TarMet

Hongchao Ji

2017/12/17

Introduction

TarMet is a shiny application for targeted metabolic analyses based on mass spectrometry. TarMet can detect all of the isotopologues of target metabolites, and quantify the metabolites based on the peak areas automatically and efficiently.

Usage

Isotopic Analysis

Upload Raw Data

TarMet uses *mzR* to parse raw data. The upload files can be *mzML*, *mzXML*, *mzData* or *CDF* format. Only one file can be upload in the isotope analysis step, while multiple files can be accepted in the quantitative analysis step.

Metabolite Information

The matabolite can be input as formulas or monoisotopic mass. If you choose to input a formula, please keep the monoisotopic mass input as *-1*, or the input formula is useless. You can select which kinds of adduct to be detected. Usually *M+H*, *M+Na* and *M+K* are common in positive mode, while *M-H* and *M+Cl* are common in negative mode.

Isotopic Information

The only parameter is the threshold of relative abundance. Only peaks above the threshold are under consideration.

EIC Extraction

The extraction ion chromatograms are obtained by sum the data points within a specific m/z tolerance, given as ppm. You can also choose to remove the baseline via airPLS algorithm.

Peak Detection

The peak detection is based on *MassSpecWavelet* package. Three parameters should be given, which are the minimum snrs, minimum scales of peaks and minimum intensities of peaks.

EIC Plot

The EIC Plot will contain the EICs obatined based on the input, as well as the detected peaks and the bounds of peaks. If there is no peaks in the given m/z range, it will show nothing, please adjust your input.

Peak Information

This table will shows the information of obtained peaks based on the input, including m/z, peak range, peak area, etc.

Quantitative Analysis

Upload Raw Data

In this step, you can upload multiple data files. The target metabolite will be extracted from all of the input data files. The information used is the same as the last step.

Alignment

Usually, retention time of the same metabolite may differs between different samples. You can choose whether to correct the time shift.

Isotope Information

Choose which isotopologue is used for quantitative analysis. It is the index of the peaks in the peak information table.

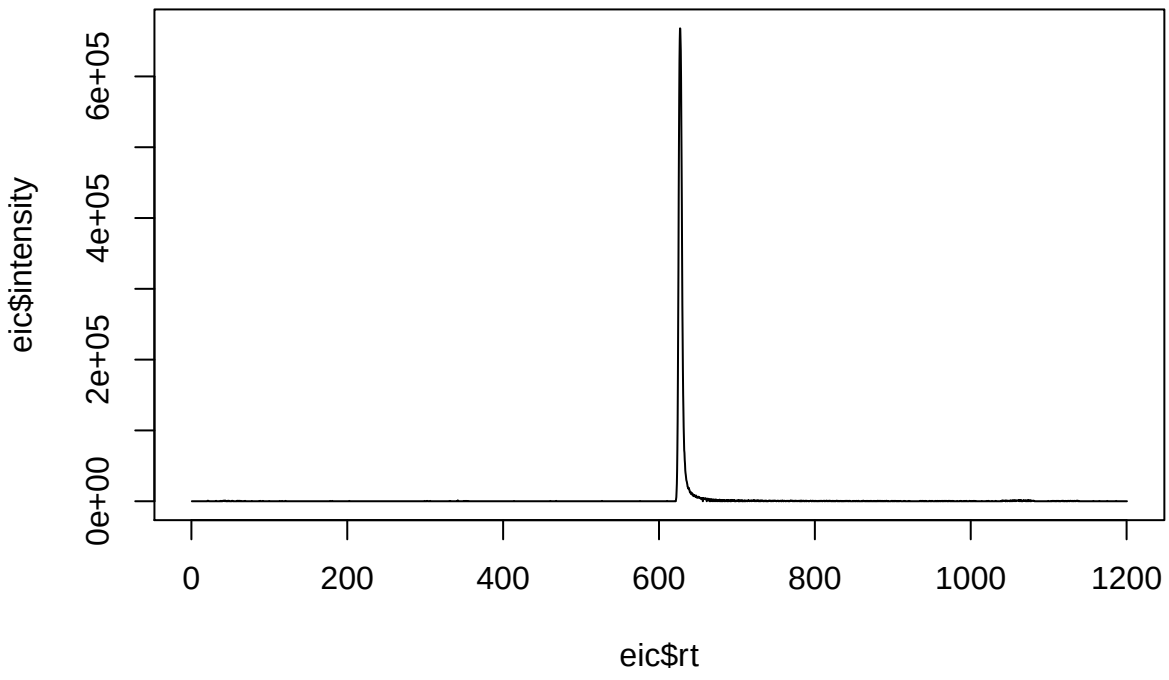
Functions

Actually, for the end user, all of the functions can be used in the shiny app. Here is some funtions invoked in the app. The MTBLS2 dataset is used for an example.

```
# Load Dataset
library(TarMet)
library(mtbls2)
filepath <- file.path(find.package("mtbls2"), "mzData")
files <- file.path(filepath, list.files(filepath, recursive = TRUE))[1:3]

file1 <- files[1]
raw <- LoadData(file1)

# Get EICs from raw data directly
eic <- getEIC(raw, rtrange = c(0, Inf), mzrange = c(201.03, 201.05))
plot(eic$rt, eic$intensity, 'l')
```




```
# But usually we usually get the isotopic EICs of a specific formula at the same time.
eics <- getIsoEIC(raw, formula = NULL, fmz= 200.04, adduct = "M+H") # or you can give the formula, if s
# If the formula is given, the isotopic pattern is obtained by the function of
pattern <- getIsoPat('C16H12O5', 'M+H', 0.2)
```

```
##  done.
```

```
pattern
```

```
##           m/z     abundance
## 1 285.0762985 100.000000000
## 2 286.0796534  17.305165268
## 3 287.0805443   1.027496817
## 4 287.0830082   1.403759742
```

```
# Then we can detect peak from the EICs, and get the peak areas.
peaks <- getIsoPeaks(eics, SNR.Th = 4, peakScaleRange = 5)
# The user can also define the start and the end of a peak manually, and the area is calculated by:
areas <- getArea(eics, rtmin = 0, rtmax = Inf)
# The method of calculating the area is by trapezoidal integration, for example:
res <- integration(x = 1:20,yf = 20:1)

# If more than one example is processed, TarMet uses a alignment method to correct the time shift of EI

# get eics from a set of samples:
rawfiles <- lapply(files, LoadData)
eics <- lapply(rawfiles, function(raw){
  getEIC(raw, c(0, Inf), mzrange = c(201.03, 201.05))
```

```
})
# correct the time shift between sample 1 and sample 2 (for example)
spectra <- t(eics[[1]]$intensity)
reference <- t(eics[[2]]$intensity)
res <- PAFFT(spectra, reference, segSize = 150, shift = 50)
```