Package: MsQuantitation (via r-universe)

August 28, 2024

Title Mass Spectrometry Quantitation

Version 0.0.2

Description This package quantitates raw mass spectrometry data contained in Spectra objects into QFeaures instances.

Depends Spectra, QFeatures

Imports methods, BiocGenerics, ProtGenerics, S4Vectors, BiocParallel, MsCoreUtils, SummarizedExperiment

Suggests msdata, mzR, knitr, testthat (>= 3.0.0)

License Artistic-2.0

VignetteBuilder knitr

URL https://github.com/RforMassSpectrometry/MsQuantification

BugReports https://github.com/RforMassSpectrometry/MsQuantification/issues

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Config/testthat/edition 3

Repository https://rformassspectrometry.r-universe.dev

RemoteUrl https://github.com/rformassspectrometry/MsQuantitation

RemoteRef HEAD

RemoteSha 57bcf311e58038f011e4f8d36cc825954bf925c7

Contents

quantify,Spectra-method

Quantitation of Raw MS Data

Description

The `quantify()` method is the entry point for quantitation of raw mass spectrometry data. The raw data is contained in a `Spectra()` object while the details and parameters of the quantitation method are defined in a dedicated `QuantParam()`.

Usage

```
## S4 method for signature 'Spectra'
quantify(object, param, ...)
```

Arguments

`object`	An instance of class `Spectra()`.
`param`	An instance of class `QuantParam()` that defines the quantitation method.
`...`	additional parameters controlled parallel processing of the data. See `BiocParallel::bplapply()` for details.

Value

An instance of `QFeatures()` with as many assays as there where acquisitions (files) in `object`.

Examples

```
## ----------------------------------------
## Labelled MS2 quantitation
## ----------------------------------------

## Test data from the msdata package
f <- msdata::proteomics(pattern = "01.mzML.gz", full.names = TRUE)
rw <- Spectra(f)

## Quantitation parameters
p <- QuantParam(msLevel = 2L, label = TRUE,
                params = list(reporters = TMT6))
p

quantify(rw, p)

## Simulate data from 2 files
rw <- filterMsLevel(rw, 2L)
rw <- setBackend(rw, MsBackendDataFrame())

rw$dataOrigin <- sample(c("file1", "file2"),
```

```
                        length(rw),
                        replace = TRUE)

quantify(rw, p)

## ----------------------------------------
## Labelled MS3 quantitation
## ----------------------------------------

## Test data from the msdata pakage
basename(f <- msdata::proteomics(pattern = "MS3TMT11", full.names = TRUE))
x <- Spectra(f)

## Quantitation parameters
p <- QuantParam(msLevel = 3L, label = TRUE,
                params = list(reporters = TMT11))
p

quantify(x, p)
```

`quantify_labelled_ms2` *Labelled MS2 Quantitation*

Description

Function that converses a `Spectra` instance into an object of class `QFeatures` by quantifying reporter ions of the MS2 spectra.

Usage

```
quantify_labelled_ms2(x, reporters, ...)
```

Arguments

`x`	An instance of class Spectra with MS2 spectra.
`reporters`	An instance of class ReporterIons.
`...`	Additional parameters passed to `BiocParallel::bplapply()`.

Value

A instance of class QFeatures with as many assays as there where acquisitions in `x`.

Author(s)

Laurent Gatto

quantify_labelled_ms3 *Labelled MS3 Quantitation*

Description

Function that converses a `Spectra` instance into an object of class `QFeatures` by quantifying reporter ions of the MS3 spectra. The assays rowData are pulled from the respective MS2 precursor scans.

Usage

```
quantify_labelled_ms3(x, reporters, ...)
```

Arguments

`x`	An instance of class Spectra with MS2 and MS3 spectra.
`reporters`	An instance of class ReporterIons.
`...`	Additional parameters passed to `BiocParallel::bplapply()`.

Value

A instance of class QFeatures with as many assays as there where acquisitions in `x`.

Author(s)

Laurent Gatto

QuantParam *Quantiation Parameters*

Description

Quantitation of mass spectrometry data is implemented through the `quantify()` method, that converts raw data objects such as `Spectra()` into `QFeatures()` objects that store and hande quantitative data. Quantitation methods for MS data a numerous and varied, and the exact method to be applied on the raw data is paramterised by a `QuantParam` instance, i.e. a dedicated class that stores all the parameters needed to perform a specific quantitation method.

MS quantitation can be performed at the MS1 or MS2 (and 3) levels. Additionally, quantitation can be label-free or labelled. These two states are defined by the `msLevel` (an integer) and `label` (a logical) parameters.

Additional details about the quantitation method are defined as a named list, as described in the 'Details' section. The example below illustrates how to defined labelled (`label = TRUE`) MS2 (`msLevel = 2L`) quantitation using TMT 10-plex isobaric tagging (defined by the TMT10' `ReporterIons()` instance).

Usage

```
## S4 method for signature 'QuantParam'
isEmpty(x)

QuantParam(msLevel = NA_integer_, label = NA, params = list())
```

Arguments

x	A instance of class QuantParam.
msLevel	integer(1) indicating the MS levels to be quantified. Default is NA.
label	logical(1) defining is labelled or label-free quantitation. Default is NA.
params	list() containing additional method-specific quantitation parameters. The list must be named.

Details

Labelled MS2 Quantitation:

- The params need only to contain a single mandatory element, a ReporterIons() instance, named reporters.

Labelled MS3 Quantitation:

- The params need to contain a mandatory element, a ReporterIons() instance, named reporters.
- Two optional character strings, named acquisitionNum and precScanNum, containing the respective column names of the acquition number and precursor scan acquisition numbers in the raw data to be quantified. The default values are "acquisitionNum" and "precScanNum" respectively, which match the names in standard Spectra objects.

Slots

msLevel integer(1) indicating the MS level the quantitation will be performed on.

label

params named list() of additional parameters.

Author(s)

Laurent Gatto

Examples

```
## default (empty) parameters
QuantParam()

isEmpty(QuantParam())

## MS2 quantitation using TMT10 plex
QuantParam(msLevel = 2L,
           label = TRUE,
```

```
           params = list(reporters = TMT10))

## MS3 quantitation using TMT16 plex and non-standard acquisition
## number and precursor scan acqusition number
QuantParam(msLevel = 2L, label = TRUE,
           params = list(reporters = TMT10,
                         acqusitionNum = "aquisition_number",
                         preScanNum = "prec_scan_number"))
```

ReporterIons *Reporter Ions Class*

Description

The ReporterIons class defines a set of isobaric reporter ions that are used for quantification of labelled MS2 data, e.g. iTRAQ (isobaric tag for relative and absolute quantitation) or TMT (tandem mass tags).

Many commercial reporter ions are readily available. Custom instances can be created with the ReporterIons() function.

Usage

```
ReporterIons(name, reporterNames, mz, width)

## S4 method for signature 'ReporterIons'
show(object)

## S4 method for signature 'ReporterIons,ANY,ANY,ANY'
x[i = "numeric", j = "missing", drop = "missing"]

## S4 method for signature 'ReporterIons'
length(x)

## S4 method for signature 'ReporterIons'
mz(object)

## S4 method for signature 'ReporterIons'
width(x)

reporterNames(x)

reporterNames(x) <- value

## S4 method for signature 'ReporterIons'
names(x)

iTRAQ4
```

iTRAQ5

iTRAQ8

iTRAQ9

TMT10HCD

TMT10

TMT10ETD

TMT11HCD

TMT11

TMT16HCD

TMT16

TMT6

TMT6b

TMT7

TMT7b

Arguments

name	Parameter to set the name slot.
reporterNames	Parameter to set the reporterNames slot.
mz	Parameter to set the mz slot.
width	Parameter to set the width slot.
object	An instance of class ReporterIons.
x	An instance of class ReporterIons.
i	numeric() to subset x.
j	ignored.
drop	ignored.
value	A value for replacement.

Format

An object of class ReporterIons of length 4.

An object of class ReporterIons of length 5.

An object of class `ReporterIons` of length 8.

An object of class `ReporterIons` of length 9.

An object of class `ReporterIons` of length 10.

An object of class `ReporterIons` of length 10.

An object of class `ReporterIons` of length 10.

An object of class `ReporterIons` of length 11.

An object of class `ReporterIons` of length 11.

An object of class `ReporterIons` of length 16.

An object of class `ReporterIons` of length 16.

An object of class `ReporterIons` of length 6.

An object of class `ReporterIons` of length 6.

An object of class `ReporterIons` of length 7.

An object of class `ReporterIons` of length 7.

Slots

`name character(1)` naming the instance.

`reporterNames character()` of length equal to the number of reporter ions in the instance. Used to uniquely name each ion.

`mz numeric()` with the reporter ions' m/z values.

`width numeric(1)` defining the region where the reporter ion can be expected. This region is calculated as `mz` +/- `width` for earch reporter ion.

Author(s)

Laurent Gatto

References

Ross PL, Huang YN, Marchese JN, Williamson B, Parker K, Hattan S, Khainovski N, Pillai S, Dey S, Daniels S, Purkayastha S, Juhasz P, Martin S, Bartlet-Jones M, He F, Jacobson A, Pappin DJ. *Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents.* Mol Cell Proteomics, 2004 Dec;3(12):1154-69. Epub 2004 Sep 22. PubMed PMID: 15385600.

Thompson A, Schäfer J, Kuhn K, Kienle S, Schwarz J, Schmidt G, Neumann T, Johnstone R, Mohammed AK, Hamon C. *Tandem mass tags: a novel quantification strategy for comparative analysis of complex protein mixtures by MS/MS.* Anal Chem. 2003 Apr 15;75(8):1895-904. Erratum in: Anal Chem. 2006 Jun 15;78(12):4235. Mohammed, A Karim A and Anal Chem. 2003 Sep 15;75(18):4942. Johnstone, R. PubMed PMID: 12713048.

Examples

```
## Manual construction of a ReporterIons instance:
ReporterIons(name = "iTRAQ4",
             reporterNames = c("iTRAQ4.114", "iTRAQ4.115",
                               "iTRAQ4.116", "iTRAQ4.117"),
             mz = c(114.1, 115.1, 116.1, 117.1),
             width = 0.05)

## Some pre-defined reporter ions constructors:
TMT10

iTRAQ4

TMT10[1:5]

reporterNames(TMT10)

names(TMT10)

mz(TMT10)

width(TMT10)

as(TMT10, "DataFrame")
```

Index