Package 'yamss'

November 12, 2026

Version 1.38.0

Title Tools for high-throughput metabolomics

Description Tools to analyze and visualize high-throughput metabolomics data aquired using chromatography-mass spectrometry. These tools preprocess data in a way that enables reliable and powerful differential analysis. At the core of these methods is a peak detection phase that pools information across all samples simultaneously. This is in contrast to other methods that detect peaks in a sample-by-sample basis.

Depends R (>= 4.3.0), methods, BiocGenerics (>= 0.15.3), SummarizedExperiment

Suggests BiocStyle, knitr, rmarkdown, digest, mtbls2, testthat

Imports IRanges, stats, S4Vectors, EBImage, Matrix, mzR, data.table, grDevices, limma

VignetteBuilder knitr

License Artistic-2.0

URL https://github.com/hansenlab/yamss

BugReports https://github.com/hansenlab/yamss/issues

biocViews MassSpectrometry, Metabolomics, PeakDetection, Software

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

git_branch RELEASE_3_23

git_last_commit f6f8c17

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Leslie Myint [cre, aut] (ORCID: <https://orcid.org/0000-0003-2478-0331>), Kasper Daniel Hansen [aut]

Maintainer Leslie Myint <leslie.myint@gmail.com>

Contents

bakedpi . . . 2
CMSproc-class . . . 3
CMSraw-class . . . 5
cmsRawExample . . . 6
CMSslice-class . . . 6
diffrep . . . 7
getEICS . . . 8
getTIC . . . 9
plotDensityRegion . . . 9
readMSdata . . . 10
slicepi . . . 11

Index **12**

`bakedpi` *Process raw data to compute density estimate.*

Description

The `bakedpi` method stands for bivariate approximate kernel density estimation for peak identification. It performs background correction, retention time correction, and bivariate kernel density estimation.

Usage

```
bakedpi(cmsRaw, dbandwidth = c(0.005, 10), dgridstep = c(0.005, 1),
  outfileDens = NULL, dortalign = FALSE, mzsubset = NULL, verbose = TRUE)
```

Arguments

`cmsRaw`	An object of class `CMSraw`.
`dbandwidth`	A length-2 vector indicating the kernel density bandwidth in the M/Z and retention time (scan) directions. Default: `c(0.005,10)`
`dgridstep`	A length-2 vector indicating the grid step sizes. Default: `c(0.005,1)`.
`outfileDens`	Name of a file to save density estimate. If NULL, no output is saved.
`dortalign`	A logical value. Should retention time correction be performed?
`mzsubset`	A length-2 vector indicating a subset of the M/Z range to process. `NULL` otherwise.
`verbose`	Should the function be verbose?

Details

bakedpi first performs region-specific background correction. An optional retention time correction step follows in which M/Z region-specific shifts are computed to align the raw data. Next the two-dimensional density estimate is computed. The purpose of this function is to take the raw data read in by readMSdata and perform the steps necessary for bivariate kernel density estimation. The output of this function is used by slicepi to detect peaks and provide peak quantifications.

Value

An object of class CMSproc containing background corrected intensities, the bivariate kernel density estimate, and quantiles of the nonzero values in the density estimate.

Examples

```
## A very small dataset
data(cmsRawExample)
cmsProc1 <- bakedpi(cmsRawExample,
                    dbandwidth = c(0.01, 10), dgridstep = c(0.01, 1),
                    dortalign = TRUE, mzsubset = c(500,510))

## A longer example which takes a few minutes to run.
## This is still a smaller mz-slice of the full data.

if (require(mtbls2)) {

data(mtbls2)
filepath <- file.path(find.package("mtbls2"), "mzML")
files <- list.files(filepath, pattern = "MSpos-Ex1", recursive = TRUE, full.names = TRUE)
colData <- DataFrame(sampClasses = rep(c("wild-type", "mutant"), each = 4))
cmsRaw <- readMSdata(files = files, colData = colData, verbose = TRUE)
cmsProc2 <- bakedpi(cmsRaw, dbandwidth = c(0.01, 10), dgridstep = c(0.01, 1),
                   outfileDens = NULL, dortalign = TRUE, mzsubset = c(500, 520))

}
```

CMSproc-class *A class to hold chromatography-mass spectrometry preprocessing information.*

Description

This class builds on the CMSraw class to additionally store background-corrected intensities as well as the bivariate kernel density estimate.

Slots

`colData`: a `DataFrame` of phenotype and sample information.

`rawDT`: a `data.table` of raw spectral information.

`mzParams`: a list containing the minimum and maximum M/Z value and number of scans in each sample.

`rtAlign`: a logical indicating whether the data has been retention time aligned or not.

`bgcorrDT`: a `data.table` of background-corrected spectral information.

`density`: a matrix with rows corresponding to M/Z values and columns corresponding to scans containing the kernel density estimate.

`densityQuantiles`: a numeric vector containing the 100 percent quantiles of the nonzero density values.

Utility functions

We have the following utility functions:

`show`: The show method; prints the object.

`getEICS`: Gets extracted ion chromatograms (EICs) for the supplied M/Z ranges.

`plotDensityRegion`: Makes an image plot of the density estimate in a specified M/Z and scan region.

Accessors

We have the following accessor functions:

`colData`: Gets the `DataFrame` containing phenotype and sample information.

`densityEstimate`: Gets the matrix containing the density estimate.

`densityQuantiles`: Gets the quantiles of the nonzero values in the density estimate.

Examples

```
## Construct a completely fake example
densmat <- matrix(rnorm(600), nrow = 20, ncol = 30)
colnames(densmat) <- 1:ncol(densmat)
rownames(densmat) <- seq(350, by = 0.005, length.out = nrow(densmat))
cmsobj <- new("CMSproc", density = densmat)
head(densityEstimate(cmsobj))

## Takes about 20s to run


data(cmsRawExample)
cmsProc <- bakedpi(cmsRawExample,
                   dbandwidth = c(0.01, 10), dgridstep = c(0.01, 1),
                   dortalign = TRUE, mzsubset = c(500,510))
cmsProc
```

CMSraw-class *A class to hold chromatography-mass spectrometry raw data and metadata.*

Description

This class saves the data from raw mass spectral data files in a `data.table` and is used in subsequent processing. Objects of this class are created by the `readMSdata` function.

Slots

`colData`: a `DataFrame` of phenotype and sample information.

`rawDT`: a `data.table` of raw spectral information.

`mzParams`: a list containing the minimum and maximum M/Z value and number of scans in each sample.

Utility functions

We have the following utility functions:

`show`: The show method; prints the object.

`getEICS`: Gets extracted ion chromatograms (EICs) for the supplied M/Z ranges.

Accessors

We have the following accessor functions:

`colData`: Gets the `DataFrame` containing phenotype and sample information.

Examples

```
data(cmsRawExample)
cmsRawExample

if (require(mtbls2)) {
data(mtbls2)
filepath <- file.path(find.package("mtbls2"), "mzML")
files <- list.files(filepath, pattern = "MSpos-Ex1", recursive = TRUE, full.names = TRUE)[1]
colData <- DataFrame(group = "wild-type")
cmsRaw <- readMSdata(files = files, colData = colData, verbose = TRUE)
colData(cmsRaw)
}
```

cmsRawExample *An example cmsRaw object*

Description

This object contains parsed raw data for 4 samples in the MTBLS2 dataset.

Usage

```
cmsRawExample
```

Format

A `CMSraw` object containing information on 4 samples in the MTBLS2 dataset.

Value

An object of class `CMSraw` containing parsed data for 4 samples in the MTBLS2 dataset.

Source

The `mtbls2` Bioconductor data package.

CMSslice-class *A class to hold chromatography-mass spectrometry peak bounds and quantifications.*

Description

This class is based on the `SummarizedExperiment` class. It holds information on peak quantifications, M/Z and scan bounds, sample information, and preprocessing metadata. Objects of the class can be constructed using `CMSslice`.

Utility functions

We have the following utility functions:

`show`**:** The show method; prints the object.

Accessors

We have the following accessor functions:

`colData`**:** Gets the `DataFrame` containing phenotype and sample information.

`densityCutoff`**:** Gets the value used to threshold the density for peak calling.

`densityQuantiles`**:** Gets the quantiles of the nonzero values in the density estimate.

`peakBounds`**:** Gets the `DataFrame` of M/Z bounds, scan bounds, and ID numbers for detected peaks.

`peakQuants`**:** Gets the matrix of peak quantifications (rows: peaks, columns: samples).

Examples

```
## Construct a fake class
quants <- matrix(rnorm(12*5000), nrow = 5000, ncol = 12)
bounds <- cbind(mzmin = seq(from = 100, to = 1100, length.out = 5000),
                mzmax = seq(from = 100, to = 1100, length.out = 5000) + 0.1,
                scan.min = rep(10,5000), scan.max = rep(20, 5000),
                peaknum = 1:5000)
cmsobj <- CMSslice(assays = SimpleList(peakQuants = quants),
                   rowData = DataFrame(bounds))
head(peakQuants(cmsobj))

## A better example which takes 20s to run


data(cmsRawExample)
cmsProc <- bakedpi(cmsRawExample,
                   dbandwidth = c(0.01, 10), dgridstep = c(0.01, 1),
                   dortalign = TRUE, mzsubset = c(500,510))
cutoff <- tail(densityQuantiles(cmsProc), 2)[1]
sliced <- slicepi(cmsProc, cutoff = cutoff, verbose = TRUE)
sliced
```

`diffrep` *Perform differential analysis.*

Description

Performs differential abundance analysis on quantification information in a `CMSslice` object.

Usage

```
diffrep(cms, classes)
```

Arguments

`cms`	An object of class `CMSslice`.
`classes`	A character vector of class labels for the samples.

Details

Differential analysis is performed using the `limma` package which uses empirical Bayes methods in the estimation of feature-wise variances.

Value

A `data.frame` containing differential analysis information including log fold changes and p-values.

Examples

```
quantmat <- matrix(rnorm(12*5000), nrow = 5000, ncol = 12)
cmsobj <- CMSslice(assays = SimpleList(peakQuants = quantmat))
classes <- rep(c("case", "control"), each = 6)
difftab <- diffrep(cmsobj, classes)
```

`getEICS` *Compute extracted ion chromatograms for multiple regions.*

Description

Computes extracted ion chromatrograms (EICs) for the given M/Z ranges. Intensities are on the log2 scale.

Usage

```
getEICS(object, mzranges)
```

Arguments

`object`	An object of class `CMSraw` or `CMSproc`.
`mzranges`	A 2-column matrix where each row corresponds to one M/Z range and the first and second columns are the minimum and maximum M/Z values for the range respectively.

Details

In a given M/Z range, the maximum intensity observed in each scan gives the extracted ion chromatogram.

Value

A list with length equal to the number of rows of `mzranges` where each list element is a # scans by # samples matrix of EICs (on the log2 scale).

Examples

```
data(cmsRawExample)
mzranges <- rbind(c(500.01, 500.03), c(501.3, 501.5))
eicList <- getEICS(cmsRawExample, mzranges)
```

getTIC *Compute total ion chromatogram (TIC) for a sample.*

Description

Computes total ion chromatogram (TIC) for a single sample. Intensities are on the log2 scale. This requires a CMSraw object, typically produced from readMSdata.

Usage

```
getTIC(object, sample)
```

Arguments

object	An object of class CMSraw.
sample	An integer - for which sample should the TIC be computed?.

Value

A vector with length equal to the number of scans containing the log2 sum of intensities at each scan.

Examples

```
data(cmsRawExample)
tic <- getTIC(cmsRawExample, sample = 1)
```

plotDensityRegion *Image plot of region of density estimate.*

Description

Makes an image plot of the density estimate in the specified M/Z and scan region.

Usage

```
plotDensityRegion(cms, mzrange, scanrange)
```

Arguments

cms	An object of class CMSproc.
mzrange	A length-2 vector indicating the M/Z range to plot.
scanrange	A length-2 vector indicating the scan range to plot.

Value

This function is invoked for its side effect of plotting.

Examples

```
## For illustration purposes, we make a "dummy" object
## with a random matrix as the density estimate

densmat <- matrix(rnorm(600), nrow = 20, ncol = 30)
colnames(densmat) <- 1:ncol(densmat)
rownames(densmat) <- seq(350, by = 0.005, length.out = nrow(densmat))
densityQuantiles <- quantile(densmat, seq(from = 0, to = 1, by = 0.001))
cmsobj <- new("CMSproc", density = densmat, densityQuantiles = densityQuantiles)
plotDensityRegion(cmsobj, mzrange = c(350.01, 350.03), scanrange = c(10,20))
```

---

readMSdata *Read raw mass spectral data files.*

Description

Creates a `CMSraw` object that contains a `data.table` of raw mass spectral information for all samples. The resulting object also stores phenotype and sample information. This object is the basic encapsulation of essential raw experimental data and serves as the output for further processing methods.

Usage

```
readMSdata(files, colData, mzsubset, verbose)
```

Arguments

`files`	A character vector of filenames pointing to the raw data.
`colData`	A `DataFrame` of phenotype and sample information.
`mzsubset`	A length-2 vector indicating a subset of the M/Z range to process. `NULL` otherwise.
`verbose`	Should the function be verbose?

Value

A vector with length equal to the number of scans containing the log2 sum of intensities at each scan.

Examples

```
if (require(mtbls2)) {
data(mtbls2)
filepath <- file.path(find.package("mtbls2"), "mzML")
file <- list.files(filepath, pattern = "MSpos-Ex1",
                   recursive = TRUE, full.names = TRUE)[1]
colData <- DataFrame(group = "wild-type")
cmsRaw <- readMSdata(files = file, colData = colData, verbose = TRUE)
}
```

`slicepi` *Process raw data to compute density estimate.*

Description

The `slicepi` method uses the bivariate approximate kernel density estimate computed by `bakedpi` and uses a cutoff to bound and quantify peaks.

Usage

```
slicepi(object, cutoff = NULL, verbose = TRUE)
```

Arguments

`object`	An object of class `CMSproc`.
`cutoff`	A number indicating the threshold to apply to the density estimate. `NULL` indicates that a data-driven threshold should be chosen.
`verbose`	Should the function be verbose?

Details

`slicepi` uses the most intense features in set regions of the M/Z space to identify a data-driven density cutoff to detect peaks. Once peak bounds have been computed, the extracted ion chromatograms for the peaks are computed, and the EICs are integrated to obtain peak quantifications.

Value

An object of class `CMSslice` containing peak bounds and quantifications as well as sample and preprocessing metadata.

Examples

```
data(cmsRawExample)
cmsProc <- bakedpi(cmsRawExample, dbandwidth = c(0.01, 10), dgridstep = c(0.01, 1),
                   outfileDens = NULL, dortalign = FALSE, verbose = TRUE)
dqs <- densityQuantiles(cmsProc)
cmsSlice <- slicepi(cmsProc, cutoff = dqs[996], verbose = TRUE)
cmsSlice
```

Index

∗ **datasets**
cmsRawExample, 6

bakedpi, 2

CMSproc-class, 3
CMSraw-class, 5
cmsRawExample, 6
CMSslice *(*CMSslice-class*)*, 6
CMSslice-class, 6
colData,CMSraw-method *(*CMSraw-class*)*, 5

densityCutoff *(*CMSslice-class*)*, 6
densityEstimate *(*CMSproc-class*)*, 3
densityQuantiles *(*CMSproc-class*)*, 3
diffrep, 7

getEICS, 8
getTIC, 9

peakBounds *(*CMSslice-class*)*, 6
peakQuants *(*CMSslice-class*)*, 6
plotDensityRegion, 9

readMSdata, 10

show,CMSproc-method *(*CMSproc-class*)*, 3
show,CMSraw-method *(*CMSraw-class*)*, 5
show,CMSslice-method *(*CMSslice-class*)*, 6
slicepi, 11