Package 'PepSAVIms'

July 21, 2025

Type Package

Title PepSAVI-MS Data Analysis

Version 0.9.1

Date 2016-12-16

Description An implementation of the data processing and data analysis portion of a pipeline named the PepSAVI-MS which is currently under development by the Hicks laboratory at the University of North Carolina. The statistical analysis package presented herein provides a collection of software tools used to facilitate the prioritization of putative bioactive peptides from a complex biological matrix. Tools are provided to deconvolute mass spectrometry features into a single representation for each peptide charge state, filter compounds to include only those possibly contributing to the observed bioactivity, and prioritize these remaining compounds for those most likely contributing to each bioactivity data set.

License CC BY-NC-SA 4.0

URL https://github.com/dpritchLibre/PepSAVIms

BugReports https://github.com/dpritchLibre/PepSAVIms/issues

LazyData TRUE

Depends R (>= 3.0.0)

Suggests testthat, knitr

Imports elasticnet

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Author Pritchard David [aut, cre],
Kirkpatrick Christine [aut]

Maintainer Pritchard David <dpritch@live.unc.edu>

Repository CRAN

Date/Publication 2016-12-17 01:38:41

Contents

`binMS` *Consolidate mass spectrometry observations*

Description

Combines mass spectrometry observations that are believed to belong to the same underlying compound into a single observation. In concept, the data produced by the mass spectrometer may produce multiple reads for a single compound; thus, `binMS` attempts to recover these underlying compounds through a binning procedure, described in more detail in `Details`.

Usage

```
binMS(mass_spec, mtoz, charge, mass = NULL, time_peak_reten,
  ms_inten = NULL, time_range, mass_range, charge_range, mtoz_diff, time_diff)
```

Arguments

`mass_spec`	Either a `matrix` or `data.frame`. This object must contain mass spectrometry abundances, and may optionally contain mass-to-charge values, charge state information, or additional extraneous variables. The mass spectrometry data is expected to be in a form with each column corresponding to a variable and each row corresponding to a mass-to-charge level. For example, suppose that a collection of mass spectrometry intensity observations has provided data for 50 fractions across 20,000 mass-to-charge values. Then the input for `mass_spec` should be a `matrix` or `data.frame` with 20,000 rows and 50 or more columns. The additional columns beyond the 50 containing the mass spectrometry intensities can be the mass-to-charge data, the charge data, or other extraneous variables (the extraneous variables will be discarded when constructing the `msDat` object).

`mtoz` A vector of either length 1 or length equal to the number of mass-to-charge values for which mass spectrometry data was collected, and which helps identify the mass-to-charge values for this data in one of several ways.

One way to provide the information is to provide a numeric vector where each entry provides the mass-to-charge value for a corresponding row of mass spectrometry data. Then the k-th entry of the vector would provide the mass-to-charge value for the k-th row of the mass spectrometry data.

A second way is to provide a single number which specifies the column index in the `matrix` or `data.frame` provided as the argument for the `mass_spec` parameter, such that this column contains the mass-to-charge information.

A third way is provide a single character string which provides the column name in the `matrix` or `data.frame` provided as the argument for the `mass_spec` parameter, such that this column contains the mass-to-charge information. Partial matching is supported.

`charge` The information for the `charge` parameter can be provided in the same manner as for the mass-to-charge values.

`mass` The information for the mass need not be provided, as it can be derived using the mass-to-charge and charge information; in this case the parameter should be given its default, i.e. `NULL`. If however the information for mass is already included in the dataset in hand, then providing it to the function will be slightly more efficient then re-performing the calculations. The information for the `charge` parameter can be provided in the same manner as for the mass-to-charge values.

`time_peak_reten`

The information for the `time_peak_reten` parameter can be provided in the same manner as for the mass-to-charge and other information; this paramater specifies the time at which the peak retention level of the compound was achieved.

`ms_inten` Either `NULL` or a vector either of mode character or mode numeric specifying which of the variables in the argument to `mass_spec` are to be retained as the mass spectrometry intensity data. If `NULL`, then it is taken to mean that the entirety of the data in `mass_spec`, after removing variables in the data that are specified as arguments, is the mass spectrometry intensity data. If it is a numeric vector, then the entries should provide the indices for the region of interest in the mass spectrometry data in the argument for `msObj`. If it is a character vector, then the entries should uniquely specify the region of interest through partial string matching.

`time_range` A length-2 numeric vector specifying the lower bound and upper bound (inclusive) of allowed peak retention time occurance for an observation to be included in the consolidation process.

`mass_range` A length-2 numeric vector specifying the lower bound and upper bound (inclusive) of allowed mass for an observation to be included in the consolidation process.

`charge_range` A length-2 numeric vector specifying the lower bound and upper bound (inclusive) of allowed electrical charge state for an observation to be included in the consolidation process.

`mtoz_diff` A single numerical value such that any two observations with a larger absolute difference between their mass-to-charge values are considered to have originated from different underlying compounds. Two observations with a smaller absolute difference between their mass-to-charge values could potentially be considered to originate from the same underlying compound, contingent on other criteria also being met. Nonnegative values are allowed; such a value has the effect of not consolidating any groups, and consequently reduces the function to a filtering routine only.

`time_diff` A single numerical value such that any two observations with a larger absolute difference between their peak elution times are considered to have originated from different underlying compounds. Two observations with a smaller absolute difference between their peak elution times could potentially be considered to originate from the same underlying compound, contingent on other criteria also being met. Nonnegative values are allowed; such a value has the effect of not consolidating any groups, and consequently reduces the function to a filtering routine only.

Details

The algorithm described in what follows attempts to combines mass spectrometry observations that are believed to belong to the same underlying compound into a single observation for each compound. There are two conceptually separate steps.

The first step is as follows. All observations must satisfy each of the following criteria for inclusion in the binning process.

1. Each observation must have its peak elution time occur during the interval specified by `time_range`
2. Each observation must have a mass that falls within the interval specified by `mass_range`
3. Each observation must have an electrical charge state that falls within the interval specified by `charge_range`

Once that a set of observations satisfying the above criteria is obtained, then a second step attempts to combine observations believed to belong to the same underlying compound. The algorithm considers two observations that satisfy each of the following criteria to belong to the same compound.

1. The absolute difference in Daltons of the mass-to-charge value between the two observations is less the the value specified by `mtoz_diff`
2. The absolute difference of the peak elution time between the two observations is less than the value specified by `time_pr_diff`
3. The electrical charge state must be the same for the two observations

Then the binning algorithm is defined as follows. Consider an observation that satisfies the inclusion criteria; this observation is compaired pairwise with every other observation that satisfies the inclusion criteria. If a pair of observations satisfies the criteria determining them to belong to the same underlying compound then the two observations are merged into a single observation. The two previous compounds are removed from the working set, and the process starts over with the newly created observation. The process repeats until no other observation in the working set meets

the criteria determining it to belong to the same underlying compound as that of the current observation; at this point it is considered that all observations belonging to the compound have been found, and the process starts over with a new observation.

The merging process has not yet been defined; it is performed by averaging the mass-to-charge values and peak elution times, and summing the mass spectrometry intensities at each fraction. Although observations are merged pairwise, when multiple observations are combined in a sequence of pairings, the averages are given equal weight for all of the observations. In other words, if a pair of observations are merged, and then a third observation is merged with the new observation created by combining the original two, then the mass-to-charge value and peak elution time values of the new observation are obtained by summing the values for each of the three original observations and dividing by three. The merging process for more than three observations is conducted similarly.

Having described the binning algorithm, it is apparent that there are scenarios in which the order in which observations are merged affects the outcome of the algorithm. Since it seems that a minumum requirement of any binning algorithm is that the algorithm is invariant to the ordering of the observations in the data, this algorithm abides by the following rules. The observations in the data are sorted in increasing order by mass-to-charge value, peak elution time, and electical charge state, respectively. Then when choosing an observation to compare to the rest of the set, we start with the observation at the top of the sort ordering, and compare it one-at-a-time to the other elements in the set according to the same ordering. When a consolidated observation is complete in that no other observation left in the working set satisfies the merging criteria, then this consolidated observation can be removed from consideration for all future merges.

Value

Returns an object of class `binMS` which inherits from `msDat`. This object is a `list` with elements described below. The class is equipped with a `print`, `summary`, and `extractMS` function.

`msDatObj` An object of class `msDat` that encapsulates the mass spectrometry data for the consolidated data.

`summ_info` A list containing information pertaining to the consolidation process; for use by the summary function.

Examples

```
# Load mass spectrometry data
data(mass_spec)

# Perform consolidation via binMS
bin_out <- binMS(mass_spec = mass_spec,
                 mtoz = "m/z",
                 charge = "Charge",
                 mass = "Mass",
                 time_peak_reten = "Reten",
                 ms_inten = NULL,
                 time_range = c(14, 45),
                 mass_range = c(2000, 15000),
                 charge_range = c(2, 10),
                 mtoz_diff = 0.05,
                 time_diff = 60)
```

```
# print, summary function
bin_out
summary(bin_out)

# Extract consolidated mass spectrometry data as a matrix or msDat object
bin_matr <- extractMS(msObj = bin_out, type = "matrix")
bin_msDat <- extractMS(msObj = bin_out, type = "matrix")
```

`bioact` *Bioactivity data*

Description

The relative relative growth inhibition of bioactivity levels for the bacteria and virus strains studies in Kirkpatrick et al. (2016).

Usage

```
data(bioact)
```

Format

A `list` containing relative growth inhibition of bioactivity levels for the bacteria and virus strains listed below. Each of the following elements in the `list` is a `data.frame` with 3 rows and 44 columns (with the exception of fg ********* which has 2 rows). The rows in each `data.frame` correspond to replications of the data collection process, while the columns correspond to relative growth inhibition bioactivity levels when subject to peptide libraries across fractions 1-43 and fraction 47.

ec E. Coli

bc S. aureus

pc K. pneumoniae

oc A. baumannii

ef E. cloacae

ab ??

pa ??

fg ??

extractMS *Extract embedded mass spectrometry data*

Description

Extract mass spectrometry data from an object with class binMS, class filterMS, or class msDat.

Usage

```
extractMS(msObj, type = "matrix")
```

Arguments

msObj	An an object with class binMS, class filterMS, or class msDat.
type	A character string with value either "matrix", or "msDat". If "matrix" is provided as the argument, then the mass-to-charge values, charge values, and mass spectrometry data are combined into a single matrix and returned. If "msDat" is provided as the argument, then an msDat object containing this data is returned.

Details

A convenience function for extracting and inspecting the mass spectrometry data in a binMS, filterMS, or msDat object. binMS and filterMS objects are lists that contain an msDat object, and specifying "msDat" for type merely returns the msDat element from the list for these classes of object. specifying "msDat" for an object with class "msDat" merely returns the argument, i.e. is the identity function. When "matrix" is specified, then the elements in the embedded msDat object are combined into a single matrix using cbind and returned.

Value

Returns either a matrix containing the mass spectrometry data if "matrix" is specified as the argument to type, or an object with class msDat if "msDat" is specified as the argument to type. See Details for more detail regarding the return objects.

extract_ranked *Extract candidate compounds*

Description

Extract an ordered list of candidate compounds from a rankEN object. The list is presented in the form of a data.frame, such that each row provides the identifying information for a particular candidate compound, and with the rows arranged in the order that the compounds entered the elastic net model (i.e. row 1 is the earliest, row 2 the 2nd earliest, etc.). The columns of the data.frame provide the mass-to-charge information, charge information, and possibly the correlation between the compound and the within-fraction average of the bioactivity replicates in the region of interest.

Usage

```
extract_ranked(rankEN_obj, include_cor = TRUE)
```

Arguments

rankEN_obj	An object of class rankEN.
include_cor	Either TRUE or FALSE, specifying whether a column should be included in the returning data.frame providing the correlation between the compound and the within-fraction average of the bioactivity replicates in the region of interest.

---

filterMS *Filter compounds from mass spectrometry data*

Description

Filters mass spectrometry data using a set of criteria, described in Details. Returns an object of classes msDat and filterMS.

Usage

```
filterMS(msObj, region, border = "all", bord_ratio = 0.05,
  min_inten = 1000, max_chg = 7L)
```

Arguments

msObj	An object class msDat. Note that this includes objects created by the functions binMS and msDat.
region	A vector either of mode character or mode numeric. If numeric then the entries should provide the indices for the region of interest in the mass spectrometry data provided as the argument for msObj. If character then the entries should uniquely specify the region of interest through partial string matching (see criterion 1, 4).
border	Either a character string "all", or a character string "none", or a length-1 or length-2 numeric value specifying the number of fractions to either side of the region of interest to comprise the bordering region. If a single numeric value, then this is the number of fractions to each side of the region of interest; if it is two values, then the first value is the number of fractions to the left, and the second value is the number of fractions to the right. If there are not enough fractions in either direction to completely span the number of specified fractions, then all of the available fractions to the side in question are considered to be part of the bordering region (see criterion 2).
bord_ratio	A single nonnegative numeric value. A value of 0 will not admit any compounds, while a value greater than 1 will admit all compounds (see criterion 2).
min_inten	A single numeric value. A value less than the minimum mass spectrometry value in the data will admit all compounds (see criterion 4).
max_chg	A single numeric value specifying the maximum charge which a compound may exhibit (see criterion 5)

Details

Attempts to filter out candidate compounds via subject-matter knowledge, with the goal of removing spurious noise from downstream models. The criteria for the downstream inclusion of a candidate compound is listed below.

1. The m/z intensity maximum must fall inside the range of the bioactivity region of interest
2. The ratio of the m/z intensity of a species in the areas bordering the region of interest and the species maximum intensity must be less than `bord_ratio`. When there is no bordering area then it is taken to mean that all observations satisfy this criterion.
3. The immediately right adjacent fraction to its maximum intensity fraction for a species must have a non-zero abundance. In the case of ties for the maximum, it is the fraction immediately to the right of the rightmost maximum fraction which cannot have zero abundance. When the fraction with maximum intensity is the rightmost fraction in the data for an observation, then it is taken to mean that the observation satisfies this criterion.
4. At least 1 fraction in the region of interest must have intensity greater than `min_inten`
5. Compound charge state must be less than or equal to `max_chg`

Value

Returns an object of class `filterMS` which inherits from `msDat`. This object is a `list` with elements described below. The class is equipped with a `print`, `summary`, and `extractMS` function.

`msDatObj` An object of class `msDat` such that the encapsulated mass spectrometry data corresponds to each of the candidate compounds that satisfed each of the criteria. If no criteria are satisfied then `NULL` is returned.

`cmp_by_crit` A list containing `data.frames`, one for each criterion. Each row (if any) in one of the sub-`data.frames` contains the mass-to-charge and charge information for a candidate compound that satisfies the criterion represented by the `data.frame`; all of the compounds that satisfied the criterion are included in the data. The `data.frames` are named `c1`, ..., `c5`, etc corresponding to criterion 1, ..., criterion 5.

`summ_info` A list containing information pertaining to the filtering process; for use by the summary function.

Examples

```
# Load mass spectrometry data
data(mass_spec)

# Convert mass_spec from a data.frame to an msDat object
ms <- msDat(mass_spec = mass_spec,
            mtoz = "m/z",
            charge = "Charge",
            ms_inten = c(paste0("_", 11:43), "_47"))

# Filter out potential candidate compounds
filter_out <- filterMS(msObj = ms,
                       region = paste0("VO_", 17:25),
                       border = "all",
```

```
                        bord_ratio = 0.01,
                        min_inten = 1000,
                        max_chg = 7)

# print, summary function
filter_out
summary(filter_out)

# Extract filtered mass spectrometry data as a matrix or msDat object
filter_matr <- extractMS(msObj = filter_out, type = "matrix")
filter_msDat <- extractMS(msObj = filter_out, type = "matrix")
```

`mass_spec` *Mass spectrometry data*

Description

The mass spectrometry data collected for and described in Kirkpatrick et al. (2016). See paper for a full description of the data collection process, or the package vignette for an abridged description.

Usage

```
data(mass_spec)
```

Format

A `data.frame` with 30,799 mass spectrometry levels and 38 variables:

m/z mass-to-charge ratio

Retention time (min) The time in minutes at which the peak retention time was achieved

Mass mass in daltons

Charge electrical charge state

20150207_CLK_BAP_VO_11 intensity state at fraction 11

20150207_CLK_BAP_VO_12 intensity state at fraction 12

20150207_CLK_BAP_VO_13 intensity state at fraction 13

20150207_CLK_BAP_VO_14 intensity state at fraction 14

20150207_CLK_BAP_VO_15 intensity state at fraction 15

20150207_CLK_BAP_VO_16 intensity state at fraction 16

20150207_CLK_BAP_VO_17 intensity state at fraction 17

20150207_CLK_BAP_VO_18 intensity state at fraction 18

20150207_CLK_BAP_VO_19 intensity state at fraction 19

20150207_CLK_BAP_VO_20 intensity state at fraction 20

20150207_CLK_BAP_VO_21 intensity state at fraction 21

20150207_CLK_BAP_VO_22 intensity state at fraction 22

20150207_CLK_BAP_VO_23 intensity state at fraction 23

20150207_CLK_BAP_VO_24 intensity state at fraction 24

20150207_CLK_BAP_VO_25 intensity state at fraction 25

20150207_CLK_BAP_VO_26 intensity state at fraction 26

20150207_CLK_BAP_VO_27 intensity state at fraction 27

20150207_CLK_BAP_VO_28 intensity state at fraction 28

20150207_CLK_BAP_VO_29 intensity state at fraction 29

20150207_CLK_BAP_VO_30 intensity state at fraction 30

20150207_CLK_BAP_VO_31 intensity state at fraction 31

20150207_CLK_BAP_VO_32 intensity state at fraction 32

20150207_CLK_BAP_VO_33 intensity state at fraction 33

20150207_CLK_BAP_VO_34 intensity state at fraction 34

20150207_CLK_BAP_VO_35 intensity state at fraction 35

20150207_CLK_BAP_VO_36 intensity state at fraction 36

20150207_CLK_BAP_VO_37 intensity state at fraction 37

20150207_CLK_BAP_VO_38 intensity state at fraction 38

20150207_CLK_BAP_VO_39 intensity state at fraction 39

20150207_CLK_BAP_VO_40 intensity state at fraction 40

20150207_CLK_BAP_VO_41 intensity state at fraction 41

20150207_CLK_BAP_VO_42 intensity state at fraction 42

20150207_CLK_BAP_VO_43 intensity state at fraction 43

20150207_CLK_BAP_VO_47 intensity state at fraction 47

`msDat` *Constructor for class* `msDat`

Description

Creates a data structure encapsulating the mass spectrometry intensity readings as well as identifying information

Usage

```
msDat(mass_spec, mtoz, charge, ms_inten = NULL)
```

Arguments

mass_spec	Either a matrix or data.frame. This object must contain mass spectrometry abundances, and may optionally contain mass-to-charge values, charge state information, or additional extraneous variables. The mass spectrometry data is expected to be in a form with each column corresponding to a variable and each row corresponding to a mass-to-charge level. For example, suppose that a collection of mass spectrometry intensity observations has provided data for 50 fractions across 20,000 mass-to-charge values. Then the input for mass_spec should be a matrix or data.frame with 20,000 rows and 50 or more columns. The additional columns beyond the 50 containing the mass spectrometry intensities can be the mass-to-charge data, the charge data, or other extraneous variables (the extraneous variables will be discarded when constructing the msDat object).
mtoz	A vector of either length 1 or length equal to the number of mass-to-charge values for which mass spectrometry data was collected, and which helps identify the mass-to-charge values for this data in one of several ways. One way to provide the information is to provide a numeric vector where each entry provides the mass-to-charge value for a corresponding row of mass spectrometry data. Then the k-th entry of the vector would provide the mass-to-charge value for the k-th row of the mass spectrometry data. A second way is to provide a single number which specifies the column index in the matrix or data.frame provided as the argument for the mass_spec parameter, such that this column contains the mass-to-charge information. A third way is provide a single character string which provides the column name in the matrix or data.frame provided as the argument for the mass_spec parameter, such that this column contains the mass-to-charge information. Partial matching is supported.
charge	The information for the charge parameter can be provided in the same manner as for the mass-to-charge values.
ms_inten	Either NULL or a vector either of mode character or mode numeric specifying which of the variables in the argument to mass_spec are to be retained as the mass spectrometry intensity data. If NULL, then it is taken to mean that the entirety of the data in mass_spec, after removing variables in the data that are specified as arguments, is the mass spectrometry intensity data. If it is a numeric vector, then the entries should provide the indices for the region of interest in the mass spectrometry data in the argument for msObj. If it is a character vector, then the entries should uniquely specify the region of interest through partial string matching.

Details

Since the mass spectrometry data could conceivably be available to the researcher in a variety forms, this function attempts to provide a uniform data structure for encapsulating this information. It is the fundamental data structure containing the mass spectrometry data used internally by the filterMS and rankEN routines. The external interface for msDat is provided to the user so that specifying the mass spectrometry information can be made in a distinct step from performing statistical analyses,

which it is hoped makes interfaces for the downstream analysis routines simpler and more intuitive to use.

Value

Returns an object of class `msDat`. This class is a `list` with elements described below. The class is equipped with a `print` and `extractMS` function.

`ms` A `matrix` containing mass spectrometry intensity readings. Each column provides the mass spectrometry values for a given fraction, and each row provides the mass spectrometry values for a given mass-to-charge ratio value across the fractions.

`mtoz` A vector with length equal to the number of mass-to-charge values provided in the mass spectrometry data, such that the k-th entry in the vector provides the mass-to-charge value for the k-th row of mass spectrometry data

`chg` A vector with length equal to the number of mass-to-charge values provided in the mass spectrometry data, such that the k-th entry in the vector provides the charge information for the k-th row of mass spectrometry data

Examples

```
# Load mass spectrometry data
data(mass_spec)

# Convert mass_spec from a data.frame to an msDat object
ms <- msDat(mass_spec = mass_spec,
            mtoz = "m/z",
            charge = "Charge",
            ms_inten = c(paste0("_", 11:43), "_47"))

# Dimension of the data
dim(ms)

# Print the first few rows and columns
ms[1:5, 1:2]

# Let's change the fraction names to something more concise
colnames(ms) <- c(paste0("frac", 11:43), "frac47")

# Print the first few rows and columns with the new fraction names
ms[1:5, 1:8]

# Suppose there are some m/z levels that we wish to remove
ms <- ms[-c(2, 4), ]
# Print the first few rows and columns after removing rows 2 and 4
ms[1:5, 1:8]

# Suppose that there was an instrumentation error and that we need to change
# some values
ms[1, paste0("frac", 12:17)] <- c(55, 57, 62, 66, 71, 79)
# Print the first few rows and columns after changing some of the values in
# the first row
```

```
ms[1:5, 1:10]
```

`print.filterMS` *Basic information for class* `filterMS`

Description

Displays the number of candidate compounds left in the data after filtering

Usage

```
## S3 method for class 'filterMS'
print(x, ...)
```

Arguments

`x`	An object of class `filterMS`
`...`	Arguments passed to dot-dot-dot are ignored

`print.msDat` *Print method for class* `msDat`

Description

Prints the mass spectrometry data encapsulated by the `msDat` object

Usage

```
## S3 method for class 'msDat'
print(x, ...)
```

Arguments

`x`	An object of class `msDat`
`...`	Arguments passed to dot-dot-dot are ignored

print.rankEN *Basic information for class* rankEN

Description

Displays the data dimensions used to fit the elastic net model

Usage

```
## S3 method for class 'rankEN'
print(x, ...)
```

Arguments

x	An object of class rankEN
...	Arguments passed to dot-dot-dot are ignored

rankEN *Rank compounds via the Elastic Net path*

Description

Returns identifying information for the compounds in the order in which the corresponding regression coefficient for a given compound first becomes nonzero as part of the Elastic Net path

Usage

```
rankEN(msObj, bioact, region_ms = NULL, region_bio = NULL, lambda,
  pos_only = TRUE, ncomp = NULL)
```

Arguments

msObj	An object of class msDat containing mass spectrometry abundances data and identifying information. Note that this includes objects created by the functions binMS, filterMS, and msDat.
bioact	Either a numeric vector or matrix, or a data frame providing bioactivity data. If a numeric vector, then it is assumed that each entry corresponds to a particular fraction. If the data is 2-dimensional, then it is assumed that each column corresponds to a particular fraction, and that each row corresponds to a particular bioactivity replicate.

region_ms — Either NULL, or a vector either of mode character or mode numeric providing information specifying which fractions from the mass spectrometry abundances data are to be included in the data analysis. If NULL, then it is assumed that the entirety of the mass spectrometry abundances data encapsulated in the argument to msObj is to be included in the analysis. If numeric then the entries should provide the indices for the region of interest in the mass spectrometry data (i.e. the indices of the columns corresponding to the appropriate fractions in the data). If character then the entries should uniquely specify the region of interest through partial string matching (i.e. the names of the columns corresponding to the appropriate fractions in the data). The methods dim, dimnames, and colnamesMS can be used as interfaces to the mass spectrometry data encapsulated in msObj.

region_bio — Either NULL, or a vector either of mode character or mode numeric providing information specifying which fractions from the bioactivity data are to be included in the data analysis. If NULL, then it is assumed that the entirety of bioactivity data provided as the argument to bioact is to be included in the analysis. If numeric then the entries should provide the indices for the region of interest in the bioactivity data (i.e. the indices of the columns corresponding to the appropriate fractions in the data). If character then the entries should uniquely specify the region of interest through partial string matching (i.e. the names of the columns corresponding to the appropriate fractions in the data).

lambda — A single nonnegative numeric value providing the quadratic penalty mixture parameter argument for the elastic net model. The elastic net fits the least squares model with penalty function

$$\gamma|\beta|_1 + \lambda|\beta|^2$$

where β is the vector of regression coefficients and $\gamma, \lambda \geq 0$. rankEN constructs a list of candidate compounds by tracking the entrance of compounds into the elastic net model as γ is decreased from ∞ to 0.

pos_only — Either TRUE or FALSE; specifies whether the list of candidate compounds that the algorithm produces should include only those compounds that are positively correlated with bioactivity levels, or conversely should include all compounds. The correlation is calculated using only observations from the region of interest, and when bioactivity replicates are present, the within-fraction replicates are averaged prior to calculation.

ncomp — Either NULL, or a numeric value no less than 1 specifying the maximum number of candidate compounds that the function should report. When NULL, this is taken to mean that all compounds that enter the model should be reported, possibly after removing compounds nonpositively correlated with bioactivity levels, as specified by pos_only.

Details

rankEN prepares the data by extracting the region of interest from the mass spectrometry abundance data and from the bioactivity data. If bioactivity replicates are present, then the within-fraction replicates are averaged. Once the data has been converted into the appropriate form, then an elastic net model is fitted by invoking the enet function from the elasticnet package, and an ordered list of candidate compounds is constructed such that compounds are ranked by the order in which they

first enter the model. The list may be filtered and / or pruned before being returned to the user, as determined by the arguments to pos_only and ncomp.

Value

Returns an object of class rankEN. This object is a list with elements described below. The class is equipped with a print, summary, and extract_ranked function.

mtoz A vector providing the mass-to-charge values of the candidate compounds, such that the k-th element of the vector provides the mass-to-charge value of the k-th compound to enter the elastic net model, possibly after removing compounds nonpositively correlated with bioactivity levels.

charge A vector providing the charge state of the candidate compounds, such that the k-th element of the vector provides the charge state of the k-th compound to enter the elastic net model, possibly after removing compounds nonpositively correlated with bioactivity levels.

comp_cor A vector providing the correlation between each of the candidate compounds and the bioactivity levels, such that the k-th element of the vector provides the correlation between the k-th compound to enter the elastic net model and the bioactivity levels, possibly after removing compounds nonpositively correlated with bioactivity levels.

enet_fit The fitted model object produced by rankEN's internal invokation of the enet function from the elasticnet package.

summ_info A list containing information related to the data used to fit the elastic net model; used by the summary function.

Examples

```
# Load mass spectrometry data
data(mass_spec)

# Convert mass_spec from a data.frame to an msDat object
ms <- msDat(mass_spec = mass_spec,
            mtoz = "m/z",
            charge = "Charge",
            ms_inten = c(paste0("_", 11:43), "_47"))

# Load growth inhibition bioactivity data.  Each element in bioact is a
# stand-alone dataset for a species of virus or bacteria.
data(bioact)

# Perform the candidate ranking procedure with fractions 21-24 as the region
# of interest.  Note that it is not advisable to calculate the elastic net
# estimates with 30,799 candidate compounds on 4 data points!

## Not run:

    rank_out <- rankEN(msObj = ms,
                       bioact = bioact$ec,
                       region_ms = paste0("_", 21:24),
                       region_bio = paste0("_", 21:24),
                       lambda = 0.001,
```

```
                    pos_only = TRUE,
                    ncomp = NULL)

    # print, summary function
    rank_out
    summary(rank_out)

    # Extract ranked compounds as a data.frame
    ranked_candidates <- extract_ranked(rank_out)


## End(Not run)
```

---

summary.binMS *Overview of the binning process*

Description

Prints a text description of the binning process. Displays arguments passed to the binMS routine, how many m/z levels were chosen for each criterion, how many candidate compounds were chosen overall, and how many candidate compounds were obtained after consolidation.

Usage

```
## S3 method for class 'binMS'
summary(object, ...)
```

Arguments

object	An object of class binMS
...	Arguments passed to dot-dot-dot are ignored

---

summary.filterMS *Overview of the filtering process*

Description

Prints a description of the filtering process. Displays arguments chosen for the filterMS constructor, how many candidate compounds were chosen for each criterion, and how many candidate compounds were chosen overall.

Usage

```
## S3 method for class 'filterMS'
summary(object, ...)
```

Arguments

object	An object of class filterMS
...	Arguments passed to dot-dot-dot are ignored

---

summary.rankEN *Overview of the elastic net selection process*

Description

Prints a description of the elastic net variable selection process. Includes the dimensions used to fit the elastic net model, the fraction names for the mass spectrometry and the bioactivity data in the region of interest, the parameter specifications for the model, and a table with the identifying information of the candidate compounds produced by the model fit.

Usage

```
## S3 method for class 'rankEN'
summary(object, max_comp_print = 20L, ...)
```

Arguments

object	An object of class rankEN.
max_comp_print	A numeric value >= 1 specifying the maximum number of compounds to print
...	Arguments passed to dot-dot-dot are ignored

Index