Package 'MALDIquantForeign'

January 20, 2025

Version 0.14.1

Date 2024-01-22

Title Import/Export Routines for 'MALDIquant'

Depends R (>= 3.2.2), methods, MALDIquant (>= 1.16.4)

Imports base64enc, digest, readBrukerFlexData (>= 1.7), readMzXmlData
(>= 2.7), XML

Suggests knitr, testthat (>= 0.8), RNetCDF (>= 1.6.1)

Description Functions for reading (tab, csv, Bruker fid, Ciphergen
XML, mzXML, mzML, imzML, Analyze 7.5, CDF, mMass MSD) and
writing (tab, csv, mMass MSD, mzML, imzML) different file
formats of mass spectrometry data into/from 'MALDIquant'
objects.

License GPL (>= 3)

URL <https://strimmerlab.github.io/software/maldiquant/>,

<https://github.com/sgibb/MALDIquantForeign/>

BugReports <https://github.com/sgibb/MALDIquantForeign/issues/>

LazyLoad yes

VignetteBuilder knitr

RoxygenNote 7.3.0

Encoding UTF-8

NeedsCompilation no

Author Sebastian Gibb [aut, cre] (<<https://orcid.org/0000-0001-7406-4443>>),
Pietro Franceschi [ctb]

Maintainer Sebastian Gibb <mail@sebastiangibb.de>

Repository CRAN

Date/Publication 2024-01-22 21:32:54 UTC

Contents

MALDIquantForeign-package

Import/Export routines for 'MALDIquant'

Description

This package reads and writes different file formats of mass spectrometry data into/from 'MALDIquant' objects.

Details

Package:	MALDIquantForeign
License:	GPL (>= 3)
URL:	https://strimmerlab.github.io/software/maldiquant/

Author(s)

Sebastian Gibb <mail@sebastiangibb.de>

References

https://strimmerlab.github.io/software/maldiquant/

See Also

Useful links:

- <https://strimmerlab.github.io/software/maldiquant/>
- <https://github.com/sgibb/MALDIquantForeign/>
- Report bugs at <https://github.com/sgibb/MALDIquantForeign/issues/>

---

export,AbstractMassObject-method
Export files

Description

This function provides a general interface to export AbstractMassObject-class objects (e.g. MassSpectrum-class, MassPeaks-class) into different file formats.

Usage

```
## S4 method for signature 'AbstractMassObject'
export(x, file, type="auto", force=FALSE, ...)

## S4 method for signature 'list'
export(x, path, type, force=FALSE, ...)
```

Arguments

x	a AbstractMassObject-class object or a list of AbstractMassObject-class objects.
file	character, file name.
type	character, file format. If type is set to "auto" the file extension is used.
force	logical, If TRUE the file would be overwritten or path would be created.
path	character, path to directory in which the list of AbstractMassObject-class would be exported.
...	arguments to be passed to specific export functions.

Details

Specific export functions:

tab	exportTab
csv	exportCsv
imzML	exportImzMl
msd	exportMsd
mzML	exportMzMl

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>

See Also

[MassPeaks-class](), [MassSpectrum-class]()

Examples

```
## Not run:
library("MALDIquant")
library("MALDIquantForeign")

s <- list(createMassSpectrum(mass=1:5, intensity=1:5),
          createMassSpectrum(mass=1:5, intensity=1:5))

## export a single spectrum
export(s[[1]], file="spectrum.csv")
## identical to exportCsv(s[[1]], file="spectrum.csv")

## export a list of spectra
export(s, path="spectra", type="csv")
## identical to exportCsv(s, path="spectra")

## End(Not run)
```

---

exportImzMl,MassSpectrum-method
                              Export to imzML files

Description

This function exports [MassSpectrum-class]() objects into imzML files.

Usage

```
## S4 method for signature 'MassSpectrum'
exportImzMl(x, file, force=FALSE, processed=TRUE,
coordinates=NULL, pixelSize=c(100, 100), ...)

## S4 method for signature 'list'
exportImzMl(x, path, force=FALSE, processed=TRUE,
coordinates=NULL, pixelSize=c(100, 100), ...)
```

Arguments

x	a `AbstractMassObject-class` object or a `list` of `AbstractMassObject-class` objects.
file	character, file name.
force	logical, If TRUE the `file` would be overwritten or `path` would be created.
processed	logical, If TRUE (default) the spectra will be saved in processed mode (means mass and intensity is stored for each spectra separately in contrast to continuous mode where the mass is stored only for one spectrum).
coordinates	matrix, 2 column matrix that contains the x- and y-coordinates for the spectra.
pixelSize	numeric, a vector of length 2 that contains the x and y pixel size in micrometers (default: c(100, 100)).
path	character, path to directory in which the `list` of `MassSpectrum-class` would be exported. If `path` is a single filename all spectra will be exported to a single imzML file.
...	arguments to be passed to internal functions.

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>

Schramm T, Hester A, Klinkert I, Both J-P, Heeren RMA, Brunelle A, Laprevote O, Desbenoit N, Robbe M-F, Stoeckli M, Spengler B, Roempp A (2012)
imzML - A common data format for the flexible exchange and processing of mass spectrometry imaging data.
Journal of Proteomics 75 (16):5106-5110.
doi:10.1016/j.jprot.2012.07.026

See Also

MassSpectrum-class

Examples

```
## Not run:
library("MALDIquant")
library("MALDIquantForeign")

s <- list(createMassSpectrum(mass=1:5, intensity=1:5),
          createMassSpectrum(mass=1:5, intensity=1:5))

## export a list of spectra
exportImzMl(s, path="processed.imzML", coordinates=cbind(x=1:2, y=c(1, 1)))

## End(Not run)
```

---

exportMsd,MassSpectrum-method

Export to MSD files

Description

This function exports `AbstractMassObject-class` objects (e.g. `MassSpectrum-class`, `MassPeaks-class`) into mMass MSD files.

Usage

```
## S4 method for signature 'MassSpectrum'
exportMsd(x, file, force=FALSE, peaks, ...)

## S4 method for signature 'list'
exportMsd(x, path, force=FALSE, peaks, ...)
```

Arguments

x	a `MassSpectrum-class` object or a list of `MassSpectrum-class` objects.
file	character, file name.
force	logical, If TRUE the file would be overwritten or path would be created.
peaks	a `MassPeaks-class` object or a list of `MassPeaks-class` objects.
path	character, path to directory in which the list of `AbstractMassObject-class` would be exported.
...	arguments to be passed to `write.table`.

Author(s)

Sebastian Gibb

References

https://strimmerlab.github.io/software/maldiquant/,
mMass homepage: http://mmass.org/

See Also

`MassPeaks-class`, `MassSpectrum-class`

Examples

```
## Not run:
library("MALDIquant")
library("MALDIquantForeign")

s <- list(createMassSpectrum(mass=1:5, intensity=1:5),
          createMassSpectrum(mass=1:5, intensity=1:5))
p <- list(createMassPeaks(mass=4:5, intensity=4:5, snr=1:2),
          createMassPeaks(mass=4:5, intensity=4:5, snr=1:2))

## export a single spectrum
exportMsd(s[[1]], file="spectrum.msd")

## export a single spectrum with corresponding peaks
exportMsd(s[[1]], file="spectrum.msd", peaks=p[[1]])

## export a list of spectra with corresponding peaks
exportMsd(s, path="spectra", peaks=p, force=TRUE)

## End(Not run)
```

---

exportMzMl,MassSpectrum-method
Export to mzML files

Description

This function exports MassSpectrum-class objects into mzML files.

Usage

```
## S4 method for signature 'MassSpectrum'
exportMzMl(x, file, force=FALSE, ...)

## S4 method for signature 'list'
exportMzMl(x, path, force=FALSE, ...)
```

Arguments

x	a AbstractMassObject-class object or a list of AbstractMassObject-class objects.
file	character, file name.
force	logical, If TRUE the file would be overwritten or path would be created.
path	character, path to directory in which the list of MassSpectrum-class would be exported. If path is a single filename all spectra will be exported to a single mzML file.
...	arguments to be passed to internal functions.

Author(s)

Sebastian Gibb

References

https://strimmerlab.github.io/software/maldiquant/,
HUPO Proteomics Standards Inititative mzML 1.1.0 Specification: https://www.psidev.info/
mzML

See Also

MassSpectrum-class

Examples

```
## Not run:
library("MALDIquant")
library("MALDIquantForeign")

s <- list(createMassSpectrum(mass=1:5, intensity=1:5),
          createMassSpectrum(mass=1:5, intensity=1:5))

## export a single spectrum
exportMzMl(s[[1]], file="spectrum.mzML")

## export a list of spectra
exportMzMl(s, path="spectra.mzML")

## End(Not run)
```

---

exportTab,AbstractMassObject-method
                                    Export to text files

Description

This function exports AbstractMassObject-class objects (e.g. MassSpectrum-class, MassPeaks-class)
into different text file formats.

Usage

```
## S4 method for signature 'AbstractMassObject'
exportTab(x, file, force=FALSE, ...)

## S4 method for signature 'list'
exportTab(x, path, force=FALSE, ...)
```

```
## S4 method for signature 'AbstractMassObject'
exportCsv(x, file, force=FALSE, ...)

## S4 method for signature 'list'
exportCsv(x, path, force=FALSE, ...)
```

Arguments

x	a `AbstractMassObject-class` object or a list of `AbstractMassObject-class` objects.
file	character, file name.
force	logical, If TRUE the `file` would be overwritten or `path` would be created.
path	character, path to directory in which the list of `AbstractMassObject-class` would be exported.
...	arguments to be passed to `write.table`.

Details

exportTab and exportCsv use `write.table` with different defaults (sep="\t" in exportTab and sep="," in exportCsv).

Author(s)

Sebastian Gibb

References

https://strimmerlab.github.io/software/maldiquant/

See Also

`MassPeaks-class`, `MassSpectrum-class`, `write.table`

Examples

```
## Not run:
library("MALDIquant")
library("MALDIquantForeign")

s <- list(createMassSpectrum(mass=1:5, intensity=1:5),
          createMassSpectrum(mass=1:5, intensity=1:5))

## export a single spectrum
exportTab(s[[1]], file="spectrum.tab")

## export a list of spectra and use ; as separator
exportCsv(s, path="spectra", sep=";", force=TRUE)

## End(Not run)
```

---

import   *Import files*

Description

This function provides a general interface to import different file formats into `MassSpectrum-class`
or `MassPeaks-class` objects.

Usage

```
import(
  path,
  type = "auto",
  pattern,
  excludePattern = NULL,
  removeEmptySpectra = TRUE,
  centroided = FALSE,
  massRange = c(0, Inf),
  minIntensity = 0,
  mc.cores = 1L,
  verbose = interactive(),
  ...
)
```

Arguments

path	character, path to directory or file which should be read in.
type	character, file format. If `type` is set to "auto" MALDIquant tries to detect the correct file type automatically. It often depends on the file extension (if `path` is a directory the most represented file extension is used; `pattern` argument is ignored).
pattern	character, a regular expression to find files in a directory (see details).
excludePattern	character, a regular expression to exclude files in a directory (see details).
removeEmptySpectra	
	logical, should empty spectra excluded?
centroided	logical, if `centroided=FALSE` (default) the data are treated as not centroided and a list of `MassSpectrum-class` objects is returned. Use `centroided=TRUE` to assume centroided data and get a list of `MassPeaks-class` objects.
massRange	double, limits of mass import (left/minimal mass, right/maximal mass).
minIntensity	double, minimal intensity to import.
mc.cores	number of cores to use (default 1; only unix-based platforms are supported, see `MALDIquantForeign-parallel` for details).
verbose	logical, verbose output?
...	arguments to be passed to specific import functions.

Details

Specific import functions:

txt	[importTxt](#)
tab	[importTab](#)
csv	[importCsv](#)
fid	[importBrukerFlex](#)
ciphergen	[importCiphergenXml](#)
mzXML	[importMzXml](#)
mzML	[importMzMl](#)
imzML	[importImzMl](#)
analyze	[importAnalyze](#)
cdf	[importCdf](#)
msd	[importMsd](#)

path: In addition to the above mentioned file types the following (compressed) archives are supported, too: zip, tar, tar.gz, tar.bz2, tar.xz. The archives are uncompressed in a temporary directory. Afterwards the [import](#) function is called (with type="auto").

pattern: Sometimes unusual file extensions are used (e.g. "*.xml" for mzXML files). In this case a specific pattern could be defined to import files with an unusual file extension (e.g. pattern="^.*\.xml$" to read all *.xml files in a directory; see [regexp](#) for details).

excludePattern: Sometimes some files should be excluded. E.g. to ignore additional aquired Bruker LIFT spectra (MALDI-TOF/TOF; which are not support, yet) you could use excludePattern="([[:digit:]\.]+)L

Value

a list of [MassSpectrum-class](#) or [MassPeaks-class](#) objects (depending on the centroided argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>

See Also

[MassSpectrum-class](#), [MassPeaks-class](#) [MALDIquantForeign-parallel](#)

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")
```

```
## import mzXML files
s <- import(exampleDirectory, type="mzXML")

## import tab delimited file with different file extension (default: *.tab)
s <- import(exampleDirectory, type="tab", pattern="^.*\\.txt")

## import single mzML file
s <- import(file.path(exampleDirectory, "tiny1.mzML1.1.mzML"))

## import gzipped csv file
s <- import(file.path(exampleDirectory, "compressed", "csv1.csv.gz"))
```

---

importAnalyze   *Import Analyze 7.5 files*

Description

This function imports files in Analyze 7.5 file format into `MassSpectrum-class` or `MassPeaks-class` objects.

Usage

```
importAnalyze(path, ...)
```

Arguments

path	character, path to directory or file which should be read in.
...	arguments to be passed to `import`.

Value

a `list` of `MassSpectrum-class` or `MassPeaks-class` objects (depending on the `centroided` argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>
<http://www.grahamwideman.com/gw/brain/analyze/formatdoc.htm>, <https://eeg.sourceforge.net/ANALYZE75.pdf>

See Also

`MassSpectrum-class`, `MassPeaks-class`

---

importBrukerFlex   *Import Bruker Daltonics *flex files*

Description

This function imports files in Bruker Daltonics *flex-series file format into `MassSpectrum-class` or `MassPeaks-class` objects.

Usage

```
importBrukerFlex(path, ...)
```

Arguments

path   character, path to directory or file which should be read in.

...   arguments to be passed to `readBrukerFlexFile`.

Value

a `list` of `MassSpectrum-class` or `MassPeaks-class` objects (depending on the `centroided` argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>

See Also

`MassSpectrum-class`, `MassPeaks-class`, `readBrukerFlexFile`

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

s <- importBrukerFlex(exampleDirectory)
```

---

importCdf   *Import CDF files*

Description

This function imports files in NetCDF file format into MassSpectrum-class or MassPeaks-class
objects.
Please note that the *RNetCDF* is needed.

Usage

```
importCdf(path, ...)
```

Arguments

path   character, path to directory or file which should be read in.

...   arguments to be passed to import.

Value

a list of MassSpectrum-class or MassPeaks-class objects (depending on the centroided ar-
gument).

Author(s)

Sebastian Gibb

References

https://strimmerlab.github.io/software/maldiquant/

See Also

MassSpectrum-class, MassPeaks-class

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import
if (requireNamespace("RNetCDF", quietly=TRUE)) {
  s <- importCdf(exampleDirectory)
} else {
  message("You have to install the RNetCDF package to use importCdf.")
}
```

---

importCiphergenXml   *Import Ciphergen XML files*

Description

This function imports files in Ciphergen XML file format into `MassSpectrum-class` or `MassPeaks-class` objects.

Usage

```
importCiphergenXml(path, ...)
```

Arguments

path	character, path to directory or file which should be read in.
...	arguments to be passed to `import`.

Value

a `list` of `MassSpectrum-class` or `MassPeaks-class` objects (depending on the `centroided` argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>

See Also

`MassSpectrum-class`, `MassPeaks-class`

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import
s <- importCiphergenXml(exampleDirectory)
```

---

importImzMl *Import imzML files*

Description

This function imports files in imzML file format into MassSpectrum-class or MassPeaks-class objects.

Usage

```
importImzMl(path, coordinates = NULL, ...)
```

Arguments

path	character, path to directory or file which should be read in.
coordinates	matrix, 2 column matrix that contains the x- and y-coordinates for spectra that should be imported. Other spectra would be ignored.
...	arguments to be passed to import.

Value

a list of MassSpectrum-class or MassPeaks-class objects (depending on the centroided argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>,
Definition of imzML format: <https://ms-imaging.org/imzml/>

See Also

MassSpectrum-class, MassPeaks-class

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import
s <- importImzMl(file.path(exampleDirectory, "tiny_continuous.imzML"))
```

```
## import only spectra for pixel 1,1 and 2,1
s <- importImzMl(file.path(exampleDirectory, "tiny_continuous.imzML"),
                 coordinates = cbind(1:2, c(1, 1)))
```

---

importMsd   *Import MSD files*

Description

This function imports files in mMass MSD file format into `MassSpectrum-class` or `MassPeaks-class` objects.

Usage

```
importMsd(path, ...)
```

Arguments

path	character, path to directory or file which should be read in.
...	arguments to be passed to `import`.

Value

a `list` of `MassSpectrum-class` or `MassPeaks-class` objects (depending on the `centroided` argument).

Author(s)

Sebastian Gibb

References

https://strimmerlab.github.io/software/maldiquant/,
mMass homepage: http://mmass.org/

See Also

`MassSpectrum-class`, `MassPeaks-class`

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import
```

```
s <- importMsd(exampleDirectory)
```

---

importMzMl   *Import mzML files*

Description

This function imports files in mzML file format into [MassSpectrum-class](#) or [MassPeaks-class](#) objects.

Usage

```
importMzMl(path, ...)
```

Arguments

path	character, path to directory or file which should be read in.
...	arguments to be passed to [import](#).

Value

a list of [MassSpectrum-class](#) or [MassPeaks-class](#) objects (depending on the centroided argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>,
Definition of mzML format: <https://www.psidev.info/mzML>

See Also

[MassSpectrum-class](#), [MassPeaks-class](#)

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import
s <- importMzMl(exampleDirectory)
```

---

importMzXml *Import mzXML files*

Description

This function imports files in mzXML file format into [MassSpectrum-class](#) or [MassPeaks-class](#) objects.

Usage

```
importMzXml(path, ...)
```

Arguments

path   character, path to directory or file which should be read in.

...   arguments to be passed to [readMzXmlFile](#).

Value

a list of [MassSpectrum-class](#) or [MassPeaks-class](#) objects (depending on the centroided argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>,
Definition of mzXML format: http://tools.proteomecenter.org/wiki/index.php?title=Formats:mzXML

See Also

[MassSpectrum-class](#), [MassPeaks-class](#), [readMzXmlFile](#)

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import
s <- importMzXml(exampleDirectory)
```

---

importTxt   *Import text files*

Description

This function imports different text file formats into `MassSpectrum-class` or `MassPeaks-class` objects.

Usage

```
importTxt(path, ...)

importTab(path, ...)

importCsv(path, ...)
```

Arguments

path   character, path to directory or file which should be read in.

...   arguments to be passed to `read.table`.

Details

importTab, importTxt and importCsv use `read.table` with different defaults.

Value

a `list` of `MassSpectrum-class` or `MassPeaks-class` objects (depending on the `centroided` argument).

Author(s)

Sebastian Gibb

References

<https://strimmerlab.github.io/software/maldiquant/>

See Also

`MassSpectrum-class`, `MassPeaks-class`, `read.table`

Examples

```
library("MALDIquant")
library("MALDIquantForeign")

## get example directory
exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## import txt files
s <- importTxt(exampleDirectory)

## import csv files
s <- importCsv(exampleDirectory)
```

MALDIquantForeign-parallel
Parallel Support in Package **MALDIquantForeign**

Description

MALDIquantForeign-package offers multi-core support using mclapply and mcmapply. This approach is limited to unix-based platforms.

Details

Please note that not all import functions benfit from parallelisation. The current implementation is limited to run the parallelisation over different files. That's why only imports of multiple files could be run on multiple cores. E.g. a single mzML file containing 4 spectra would always be read on a single core. In contrast 4 mzML files each containing just one spectra could be read in using 4 cores.

The improvement in the runtime depends on the amount of data to read, the proportion of parsing/decoding of the data, the amount of memory and the speed of the hard disk.

Please note: It is possible that using parallelisation results in a worse runtime!

Author(s)

Sebastian Gibb <mail@sebastiangibb.de>

References

https://strimmerlab.github.io/software/maldiquant/

See Also

MALDIquant-parallel, mclapply, mcmapply

Examples

```
## load packages
library("MALDIquant")
library("MALDIquantForeign")

exampleDirectory <- system.file("exampledata", package="MALDIquantForeign")

## run single-core import
print(system.time(
  s1 <- importMzMl(exampleDirectory, centroided=TRUE, verbose=FALSE)
))

if(.Platform$OS.type == "unix") {
  ## run multi-core import
  ## (because the example spectra are very small (just 5 data points) the
  ## multi-core solution is slower on most systems)
  print(system.time(
    s2 <- importMzMl(exampleDirectory, centroided=TRUE, mc.cores=2,
                     verbose=FALSE)
  ))
  stopifnot(all.equal(s1, s2))
}
```

---

supportedFileFormats   *Supported file formats*

Description

This function prints all file formats supported by [MALDIquantForeign-package](#).

Usage

```
supportedFileFormats()
```

Details

Import:

txt	[importTxt](#)
tab	[importTab](#)
csv	[importCsv](#)
fid	[importBrukerFlex](#)
ciphergen	[importCiphergenXml](#)
mzXML	[importMzXml](#)
mzML	[importMzMl](#)
imzML	[importImzMl](#)
analyze	[importAnalyze](#)
cdf	[importCdf](#)

msd	importMsd

Export:

tab	exportTab
csv	exportCsv
imzML	exportImzMl
msd	exportMsd
mzML	exportMzMl

Value

a `list` with two named elements (`import` and `export`) containing a `character` vector of supported file types.

Author(s)

Sebastian Gibb

References

https://strimmerlab.github.io/software/maldiquant/

See Also

export, import

Examples

```
library("MALDIquantForeign")

supportedFileFormats()
```

Index