

# Package ‘MALDIquantForeign’

August 11, 2023

**Version** 0.14

**Date** 2023-08-11

**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** 7.2.3

**NeedsCompilation** no

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

**Maintainer** Sebastian Gibb <[mail@sebastiangibb.de](mailto:mail@sebastiangibb.de)>

**Repository** CRAN

**Date/Publication** 2023-08-11 21:53:46 UTC

**R topics documented:**

MALDIquantForeign-package . . . . .	2
export,AbstractMassObject-method . . . . .	3
exportImzML,MassSpectrum-method . . . . .	4
exportMs,MassSpectrum-method . . . . .	6
exportMzML,MassSpectrum-method . . . . .	7
exportTab,AbstractMassObject-method . . . . .	8
import . . . . .	10
importAnalyze . . . . .	12
importBrukerFlex . . . . .	13
importCdf . . . . .	14
importCIPHERgenXML . . . . .	15
importImzML . . . . .	16
importMs . . . . .	17
importMzML . . . . .	18
importMzXML . . . . .	19
importTxt . . . . .	20
MALDIquantForeign-parallel . . . . .	21
supportedFileFormats . . . . .	22
<b>Index</b>	<b>24</b>

---

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/maldiquest/>

**Author(s)**

Sebastian Gibb <[mail@sebastiangibb.de](mailto:mail@sebastiangibb.de)>

**References**

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

---

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 <a href="#">AbstractMassObject-class</a> object or a list of <a href="#">AbstractMassObject-class</a> 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 <a href="#">AbstractMassObject-class</a> would be exported.
...	arguments to be passed to specific export functions.

## Details

Specific export functions:

tab	<a href="#">exportTab</a>
csv	<a href="#">exportCsv</a>
imzML	<a href="#">exportImzML</a>
msd	<a href="#">exportMsd</a>
mzML	<a href="#">exportMzML</a>

## 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 <a href="#">AbstractMassObject-class</a> object or a list of <a href="#">AbstractMassObject-class</a> 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 <code>MassSpectrum-class</code> 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](https://doi.org/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 <a href="#">MassSpectrum-class</a> object or a list of <a href="#">MassSpectrum-class</a> objects.
file	character, file name.
force	logical, If TRUE the file would be overwritten or path would be created.
peaks	a <a href="#">MassPeaks-class</a> object or a list of <a href="#">MassPeaks-class</a> objects.
path	character, path to directory in which the list of <a href="#">AbstractMassObject-class</a> would be exported.
...	arguments to be passed to <a href="#">write.table</a> .

### 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 <a href="#">AbstractMassObject-class</a> object or a list of <a href="#">AbstractMassObject-class</a> 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 <a href="#">MassSpectrum-class</a> 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 Initiative mzML 1.1.0 Specification: <https://www.psivdev.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 <a href="#">AbstractMassObject-class</a> object or a list of <a href="#">AbstractMassObject-class</a> 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 <a href="#">AbstractMassObject-class</a> would be exported.
...	arguments to be passed to <a href="#">write.table</a> .

### 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 <a href="#">MassSpectrum-class</a> objects is returned. Use centroided=TRUE to assume centroided data and get a list of <a href="#">MassPeaks-class</a> 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 <a href="#">MALDIquantForeign-parallel</a> for details).
verbose	logical, verbose output?
...	arguments to be passed to specific import functions.

## Details

Specific import functions:

txt	<a href="#">importTxt</a>
tab	<a href="#">importTab</a>
csv	<a href="#">importCsv</a>
fid	<a href="#">importBrukerFlex</a>
ciphergen	<a href="#">importCiphergenXml</a>
mzXML	<a href="#">importMzXml</a>
mzML	<a href="#">importMzMl</a>
imzML	<a href="#">importImzMl</a>
analyze	<a href="#">importAnalyze</a>
cdf	<a href="#">importCdf</a>
msd	<a href="#">importMsd</a>

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 acquired Bruker LIFT spectra (MALDI-TOF/TOF; which are not supported, 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	<i>Import Bruker Daltonics *flex files</i>
------------------	--

---

### 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 <a href="#">readBrukerFlexFile</a> .

### 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**

<code>path</code>	character, path to directory or file which should be read in.
<code>...</code>	arguments to be passed to <code>import</code> .

**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
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	<i>Import imzML files</i>
-------------	---------------------------

---

### 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 <a href="#">import</a> .

### 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.psivdev.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	<i>Import mzXML files</i>
-------------	---------------------------

---

## 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 <a href="#">readMzXmlFile</a> .

## 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	<i>Import text files</i>
-----------	--------------------------

---

## 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 <a href="#">read.table</a> .

## 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 benefit 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](mailto: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	<a href="#">importTxt</a>
tab	<a href="#">importTab</a>
csv	<a href="#">importCsv</a>
fid	<a href="#">importBrukerFlex</a>
ciphergen	<a href="#">importCiphergenXml</a>
mzXML	<a href="#">importMzXml</a>
mzML	<a href="#">importMzML</a>

imzML	<a href="#">importImzML</a>
analyze	<a href="#">importAnalyze</a>
cdf	<a href="#">importCdf</a>
msd	<a href="#">importMsd</a>

**Export:**

tab	<a href="#">exportTab</a>
csv	<a href="#">exportCsv</a>
imzML	<a href="#">exportImzML</a>
msd	<a href="#">exportMsd</a>
mzML	<a href="#">exportMzML</a>

**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

- \* **misc**
  - MALDIquantForeign-parallel, [21](#)
- \* **package**
  - MALDIquantForeign-package, [2](#)
- export, [23](#)
- export
  - (export, AbstractMassObject-method), [3](#)
- export, AbstractMassObject-method, [3](#)
- export, list-method
  - (export, AbstractMassObject-method), [3](#)
- exportCsv, [3](#), [23](#)
- exportCsv
  - (exportTab, AbstractMassObject-method), [8](#)
  - exportCsv, AbstractMassObject-method
    - (exportTab, AbstractMassObject-method), [8](#)
  - exportCsv, list-method
    - (exportTab, AbstractMassObject-method), [8](#)
- exportImzMl, [3](#), [23](#)
- exportImzMl
  - (exportImzMl, MassSpectrum-method), [4](#)
  - exportImzMl, list-method
    - (exportImzMl, MassSpectrum-method), [4](#)
  - exportImzMl, MassSpectrum-method, [4](#)
- exportMsd, [3](#), [23](#)
- exportMsd
  - (exportMsd, MassSpectrum-method), [6](#)
  - exportMsd, list-method
    - (exportMsd, MassSpectrum-method), [6](#)
  - exportMsd, MassSpectrum-method, [6](#)
- exportMzMl, [3](#), [23](#)
- exportMzMl
  - (exportMzMl, MassSpectrum-method), [7](#)
  - exportMzMl, list-method
    - (exportMzMl, MassSpectrum-method), [7](#)
  - exportMzMl, MassSpectrum-method, [7](#)
- exportTab, [3](#), [23](#)
- exportTab
  - (exportTab, AbstractMassObject-method), [8](#)
  - exportTab, AbstractMassObject-method, [8](#)
  - exportTab, list-method
    - (exportTab, AbstractMassObject-method), [8](#)
- import, [10](#), [11](#), [12](#), [14–18](#), [23](#)
- importAnalyze, [11](#), [12](#), [23](#)
- importBrukerFlex, [11](#), [13](#), [22](#)
- importCdf, [11](#), [14](#), [23](#)
- importCIPHERgenXml, [11](#), [15](#), [22](#)
- importCsv, [11](#), [22](#)
- importCsv (importTxt), [20](#)
- importImzMl, [11](#), [16](#), [23](#)
- importMsd, [11](#), [17](#), [23](#)
- importMzMl, [11](#), [18](#), [22](#)
- importMzXml, [11](#), [19](#), [22](#)
- importTab, [11](#), [22](#)
- importTab (importTxt), [20](#)
- importTxt, [11](#), [20](#), [22](#)
- MALDIquantForeign-package, [2](#)
- MALDIquantForeign-parallel, [21](#)
- mclapply, [21](#), [22](#)
- mcmapply, [21](#), [22](#)
- read.table, [20](#), [21](#)
- readBrukerFlexFile, [13](#)
- readMzXmlFile, [19](#)
- regexp, [11](#)



supportedFileFormats, [22](#)

write.table, [6](#), [9](#)