Package 'IPO'

July 10, 2025

Type Package

Title Automated Optimization of XCMS Data Processing parameters

Version 1.34.0

Date 2022-08-14

Author Gunnar Libiseller <Gunnar.Libiseller@joanneum.at>, Christoph Magnes <christoph.magnes@joanneum.at>, Thomas Lieb <thomas.r.lieb@gmail.com>

Maintainer Thomas Lieb <thomas.r.lieb@gmail.com>

Depends xcms (>= 1.50.0), rsm, CAMERA, grDevices, graphics, stats, utils

Imports BiocParallel

Suggests RUnit, BiocGenerics, msdata, mtbls2, faahKO, knitr

Enhances parallel

VignetteBuilder knitr

Description The outcome of XCMS data processing strongly depends on the parameter settings. IPO (`Isotopologue Parameter Optimization`) is a parameter optimization tool that is applicable for different kinds of samples and liquid chromatography coupled to high resolution mass spectrometry devices, fast and free of labeling steps. IPO uses natural, stable 13C isotopes to calculate a peak picking score. Retention time correction is optimized by minimizing the relative retention time differences within features and grouping parameters are optimized by maximizing the number of features showing exactly one peak from each injection of a pooled sample. The different parameter settings are achieved by design of experiment. The resulting scores are evaluated using response surface models.

License GPL (>= 2) + file LICENSE

URL https://github.com/rietho/IPO

BugReports https://github.com/rietho/IPO/issues/new

biocViews ImmunoOncology, Metabolomics, MassSpectrometry

RoxygenNote 6.0.1

NeedsCompilation no

git_url https://git.bioconductor.org/packages/IPO

git_branch RELEASE_3_21

git_last_commit 6480443

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-07-09

Contents

IPO-package *Automated Optimization of Untargeted Metabolomics LC-MS Data Processing*

Description

IPO provides a framework for parameter optimization for the software package XCMS. It provides optimisation of peak picking parameters by using natural, stable 13C isotopes. Retention time correction is optimized by minimizing the relative retention time differences within features and grouping parameters are optimized by maximizing the number of features showing exactly one peak from each injection of a pooled sample.

Details

An overview of how to use the package, including the most important functions

Author(s)

Gunnar Libiseller

Maintainer: Thomas Riebenbauer <Thomas.Riebenbauer@joanneum.at>

References

Lenth, R. V. (2009). Response-Surface Methods in R , Using rsm. Journal of Statistical Software, 32(7), 1-17. Retrieved from http://www.jstatsoft.org/v32/i07

Smith, C.A. and Want, E.J. and O'Maille, G. and Abagyan,R. and Siuzdak, G.: XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching and identification, Analytical Chemistry, 78:779-787 (2006)

Ralf Tautenhahn, Christoph Boettcher, Steffen Neumann: Highly sensitive feature detection for high resolution LC/MS BMC Bioinformatics, 9:504 (2008)

H. Paul Benton, Elizabeth J. Want and Timothy M. D. Ebbels Correction of mass calibration gaps in liquid chromatography-mass spectrometry metabolomics data Bioinformatics, 26:2488 (2010)

Yu, H. (2002). Rmpi: Parallel Statistical Computing in R. R News, 2(2), 10-14. Retrieved from http://cran.r-project.org/doc/Rnews/Rnews_2002-2.pdf

See Also

xcms

Examples

```
## Not run:
mtbls2files <- list.files(file.path(find.package("mtbls2"), "mzML"),
                          full.names=TRUE)
```

```
paramsPP <- getDefaultXcmsSetStartingParams()
paramsPP$mzdiff <- -0.001
#paramsPP$ppm <- 25
paramsPP$min_peakwidth <- c(7,14)
paramsPP$max_peakwidth <- c(20,30)
paramsPP$noise <- 10000
resultPP <- optimizeXcmsSet(mtbls2files[1:2], paramsPP, subdir="mtbls2")


paramsRG <- getDefaultRetGroupStartingParams()
paramsRG$gapInit <- 0.2
paramsRG$profStep <- 1
paramsRG$minfrac <- 0.75
resultRG <- optimizeRetGroup(resultPP$best_settings$xset, paramsRG, nSlaves=2)

writeRScript(resultPP$best_settings$parameters, resultRG$best_settings,
             subdir="mtbls2", 4)

## End(Not run)
```

`attachList` *Attaching one list at the end of another*

Description

This function attaches one list at the end of another list.

Usage

```
attachList(params_1, params_2)
```

Arguments

`params_1`	A List
`params_2`	A second list which will be attached at the end of the first list.

Details

This is a convenience funktion, but the implementation is not optimized for speed.

Value

A List composed of the two input lists.

Author(s)

Gunnar Libiseller

Examples

```
a <- list("a"=1, "b"=2)
b <- list("c"=4, "d"=4)
attachList(a, b)
```

`calcPPS` *Calculation of a peak picking score (PPS) by using natural, stable 13C isotopes*

Description

This function calculates PPS by identifying natural, stable 13C isotopes within an xcmsSet object. Peaks beeing part of an isotopologue are defined as reliable peaks (RP). Peaks which are not part of an isotopologue and where the intensity of possible isotopes is below a cutoff are defined as 'low intensity peaks' (LIP). PPS is then calculated by:
PPS = RP^2/(#all_peaks - LIP)

Usage

```
calcPPS(xset, isotopeIdentification, ...)
```

Arguments

`xset`	`xcmsSet` object
`isotopeIdentification`	This parameter defines the method for isotope identification. The method IPO was especially implemented for high resolution data. CAMERA is an established isotope and adduct annotation package.
`...`	Additional parameters to CAMERA's findIsotopes function.

Details

Calculation of a peak picking score (PPS) by using natural, stable 13C isotopes

Value

An array with 5 items:
1. Space for experimentid of the Design of Experiments (0 since not known in calcPPS)
2. Number of peaks
3. Number of peaks without LIP and RP
4. Reliable peaks (RP)
5. Peak picking score (PPS)

Author(s)

Gunnar Libiseller

See Also

findIsotopes.IPO findIsotopes.CAMERA

Examples

```
mzmlfile <- file.path(find.package("msdata"), "microtofq/MM14.mzML")
xset <- xcmsSet(mzmlfile, peakwidth=c(5,12), method="centWave")
calcPPS(xset)
```

---

calculateXcmsSet *Calculation of an xcmsSet-Object*

Description

This function encapsulates xcms::findPeaks-methods for IPO

Usage

```
calculateXcmsSet(files, xcmsSetParameters, scanrange=NULL, task=1,
  BPPARAM = bpparam(), nSlaves=0)
```

Arguments

files	a vector containing the files for peak picking
xcmsSetParameters	a list with all parameters for xcmsSet-methods as list-items
scanrange	scan range to process. See findPeaks.centWave.
task	The task-id when using this method in parallel calculations.
BPPARAM	a BiocParallel parameter object to control how and if parallel processing of xcmsSet should be performed. Such objects can be created by the SerialParam, MulticoreParam or SnowParam functions.
nSlaves	xcmsSet's nSlaves-argument is deprecated., use BPPARAM argument instead.

Details

Encapsulation of xcms::findPeaks-methods used in IPO.

Value

An xcmsSet-object

Author(s)

Gunnar Libiseller, Thomas Riebenbauer (<thomas.riebenbauer@joanneum.at>)

References

Smith, C.A. and Want, E.J. and O'Maille, G. and Abagyan,R. and Siuzdak, G.: XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching and identification, Analytical Chemistry, 78:779-787 (2006)

Ralf Tautenhahn, Christoph Boettcher, Steffen Neumann: Highly sensitive feature detection for high resolution LC/MS BMC Bioinformatics, 9:504 (2008)

See Also

findPeaks

Examples

```
mzmlfile <- file.path(find.package("msdata"), "microtofq/MM14.mzML")

params <- list(min_peakwidth=5, max_peakwidth=12, ppm=58,
                  mzdiff=-0.001, snthresh=10, noise=0, prefilter=3,
                  value_of_prefilter=100,  mzCenterFun="wMean", integrate=1,
                  fitgauss=FALSE, verbose.columns=FALSE, nSlaves=1)

xset <- calculateXcmsSet(mzmlfile, params)
```

---

combineParams *Combining two lists of parameters together.*

Description

This function combines two lists of parameters. The first is a list of parameters which should be optimized. These parameters have different values set by Design of Experimnt. The second list consists of parameters which should not be optimized, hence only one value is set for each parameter. The parameters of the second list are replicated to have the same length as the number experiments in the DoE. Then the two lists are combined.

Usage

```
combineParams(params_1, params_2)
```

Arguments

params_1	A list holding parameters which should be optimized. Each parameter already has value set for each experiment of an Design of Experiment.
params_2	A list holding parameters which should not be optimized, hence only one value is set.

Details

Special treatment is needed for the findPeaks.matchedFilter-parameters 'sigma', 'mzdiff' since these two parameters are defined by default relative to the parameters 'fwhm' or 'step' and 'steps' respectively.
`sigma=fwhm/2.3548` `mzdiff=0.8-step*steps`

Value

A list of consting of all parameters needed for an xcms-method (findPeaks.centWave, findPeaks.matchedFilter, retcor.obiwarp or group.density). Each list item has the same length which is equal to the number of experiments within the DoE.

Author(s)

Gunnar Libiseller

Examples

```
params <- getDefaultXcmsSetStartingParams()
typ_params <- typeCastParams(params)
design <- getBbdParameter(typ_params$to_optimize)
xcms_design <- decode.data(design)
xcms_design <- combineParams(xcms_design, typ_params$no_optimization)
xcms_design
```

---

createModel *Creating a response surface model.*

Description

This function uses a design of experiments, a response for the experiments within the design and the used parameters to create a response surface model

Usage

```
createModel(design, params, resp)
```

Arguments

design	A design of experiments (Box-Behnken-Design or Central-Composite-Design)
params	The parameters which were used.
resp	The responses achivied for the various experiments.

Details

This function uses a design of experiments, a response for the experiments within the design and the used parameters to create a response surface model

Value

A response surface model.

Note

getBbdParameter getCcdParameter typeCastParams

Author(s)

Gunnar Libiseller

References

Lenth, R. V. (2009). Response-Surface Methods in R , Using rsm. Journal of Statistical Software, 32(7), 1-17. Retrieved from http://www.jstatsoft.org/v32/i07

Examples

```
params <- getDefaultXcmsSetStartingParams()
type_params <- typeCastParams(params)
design <- getBbdParameter(type_params$to_optimize)
resp <- runif(nrow(design),1,3)

model <- createModel(design, type_params$to_optimize, resp)
dev.new()
par(mfrow=c(3,2))
contour(model, ~ x1*x2*x3*x4, image=TRUE)
```

---

decode *En-/decodes values to/from ranges of -1 to 1.*

Description

Encode and decode values that are in a range of -1 to 1 into a specified range.

Usage

```
encode(value, bounds)
decode(value, bounds)
decodeAll(values, params)
```

Arguments

value	A value
values	A vector with values in the range [-1,1]
bounds	A vector of two values defining the lower and upper bound of a range.
params	A list where evere list-item consist of two values defining a lower and an upper bound.

Details

Decodes a values from ranges of -1 to 1 to ranges specified.

A function used to decode values that are in a range of -1 to 1 into a specified range. For every value a list item with lower and upper bound has to be supplied.

A function used to encode values that are in a specified range into a range between -1 to 1.

Value

decode: The encoded value. decodeAll: A vector of decoded values.

Author(s)

Gunnar Libiseller

Examples

```
decode(0, c(10, 20))
decode(-0.5, c(10, 20))
decode(1, c(10, 20))

bounds <- c(10, 20)
encode(decode(1, bounds), bounds)

       ## Multiple values:
values <- c(-1, -0.25, 0, 0.75)
params <- getDefaultXcmsSetStartingParams()
type_params <- typeCastParams(params)

decodeAll(values, type_params$to_optimize)

       ## Combination of encode and decode
encode(15, c(10, 20))
encode(10, c(10, 20))
encode(5, c(1, 5))

bounds <- c(1,5)
decode(encode(5, bounds), bounds)
```

`findIsotopes.CAMERA` *Identification of Isotopes using the package CAMERA.*

Description

This function finds isotopes using CAMERA's find peak function. Isotopes are separately found within each sample.

Usage

```
findIsotopes.CAMERA(xset, ...)
```

Arguments

`xset`	`xcmsSet` object
`...`	Additional parameters to the findIsotopes function of CAMERA

Details

Identification of 13C isotopes

Value

An matrix with 2 columns. Column one shows the peak id of the 12C, peak column two shows the id of the respective 13C isotope peak.

Author(s)

Gunnar Libiseller

References

C. Kuhl and R. Tautenhahn and C. Boettcher and T. R. Larson and S. Neumann: CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets Analytical Chemistry 84:283 (2012)

See Also

`findIsotopes.IPO`

Examples

```
mzmlfile <- file.path(find.package("msdata"), "microtofq/MM14.mzML")
xset <- xcmsSet(mzmlfile, peakwidth=c(5,12), method="centWave")
isotopes <- findIsotopes.CAMERA(xset, ppm=15, maxcharge=1)
```

`findIsotopes.IPO` *Identification of 13C isotopes*

Description

This function identifies natural, stable 13C isotopes within an xcmsSet object of LC-HRMS data. Isotopes have to be within a mass-, retentiontime- and intensitywindow to be recognized as isotopes. If checkBorderIntensity is TRUE the maximum intensity of each peaks has to be at least three times the intensity at rtmin and rtmax.

Usage

```
findIsotopes.IPO(xset, checkPeakShape=c("none", "borderIntensity",
                                        "sinusCurve", "normalDistr"))
```

Arguments

`xset`	`xcmsSet` object
`checkPeakShape`	character to choose if the peakshape should be checked and if so how

Details

Identification of 13C isotopes

Value

An matrix with 2 columns. Column one shows the peak id of the 12C, peak column two shows the id of the respective 13C isotope peak.

Author(s)

Gunnar Libiseller

See Also

`findIsotopes.CAMERA`

Examples

```
mzmlfile <- file.path(find.package("msdata"), "microtofq/MM14.mzML")
xset <- xcmsSet(mzmlfile, peakwidth=c(5,12), method="centWave")
isotopes <- findIsotopes.IPO(xset, "borderIntensity")
```

getBbdParameter *Creates a Box-Behnken Design of experiment*

Description

Creates a Box-Behnken Design of Experiment out of a list of parameters. Each of the list items has to be a pair defining the lower und upper limits of the value-range to test. The method then returns a Center faced Box-Behnken Design of Experiments. The list has to hold a least three pairs.

Usage

```
getBbdParameter(params)
```

Arguments

params	A list of value pairs defining lower und upper limits of an optimization range.

Details

Creates a Box-Behnken Design of Experiment out of a list of parameters. Each of the list items has to be a pair defining the lower und upper limits of the value-range to test. The method then returns a Center faced Box-Behnken Design of Experiments. The list has to hold a least three pairs.

Value

A Box-Behnken Design of Experiments

Author(s)

Gunnar Libiseller

References

Lenth, R. V. (2009). Response-Surface Methods in R , Using rsm. Journal of Statistical Software, 32(7), 1-17. Retrieved from http://www.jstatsoft.org/v32/i07

See Also

getCcdParameter

Examples

```
params <- getDefaultXcmsSetStartingParams()
typ_params <- typeCastParams(params)
design <- getBbdParameter(typ_params$to_optimize)
```

getCcdParameter *Creates a Central-Composite Design of experiment*

Description

Creates a Central-Composite Design of Experiment out of a list of parameters. Each of the list items has to be a pair defining the lower und upper limits of the value-range to test. The method then returns a Center faced Central-Composite Design of Experiments.

Usage

```
getCcdParameter(params)
```

Arguments

params	A list of value pairs defining lower und upper limits of an optimization range.

Details

Creates a Central-Composite Design of Experiment out of a list of parameters. Each of the list items has to be a pair defining the lower und upper limits of the value-range to test. The method then returns a Center faced Central-Composite Design of Experiments.

Value

A Central-Composite Design of Experiments

Author(s)

Gunnar Libiseller

References

Lenth, R. V. (2009). Response-Surface Methods in R , Using rsm. Journal of Statistical Software, 32(7), 1-17. Retrieved from http://www.jstatsoft.org/v32/i07

See Also

getBbdParameter

Examples

```
params <- getDefaultXcmsSetStartingParams()
typ_params <- typeCastParams(params)
design <- getCcdParameter(typ_params$to_optimize)
```

getDefaultRetCorCenterSample *Gets the index of the sample with most peaks in it.*

Description

Gets the index of the sample with most peaks in it. This is used if no center sample for retention time correction has been defined by the user.

Usage

```
getDefaultRetCorCenterSample(xset)
```

Arguments

xset	xcmsSet object

Details

Gets the index of the sample with most peaks in it. This is used if no center sample for retention time correction has been defined by the user.

Value

The file index of the sample with most peaks in it.

Author(s)

Gunnar Libiseller

Examples

```
## The function is currently defined as
function (xset)
{
    ret <- NULL
    for (i in 1:length(filepaths(xset))) {
        ret <- c(ret, sum(peaks(xset)[, "sample"] == i))
    }
    return(which.max(ret))
  }
```

`getDefaultRetGroupStartingParams` *Gives a List of parameters for xcms-methods retcor.obiwarp or retcor.loess and group.density which are optimized by default*

Description

This function creates a list of parameters used in the xcms-methods retcor.obiwarp and group.density. Per default the following parameters have a defined range where optimization should start:
retcor.obiwarp parameters: 'gapInit'; 'gapExtend', 'profStep'
group.density parameters: 'bw', 'minfrac', 'mzwid'

Usage

```
getDefaultRetGroupStartingParams(retcorMethod=c("obiwarp", "loess", "none"),
distfunc=c("cor_opt", "cor", "cov", "prd", "euc"), high_resolution=TRUE)
```

Arguments

`retcorMethod`	The name of the retention time correction method that should be used. The XCMS methods retcor.obiwarp and retcor.loess are supported. If no retention time correction should be done use "none".
`distfunc`	The name of the distance function used by retcor.obiwarp
`high_resolution`	If high_resolution = TRUE starting values for mzwid are set to 0.015 and 0.035; if high_resolution = FALSE to 0.15, 0.35

Details

* Do not delete a parameter from the list returned.
* Optimization of qualitative parameters is not supported yet.
* If you want to optimize additional parameter just set an lower and an upper bound (e.g. params$max <- c(4,8))
* If you dont want to optimize a parameter set a default value (e.g. params$max <- 10)

Value

A List of parameters used in the xcms-methods retcor.obiwarp or retcor.loess and group.density

Author(s)

Gunnar Libiseller

Examples

```
params <- getDefaultRetGroupStartingParams()
params$bw <- 10
params$max <- c(4,8)
params
```

`getDefaultXcmsSetStartingParams` *Creates a List of parameters for xcms-methods xcmsSet.findPeak which are optimized by default*

Description

This function creates a list of parameters used in the xcmsSet.findPeak-methods 'centWave' and 'matchedFilter'. Per default the following parameters have a defined range where optimization should start:
'centWave' parameters: 'peakwidth' (split into 'min_peakwidth' and 'max_peakwidth'), 'ppm', 'mzdiff'
'matchedFilter' parameters: 'fwhm', 'snthresh', 'step', 'steps'

Usage

```
getDefaultXcmsSetStartingParams(method = c("centWave", "matchedFilter"))
```

Arguments

`method`	Either parameters for 'centWave' or 'matchedFilter' should be created

Details

* Do not delete a parameter from the list returned.
* Optimization of qualitative parameters is not supported yet.
* If you want to optimize additional parameter just set an lower and an upper bound (e.g. params$snthresh <- c(5,20))
* If you dont want to optimize a parameter set a default value (e.g. params$snthresh <- 10)

Value

A List of parameters for the xcmsSet.findPeak-methods 'centWave' or 'matchedFilter'

Author(s)

Gunnar Libiseller

Examples

```
params <- getDefaultXcmsSetStartingParams()
params$ppm <- 10
params$snthresh <- c(5,15)
params

params <- getDefaultXcmsSetStartingParams("matchedFilter")
params
```

`getNormalizedResponse` *It combines Retention time Correction Scores (RCS) and Grouping Scores (GS)*

Description

This function does unity based normalization on Retention time Correction Scores (RCS) as well as Grouping Scores (GS).

Usage

```
getNormalizedResponse(response)
```

Arguments

`response`	A List of all responses calculated by getRGTVValues for all experiments of an Design of Experiment

Details

Grouping Score (GS) is calculated by:
'good groups'^2/'bad groups

For all RCS and GS values unitiy based normalization is done. For every experiment within the DoE these two values are added together and returned.

Value

A vector with RTGV values

Note

Since RCS and GS can be within completely different ranges, normalization has to be done to prevent an excessive influence of either RCS or GS.

Author(s)

Gunnar Libiseller

See Also

getRGTVValues

Examples

```
mtbls2files <- list.files(file.path(find.package("mtbls2"), "mzML"),
                          full.names=TRUE)

params <- list(min_peakwidth=12, max_peakwidth=30, ppm=30,
                  mzdiff=-0.001, snthresh=10, noise=10000, prefilter=3,
                  value_of_prefilter=100,  mzCenterFun="wMean", integrate=1,
                  fitgauss=FALSE, verbose.columns=FALSE, nSlaves=2)

xset <- calculateXcmsSet(mtbls2files[1:2], params)
xset <- retcor(xset, method="obiwarp")
xset <- group(xset)

result <- getRGTVValues(xset)
result
```

---

getRGTVValues — *Calculation of values used for a Retention time correction and Grouping Target Value (RGTV)*

Description

This function calculates the Retention time Correction Score (RCS) of all features within an xcmsSet-object. Also features having exactly one peak from each sample are defined as 'good groups', all others a 'bad groups'.

Usage

```
getRGTVValues(xset, exp_index = 1, retcor_penalty = 1)
```

Arguments

xset	xcmsSet object
exp_index	Experiment-id of the experiment within a Design of Experiments
retcor_penalty	Penalty if an error occured with the used retention time correction parameters

Details

This function calculates the Retention time Correction Score (RCS) of all features within an xcmsSet-object. Also features having exactly one peak from each sample are defined as 'good groups', all others a 'bad groups' which leads to a Grouping Score (GS) by calculating 'good groups'^2/'bad groups'.

Value

a list containing the items exp_index, good_groups, bad_groups, GS and RCS.

Author(s)

Gunnar Libiseller

Examples

```
mtbls2files <- list.files(paste(find.package("mtbls2"), "/mzML", sep=""),
                          full.names=TRUE)
xset <- xcmsSet(mtbls2files[1:2], method="centWave", peakwidth=(c(12, 30)),
                ppm=30, noise=10000)
xset <- retcor(xset, method="obiwarp")
xset <- group(xset)
getRGTVValues(xset)
```

`optimizeRetGroup` *Optimization for parameters for retention time correction and grouping*

Description

This function provides optimisation for parameters of the xcms-method retcor.obiwarp and group.density. The retention time correction is optimised by minimizing intra-feature retention time shifts; grouping is optimized by increasing the number of features which have exactly one peak per sample.

Usage

```
optimizeRetGroup(xset, params = getDefaultRetGroupStartingParams(),
                 nSlaves = 4, subdir = "IPO", plot = TRUE)
```

Arguments

`xset`	`xcmsSet` object
`params`	A list of parameters which are needed by xcms-methods retcor.obiwarp and group.density. List-items with two values will be optimized. The first value defines the lower test value, the second one the higher test value.
`nSlaves`	Number of slaves the optimization process should spawn.

subdir	The name of the subdirectory which is created and where the figures of the response surface models will be saved to. NULL plots the figures to the graphic device. This parameter is ignored, if plot = TRUE.
plot	Defines if plots should be generated (TRUE) or not (FALSE). This parameter overwrites the subdir-parameter. Defaults to TRUE.

Details

This function provides optimisation for parameters of the xcms-method retcor.obiwarp and group.density. The retention time correction is optimised by minimizing intra-feature retention time shifts; grouping is optimized by increasing the number of features which have exactly one peak per sample.

Value

A LIST of length n+1 with n beeing the optimization runs needed

comp1-comp(n)	A LIST containing: 1. Parameters used for the nth optimization run 2. Box-Behnken or Central Composite Design used for optimization run 3. Responses from calculateRGTV for every experiment in the design 4. Response surface model for the design 5. The normalized parameter settings giving the best Retention time and Grouping Target Value (RGTV) (values between -1 and 1) 6. Response from calculateRGTV for xcmsSet-object created with best parameters in this run 7. xcmsSet-object created with best parameters in this run
comp(n+1)	A LIST containing: 1. Parameters giving the best RGTV

Author(s)

Gunnar Libiseller

References

Obiwarp Prince, J. T., & Marcotte, E. M. (2006). Chromatographic alignment of ESI-LC-MS proteomics data sets by ordered bijective interpolated warping. Analytical chemistry, 78(17), 6140-52. doi:10.1021/ac0605344

See Also

getDefaultRetGroupStartingParams

Examples

```
mtbls2files <- list.files(file.path(find.package("mtbls2"), "mzML"),
                          full.names=TRUE)
```

```
params <- list(min_peakwidth=12, max_peakwidth=30, ppm=30,
                  mzdiff=-0.001, snthresh=10, noise=10000, prefilter=3,
                  value_of_prefilter=100,  mzCenterFun="wMean", integrate=1,
                  fitgauss=FALSE, verbose.columns=FALSE, nSlaves=2)

xset <- calculateXcmsSet(mtbls2files[1:2], params)


#optimize the retention time correction and grouping parameters
paramsRG <- getDefaultRetGroupStartingParams()
paramsRG$profStep <- 1
paramsRG$minfrac <- 0.75
resultRG <- optimizeRetGroup(xset, params=paramsRG,
                             nSlaves=4,subdir="mtbls2")

writeRScript(params, resultRG$best_settings, 4)
```

`optimizeXcmsSet` *Optimisation of peak picking parameters by using natural, stable 13C isotopes*

Description

This function provides optimisation of peak picking parameters by using natural, stable 13C isotopes.

Usage

```
optimizeXcmsSet(files, params = getDefaultXcmsSetStartingParams(),
  isotopeIdentification = c("IPO", "CAMERA"), BPPARAM = bpparam(),
  nSlaves = 4, subdir = "IPO", plot = TRUE, ...)
```

Arguments

`files`	A directory or list of files, passed to `xcmsSet`. If no files are given, xcmsSet() will check recursively all MS files in the current working directory.
`params`	A list of parameters which are needed by XCMS::findPeaks-Methods. List-items with two values will be optimized. The first value defines the lower test value, the second one the higher test value.
`isotopeIdentification`	This parameter defines the method for isotope identification. The method 'IPO' was especially implemented for high resolution data. CAMERA is an established isotope and adduct annotation package.

BPPARAM	a BiocParallel parameter object to control how and if parallel processing of xcmsSet should be performed. Such objects can be created by the SerialParam, MulticoreParam or SnowParam functions. Note: xcmsSet's nSlaves-argument is deprecated.
nSlaves	Number of slaves the optimization process should spawn.
subdir	The name of the subdirectory which is created and where the figures of the response surface models will be saved to. NULL plots the figures to the graphic device. This parameter is ignored, if plot = TRUE.
plot	Defines if plots should be generated (TRUE) or not (FALSE). This parameter overwrites the subdir-parameter. Defaults to TRUE.
...	Additional parameters to CAMERA's or IPO's findIsotopes functions

Details

This function provides optimisation of peak picking parameters by using natural, stable 13C isotopes.

Value

A LIST of length n+1 with n beeing the optimization runs (DoEs) needed

comp1-comp(n)	A LIST containing: 1. Parameters used for the nth optimization run 2. Box-Behnken or Central Composite Design used for optimization run 3. Responses from calcPPS for every experiment in the design 4. Response surface model for the design 5. The normalized parameter settings giving the best PPS (values between -1 and 1) 6. an xcmsSet-class-object calculated with the best settings from the response surface model 7. PPS calculated from the xcmsSet
comp(n+1)	A LIST containing: 1. Parameters giving the best PPS 2. An xcmsSet-object created with the optimized parameters 3. The result of calcPPS() of the xcmsSet created with the best parameters

Author(s)

Gunnar Libiseller, Thomas Riebenbauer (<thomas.riebenbauer@joanneum.at>)

References

Smith, C.A. and Want, E.J. and O'Maille, G. and Abagyan,R. and Siuzdak, G.: XCMS: Processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching and identification, Analytical Chemistry, 78:779-787 (2006)

Ralf Tautenhahn, Christoph Boettcher, Steffen Neumann: Highly sensitive feature detection for high resolution LC/MS BMC Bioinformatics, 9:504 (2008)

H. Paul Benton, Elizabeth J. Want and Timothy M. D. Ebbels: Correction of mass calibration gaps in liquid chromatography-mass spectrometry metabolomics data Bioinformatics, 26:2488 (2010)

C. Kuhl and R. Tautenhahn and C. Boettcher and T. R. Larson and S. Neumann: CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets Analytical Chemistry 84:283 (2012)

See Also

getDefaultXcmsSetStartingParams, calcPPS, findIsotopes.IPO, findIsotopes.CAMERA

Examples

```
#library(IPO)

mzmlfile <- file.path(find.package("msdata"), "microtofq/MM14.mzML")

paramsPP <- getDefaultXcmsSetStartingParams()
paramsPP$mzdiff <- -0.001
paramsPP$min_peakwidth <- c(7,14)
paramsPP$max_peakwidth <- c(20,30)
#example using IPO isotope identification
resultPP <- optimizeXcmsSet(mzmlfile, paramsPP, subdir="mtbls2")

#example using CAMERA isotope identification
resultPP <- optimizeXcmsSet(mzmlfile, paramsPP, isotopeIdentification="CAMERA",
                            subdir="mtbls2", ppm=15, maxcharge=2)
```

---

toMatrix *Converts an array into a matrix*

Description

This function converts an array into a matrix. This is useful to counter the implicit casting of matrices into arrays when only one row is selected. If a matrix is passed to the function, the matrix is returned, if an array is passed, a matrix with one row is returned.

Usage

```
toMatrix(data)
```

Arguments

data	An array or a matrix

Value

A matrix

Author(s)

Gunnar Libsieller

Examples

```
data <- matrix(1:9, nrow=3)
colnames(data) <- c("a","b","c")
x <- data[1,]
is.matrix(x)
x <- toMatrix(x)
is.matrix(x)
```

`typeCastParams` *Splits parameters into those which should be optimized and those which are fixed.*

Description

This method takes a list of parameters and returns a list consisting of another two lists; one holding parameters ment for optimization and one holding fixed parameters.

Usage

```
typeCastParams(params)
```

Arguments

`params`	A list of parameters for an xcms-method

Details

This method takes a list of parameters and returns a list consisting of another two lists; one holding parameters ment for optimization and one holding fixed parameters.

Value

A list of:

`to_optimize`	A LIST containing all parameters which should be optimized.
`no_optimization`	A LIST containing all parameters which should not be optimized.

Author(s)

Gunnar Libiseller

See Also

`optimizeXcmsSet`, `optimizeRetGroup`

Examples

```
params <- getDefaultXcmsSetStartingParams()
typ_params <- typeCastParams(params)
```

`writeParamsTable` *Writes XCMS settings to a file.*

Description

This function writes findPeaks, retcor and grouping parameters to a file using write.table.

Usage

```
writeParamsTable(peakPickingSettings, retCorGroupSettings, file, ...)
```

Arguments

`peakPickingSettings`	A list of optimized settings for xcms-methods findPeaks.centWave or findPeaks.matchedFilter
`retCorGroupSettings`	A list of optimized settings for xcms-methods for retcor.obiwarp and group.density
`file`	The name of the outputfile for the parameters.
`...`	Additional parameters for write.table.

Details

This function writes findPeaks, retcor and grouping parameters to a file using write.table.

Value

none

Author(s)

Gunnar Libiseller

See Also

`xcms`

Examples

```
#creating list of peak picking parameters
paramsPP <- list(min_peakwidth=5, max_peakwidth=12, ppm=58,
                 mzdiff=-0.001, snthresh=10, noise=0, prefilter=3,
                 value_of_prefilter=100,  mzCenterFun="wMean", integrate=1,
                 fitgauss=FALSE, verbose.columns=FALSE, nSlaves=1)


#creating list of retention time correction and grouping parameters
paramsRTCGroup <- list(retcorMethod="obiwarp", distFunc="cor", gapInit=0.2,
                       gapExtend=2.4, profStep=1, plottype="none", response=1,
                       factorDiag=2, factorGap=1, localAlignment=0, initPenalty=0,
                       bw=30, minfrac=0.5, minsamp=1, mzwid=0.25, max=50)

#writing parameters to the file "params.tsv"
writeParamsTable(paramsPP, paramsRTCGroup, "params.tsv")
```

`writeRScript` *Prints an R-script to the screen which can be used for xcms processing*

Description

This function prints a script of the optimized findPeaks, retcor and grouping parameters to the screen.

Usage

```
writeRScript(peakPickingSettings, retCorGroupSettings, nSlaves = 0)
```

Arguments

`peakPickingSettings`	The optimized settings for xcms-methods findPeaks.centWave or findPeaks.matchedFilter
`retCorGroupSettings`	The optimized settings for xcms-methods for retcor.obiwarp and group.density
`nSlaves`	DEPRECATED

Details

This function prints a script out of the optimized findPeaks, retcor and grouping parameters to the screen.

The function `message` is used to print the script. For capuring the output `capture.output(writeRScript(...), type = "message")` might be used.

Value

none

Author(s)

Gunnar Libiseller, Thomas Riebenbauer (<thomas.riebenbauer@joanneum.at>)

Examples

```
#creating list of peak picking parameters
paramsPP <- list(min_peakwidth=5, max_peakwidth=12, ppm=58,
                 mzdiff=-0.001, snthresh=10, noise=0, prefilter=3,
                 value_of_prefilter=100,  mzCenterFun="wMean", integrate=1,
                 fitgauss=FALSE, verbose.columns=FALSE, nSlaves=1)


#creating list of retention time correction and grouping parameters
paramsRTCGroup <- list(retcorMethod="obiwarp", distFunc="cor", gapInit=0.2,
                       gapExtend=2.4, profStep=1, plottype="none", response=1,
                       factorDiag=2, factorGap=1, localAlignment=0, initPenalty=0,
                       bw=30, minfrac=0.5, minsamp=1, mzwid=0.25, max=50)

#outputting an xcms-script to the display
writeRScript(paramsPP, paramsRTCGroup, 4)
```

Index