	sample number of each peak
<code>Intensity</code>	peak height (intensity)
<code>Substance.Mass</code>	x-axis position, or corresponding mass of the peak measured in <i>M/Z</i> , which were extracted from row names of the <i>X</i> matrix.

Author(s)

Jarek Tuszynski (SAIC) jaroslaw.w.tuszynski@saic.com

References

<http://finzi.psych.upenn.edu/R/library/caMassClass/html/00Index.html>

Phase.correction *Phase Correction*

Description

Zero order and first order phase corrections

Usage

```
Phase.correction()
```

Details

This correction is composed by a frequency independent parameter, or zero phase correction and a first order phase correction or linear dependent on the frequency parameter. The zero order and first order parameters (including the pivot)selection is interactive and intuitive process controlled by the user

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

References

R.R. Ernst, G. Bodenhausen, A. Wokaun. Principles of Nuclear Magnetic Resonance in One and Two Dimensions. Clarendon Press, Oxford, 2003.

See Also

[Import.data](#),

Require *Require*

Description

Graphical interface of the require function. Internal function.

Usage

```
Require(pkg)
```

Arguments

pkg package

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

 showData2

Display a Data Frame in a Tk Text Widget

Description

Modification of the showData function of the relimp package.

Usage

```
showData2(dataframe, colname.bgcolor , rowname.bgcolor , body.bgcolor , colname.tex
```

Arguments

dataframe	A data frame, or an object to which as.data.frame() can be validly applied
colname.bgcolor	A background colour for the variable_names panel
rowname.bgcolor	A background colour for the row_names panel
body.bgcolor	A background colour for the data
colname.textcolor	A colour for the variable names
rowname.textcolor	A colour for the row names
body.textcolor	A colour for the data
font	The text font used . should be a monospaced font
maxheight	The maximum number of rows to display
maxwidth	The maximum width of display, in characters
title	A title for the window. Default is to use the name of the dataframe as given in the call to showData()
rowname.bar	position of sidebar for row names, "left" or "right", or c("left","right"), or NULL
colname.bar	position of column names, "top" or "bottom", or c("top","bottom"), or NULL
rownnumbers	logical, whether row numbers should be displayed
placement	Position of the bottom right corner of the window
suppress.X11.warnings	logical, if TRUE then any X11 warnings are suppressed

Value

invisible NULL

References

relimp package <http://finzi.psych.upenn.edu/R/library/relimp/html/showData.html>

TABLE

Table

Description

Graphical interface of the table function. Internal function.

Usage

```
TABLE(a, b, title = "")
```

Arguments

a	True vector
b	Model decision vector
title	Name of the Model

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Try

Try function

Description

Internal function.

Usage

```
Try(expr)
```

Arguments

expr	Expression to try
------	-------------------

Author(s)

Jose L. Izquierdo <izquierdo@ieb.ucm.es>

Index

*Topic interface

- Baseline.Correction, 2
- Import.data, 3
- Manual.cut, 4
- manual.model, 5
- Met.Align, 5
- Met.B.STAT, 6
- Met.Binning, 6
- Met.Checkbox1, 7
- Met.delete, 7
- Met.KNN, 8
- Met.kopls, 9
- Met.LDA, 10
- Met.Load.Data, 11
- Met.Metabolites, 12
- Met.modalDialog, 12
- Met.modalDialog2, 13
- Met.modalDialog3, 13
- Met.modalDialog4, 13
- Met.model.1, 14
- Met.NN1, 14
- Met.NN2, 15
- Met.Norm, 16
- Met.Order, 17
- Met.PCA, 17
- Met.peak.detection, 18
- Met.PLS1, 19
- Met.PLS2, 20
- Met.RadioBox1, 21
- Met.RadioBox2, 21
- Met.redo, 21
- Met.Save.Data, 22
- Met.Selection, 22
- Met.Show, 23
- Met.spectrum, 23
- Met.spectrum.2, 24
- Met.undo, 24
- Metabonomic, 25
- Metabonomic-package, 1

- msc.biomarkers.fill, 26
- msc.peaks.align, 27
- msc.peaks.alignment, 28
- msc.peaks.clust, 30
- msc.peaks.find, 31
- Phase.correction, 33
- Require, 33
- showData2, 34
- TABLE, 35
- Try, 35

- Baseline.Correction, 2

- Import.data, 3, 33

- Manual.cut, 4
- manual.model, 5
- Met.Align, 5
- Met.B.STAT, 6
- Met.Binning, 6
- Met.Checkbox1, 7
- Met.delete, 7
- Met.KNN, 8
- Met.kopls, 9
- Met.LDA, 10
- Met.Load.Data, 11
- Met.Metabolites, 12
- Met.modalDialog, 12
- Met.modalDialog2, 13
- Met.modalDialog3, 13
- Met.modalDialog4, 13
- Met.model.1, 14
- Met.NN1, 14
- Met.NN2, 15
- Met.Norm, 16
- Met.Order, 17
- Met.PCA, 17
- Met.peak.detection, 5, 18, 19
- Met.PLS1, 19
- Met.PLS2, 20

Met.RadioBox1, 21
Met.RadioBox2, 21
Met.redo, 21, 25
Met.Save.Data, 22
Met.Selection, 22
Met.Show, 23
Met.spectrum, 23
Met.spectrum.2, 24
Met.undo, 21, 24
Metabonomic, 25
Metabonomic
 (*Metabonomic-package*), 1
Metabonomic GUI (*Metabonomic*), 25
Metabonomic-package, 1
msc.biomarkers.fill, 26
msc.peaks.align, 27
msc.peaks.alignment, 28
msc.peaks.clust, 30
msc.peaks.find, 31

Phase.correction, 33

Require, 33

showData2, 34

TABLE, 35
Try, 35