Package 'cosmiq'

April 8, 2026

Type Package

Title cosmiq - COmbining Single Masses Into Quantities

Version 1.45.0

Depends R (>= 3.6), Rcpp

Imports pracma, xcms, MassSpecWavelet, faahKO

Suggests RUnit, BiocGenerics, BiocStyle

Description cosmiq is a tool for the preprocessing of liquid- or gas - chromatography mass spectrometry (LCMS/GCMS) data with a focus on metabolomics or lipidomics applications. To improve the detection of low abundant signals, cosmiq generates master maps of the mZ/RT space from all acquired runs before a peak detection algorithm is applied. The result is a more robust identification and quantification of low-intensity MS signals compared to conventional approaches where peak picking is performed in each LCMS/GCMS file separately. The cosmiq package builds on the xcmsSet object structure and can be therefore integrated well with the package xcms as an alternative preprocessing step.

License GPL-3

URL http://www.bioconductor.org/packages/devel/bioc/html/cosmiq.html

Collate combine_spectra.R peakdetection.R eicmatrix.R retention_time.R quantify_combined.R create_datamatrix.R cosmiq.R

LazyData false

biocViews ImmunoOncology, MassSpectrometry, Metabolomics

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

git_branch devel

git_last_commit 5f8e85a

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-04-08

Author David Fischer [aut, cre],
Christian Panse [aut] (ORCID: <https://orcid.org/0000-0003-1975-3064>),
Endre Laczko [ctb]

Maintainer David Fischer <dajofischer@googlemail.com>

Contents

combine_spectra 2
cosmiq 3
create_datamatrix 5
eicmatrix 8
peakdetection 9
quantify_combined 10
retention_time 11

Index **13**

`combine_spectra` *Combine mass spectra of each scan and each file into a single master spectrum*

Description

`combine_spectra` imports each raw file using `xcmsRaw` and assigns each ion to a previously defined mass vector, which is created using bin size parameter `mzbin`. This process is repeated for each raw file.

Usage

```
combine_spectra(xs, mzbin=0.003, linear=FALSE, continuum=FALSE)
```

Arguments

`xs`	xcmsSet object
`mzbin`	Bin size for the generation of mass vector
`linear`	logical. If TRUE, linear vector will be generated with `mzbin` increments. If FALSE, mass vector will be generated using a non-linear function. This option is recommended for TOF-type mass detectors
`continuum`	boolean flag. default value is FALSE.

Details

This processing step calculates a combined mass spectrum. Mass spectra not only from all scans of a single LCMS run alone are combined but from all acquired datasets. As a result, signal to noise ratio increases for each additional LCMS run.

Author(s)

David Fischer 2013

Examples

```
cdfpath <- file.path(find.package("faahKO"), "cdf")

my.input.files <- dir(c(paste(cdfpath, "WT", sep='/'),
    paste(cdfpath, "KO", sep='/')), full.names=TRUE)


# create xcmsSet object
xs <- new("xcmsSet")
xs@filepaths <- my.input.files

x<-combine_spectra(xs=xs, mzbin=0.25,
    linear=TRUE, continuum=FALSE)

plot(x$mz, x$intensity, type='l',
    xlab='m/Z', ylab='ion intensity')
```

`cosmiq` *cosmiq - main wrapper function*

Description

This is the main wrapper function for the package cosmiq. Every processing step of `cosmiq` will be calculated during this function, including mass spectra combination, detection of relevant masses, generation and combination of extracted ion chromatograms, detection of chromatographic peaks, Localisation and quantification of detected peaks.

`combine_spectra` imports each raw file using `xcmsRaw` and assigns each ion to a previously defined mass vector, which is created using bin size parameter `mzbin`. This process is repeated for each raw file.

Usage

```
cosmiq (files, RTinfo=TRUE, mzbin=0.003,
    center=0, linear=FALSE, profStep=1, retcorrect=TRUE,
    continuum=FALSE, rtcombine=c(0,0), scales=c(1:10),
    SNR.Th=10, SNR.area=20, RTscales=c(1:10, seq(12,32,2)),
    RTSNR.Th=20, RTSNR.area=20, mintr=0.5)
```

Arguments

files	String vector containing exact location of each file
RTinfo	Logical indicating whether retention time should be used or not. If FALSE, the quantitative information will be retrieved as a sum of ion intensities whithin a retention time window. This retention time window
rtcombine	Numerical, with two entries. If no retention time information is to be used, rtcombine provides the retention time window where ion intensities will be summed for quantification. If rtcombine = c(0,0), All scans will be used
mzbin	Bin size for the generation of mass vector
center	Indicates number of file which will be used for retention time alignment. If zero, file will be automatically selected based on maximum summed ion intensity of the combined mass spectrum.
linear	logical. If TRUE, linear vector will be generated with mzbin increments. If FALSE, mass vector will be generated using a non-linear function. This option is recommended for TOF-type mass detectors
profStep	step size (in m/z) to use for profile generation from the raw data files.
retcorrect	Logical, should retention time correction be used? default = TRUE
continuum	Logical, is continuum data used? default = FALSE
scales	Peak Detection of Mass Peaks in the combined mass spectrum: scales for continuous wavelet transformation (CWT).
SNR.Th	Peak Detection of Mass Peaks in the combined mass spectrum: Signal to noise ratio threshold
SNR.area	Peak Detection of Mass Peaks in the combined mass spectrum: Area around the peak for noise determination. Indicates number of surrounding peaks on the first CWT scale. default = 20
RTscales	Peak Detection of Chromatographic peaks on the combined extracted ion chromatogram: scales for continuous wavelet transformation (CWT), see also the peakDetectionCWT function of the MassSpecWavelet package.
RTSNR.Th	Peak Detection of Chromatographic peaks on the combined extracted ion chromatogram: Signal to noise ratio threshold
RTSNR.area	Peak Detection of Chromatographic peaks on the combined extracted ion chromatogram: Area around the peak for noise determination. Indicates number of surrounding peaks on the first CWT scale. default = 20
mintr	Minimal peak width intensity treshold (in percentage), for which two overlapping peaks are considered as separated. The default calue is 0.5.

Details

Current data processing tools focus on a peak detection strategy where initially each LCMS run is treated independently from each other. Subsequent data processing for the alignment of different samples is then calculated on reduced peak tables. This function involves the merging of all LCMS datasets of a given experiment as a first step of raw data processing. The merged LCMS dataset contains an overlay and the sum of ion intensities from all LCMS runs. Peak detection is then

performed only on this merged dataset and quantification of the signals in each sample is guided by the peak location in the merged data.

See also `xcms::retcor.obiwarp` and `cosmiq::create_datamatrix` which executes each step of the cosmiq function.

For running examples please read the package vignette or `?cosmiq::create_datamatrix`.

Author(s)

Christan Panse, David Fischer 2014

Examples

```
## Not run:
# the following lines consume 2m17.877s
# on a Intel(R) Xeon(R) CPU X5650  @ 2.67GHz
library(faahKO)

cdfpath <- file.path(find.package("faahKO"), "cdf")
my.input.files <- dir(c(paste(cdfpath, "WT", sep='/'),
    paste(cdfpath, "KO", sep='/')), full.names=TRUE)

 x <- cosmiq(files=my.input.files, mzbin=0.25, SNR.Th=0, linear=TRUE)

 image(t(x$eicmatrix^(1/4)),
    col=rev(gray(1:20/20)),
    xlab='rt',
    ylab='m/z', axes=FALSE )

 head(xcms::peakTable(x$xcmsSet))

## End(Not run)
```

`create_datamatrix` *Quantifying mz/RT intensities using peak locations from master map*

Description

`create_datamatrix` identifies mz/RT peak boundaries in each raw file using the information from a master mass spectrum and master EIC. For each mz/RT location, the peak volume is calculated and stored in a report table.

Usage

```
create_datamatrix(xs,rxy)
```

Arguments

xs	xcmsSet object
rxy	matrix containing mz and RT boundaries for each detected peak

Details

With the information about their position in the combined datasets, each individual mz/RT feature is located in the raw data.

Author(s)

David Fischer 2013

Examples

```
## Not run:
cdfpath <- file.path(find.package("faahKO"), "cdf")
my.input.files <- dir(c(paste(cdfpath, "WT", sep='/'),
    paste(cdfpath, "KO", sep='/')), full.names=TRUE)

#
# create xcmsSet object
# todo
xs <- new("xcmsSet")
# consider only two files!!!
xs@filepaths <- my.input.files[1:2]

class<-as.data.frame(c(rep("KO",2),rep("WT", 0)))
rownames(class)<-basename(my.input.files[1:2])
xs@phenoData<-class

x<-combine_spectra(xs=xs, mzbin=0.25,
    linear=TRUE, continuum=FALSE)

plot(x$mz, x$intensity, type='l',
    xlab='m/Z', ylab='ion intensity')

xy <- peakdetection(x=x$mz, y=x$intensity, scales=1:10,
SNR.Th=0.0, SNR.area=20, mintr=0.5)

id.peakcenter<-xy[,4]

plot(x$mz, x$intensity, type='l',
    xlim=c(440,460),
        xlab='m/Z', ylab='ion intensity')

points(x$mz[id.peakcenter], x$intensity[id.peakcenter],
    col='red', type='h')


# create dummy object
```

```
    xs@peaks<-matrix(c(rep(1, length(my.input.files) * 6),
        1:length(my.input.files)), ncol=7)

    colnames(xs@peaks) <- c("mz", "mzmin", "mzmax", "rt",
        "rtmin", "rtmax", "sample")

    xs<-xcms::retcor(xs, method="obiwarp", profStep=1,
        distFunc="cor", center=1)



    eicmat<-eicmatrix(xs=xs, xy=xy, center=1)

    #  process a reduced mz range for a better package build performance
    (eicmat.mz.range<-range(which(475 < xy[,1] & xy[,1] < 485)))

    eicmat.filter <- eicmat[eicmat.mz.range[1]:eicmat.mz.range[2],]
    xy.filter <- xy[eicmat.mz.range[1]:eicmat.mz.range[2],]

    #
    # determine the new range and plot the mz versus RT map
    (rt.range <- range(as.double(colnames(eicmat.filter))))
    (mz.range<-range(as.double(row.names(eicmat.filter))))

    image(log(t(eicmat.filter))/log(2), col=rev(gray(1:20/20)),
        xlab='rt [in seconds]', ylab='m/z', axes=FALSE,
        main='overlay of 12 samples using faahKO')

    axis(1, seq(0,1, length=6), round(seq(rt.range[1], rt.range[2], length=6)))
    axis(2, seq(0,1, length=4), seq(mz.range[1], mz.range[2], length=4))

    #
    # determine the chromatographic peaks
    rxy<-retention_time(xs=xs, RTscales=c(1:10, seq(12,32, by=2)),
        xy=xy.filter,
        eicmatrix=eicmat.filter,
        RTSNR.Th=120, RTSNR.area=20)

    rxy.rt <- (rxy[,4] - rt.range[1])/diff(rt.range)
    rxy.mz <- (rxy[,1] - mz.range[1])/diff(mz.range)

    points(rxy.rt, rxy.mz, pch="X", lwd=2, col="red")


    xs<-create_datamatrix(xs=xs, rxy=rxy)

    peaktable <- xcms::peakTable(xs)

    head(peaktable)

## End(Not run)
```

`eicmatrix` *Generate matrix of combined extraced ion chromatograms (EICs)*

Description

for each selected mass window, `eicmatrix` calculates EICs of every raw file and combines them together. It is recommended to correct the retention time first using `retcor.obiwarp`

Usage

```
eicmatrix(xs, xy, center)
```

Arguments

`xs`	xcmsSet object
`xy`	table including mz location parameters
`center`	file number which is used as a template for retention time correction

Details

For each detected mass, an extracted ion chromatogram (EIC) is calculated. In order to determine the elution time for each detected mass, the EICs of every mass are combined between all acquired runs.

Make sure that xcms-retention time correction (centwave) was applied to the dataset. The output will be a matrix of EIC's

Author(s)

David Fischer 2013

Examples

```
  ## Not run:
#  see package vignette section
# 'Generation and combination of extracted ion chromatograms'
xs<-create_datamatrix(xs=xs, rxy=rxy)

## End(Not run)
```

peakdetection *An algorithm for the detection of peak locations and boundaries in mass spectra or ion chromatograms*

Description

peakdetection uses Continuous wavelet transformation (CWT) to determine optimal peak location. A modified algorithm of Du et al. (2006) is used to localize peak positions.

Usage

```
peakdetection(scales, y, x, SNR.Th, SNR.area, mintr)
```

Arguments

scales	vector with the scales to perform CWT
y	vector of ion intensities
x	vector of mz bins
SNR.Th	Signal-to-noise threshold
SNR.area	Window size for noise estimation
mintr	Minimal peak width intensity treshold (in percentage), for which two overlapping peaks are considered as separated. default is set to 0.5.

Details

A peak detection algorithm based on continuous wavelet transformation (CWT) is used for this step (modified from Du et al., 2006). Peak detection based on CWT has the advantage that a sliding scale of wavelets instead of a single filter function with fixed wavelength is used. This allows for a flexible and automatic approximation of the peak width. As a result it is possible to locate both narrow and broad peaks within a given dynamic range.

Author(s)

David Fischer 2013

References

Du, P., Kibbe, W. A., & Lin, S. M. (2006). Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. Bioinformatics, 22(17), 2059-2065. doi:10.1093/bioinformatics/btl355

Examples

```
cdfpath <- file.path(find.package("faahKO"), "cdf")

my.input.files <- dir(c(paste(cdfpath, "WT", sep='/'),
    paste(cdfpath, "KO", sep='/')), full.names=TRUE)


# create xcmsSet object
xs <- new("xcmsSet")
xs@filepaths <- my.input.files

op<-par(mfrow=c(3,1))

x<-combine_spectra(xs=xs, mzbin=0.25,
    linear=TRUE, continuum=FALSE)

plot(x$mz, x$intensity,
    type='h',
    main='original',
    xlab='m/Z', ylab='ion intensity')

xy <- peakdetection(x=x$mz, y=x$intensity,
    scales=1:10, SNR.Th=1.0, SNR.area=20, mintr=0.5)

id.peakcenter<-xy[,4]

plot(x$mz[id.peakcenter], x$intensity[id.peakcenter], type='h',
    main='filtered')


plot(x$mz, x$intensity, type='l',
    xlim=c(400,450),
    main='zoom',
    log='y',
    xlab='m/Z', ylab='ion intensity (log scale)')

points(x$mz[id.peakcenter], x$intensity[id.peakcenter],col='red', type='h')
```

`quantify_combined` *Generate report with combined ion intensities*

Description

`quantify_combined` calculates a data table of mz intensities from a sum of ions within a selected mz window and from all MS scans.

Usage

```
quantify_combined(xs,xy, rtcombine)
```

Arguments

xs	xcmsSet object
xy	table including mz location parameters
rtcombine	Numerical, with two entries. If no retention time information is to be used, rtcombine provides the retention time window where ion intensities will be summed for quantification. If rtcombine = c(0,0), All scans will be used

Details

This function is only used, when no retention time information should be used, for example with direct infusion experiments.

Author(s)

David Fischer 2013

Examples

```
  ## Not run:
#  see package vignette

## End(Not run)
```

---

retention_time *Detection of chromatographic peak locations from extracted ion chromatograms (EICs)*

Description

For each EIC in the EIC matrix, retention_time localizes chromatographic peaks using peakdetection.

Usage

```
    retention_time(xy, xs, eicmatrix, RTscales, RTSNR.Th,
RTSNR.area, mintr)
```

Arguments

xy	table including mz location parameters
xs	xcmsSet object
eicmatrix	Matrix containing extracted ion chromatograms (EICs)
RTscales	Peak Detection of Chromatographic peaks on the combined extracted ion chromatogram: scales for continuous wavelet transformation (CWT), see also the peakDetectionCWT function of the MassSpecWavelet package.

RTSNR.Th	Peak Detection of Chromatographic peaks on the combined extracted ion chromatogram: Signal to noise ratio threshold
RTSNR.area	Peak Detection of Chromatographic peaks on the combined extracted ion chromatogram: Area around the peak for noise determination. Indicates number of surrounding peaks on the first CWT scale. default = 20
mintr	Minimal peak width intensity treshold (in percentage), for which two overlapping peaks are considered as separated (default = 0.5)

Details

Based on the combined extracted ion chromatograms, there is another peak detection step to be performed. The same algorithm as described for the peak picking of mass signals (continuous wavelet transformation) is also used for peak picking in the retention time domain.

Author(s)

David Fischer 2013

Examples

```
      ## Not run:
    #  see package vignette

## End(Not run)
```

Index

combine_spectra, 2
cosmiq, 3
create_datamatrix, 5

eicmatrix, 8

peakdetection, 9

quantify_combined, 10

retention_time, 11