# Package 'readBrukerFlexData'

October 14, 2022

**Version** 1.9.0

**Date** 2022-06-19

**Title** Reads Mass Spectrometry Data in Bruker *flex Format

**Depends** R (>= 3.3.0)

**Suggests** testthat

**Description** Reads data files acquired by Bruker Daltonics' matrix-assisted laser desorption/ionization-time-of-flight mass spectrometer of the *flex series.

**License** GPL (>= 3)

**URL** https://strimmerlab.github.io/software/maldiquant/
https://github.com/sgibb/readBrukerFlexData/

**BugReports** https://github.com/sgibb/readBrukerFlexData/issues/

**LazyLoad** yes

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Sebastian Gibb [aut, cre] (<https://orcid.org/0000-0001-7406-4443>),
Samuel Granjeaud [ctb],
Alan Race [ctb] (<https://orcid.org/0000-0001-8996-2641>)

**Maintainer** Sebastian Gibb <mail@sebastiangibb.de>

**Repository** CRAN

**Date/Publication** 2022-06-19 17:40:02 UTC

## R topics documented:

readBrukerFlexData-package . . . . . 2
cpSpecHpcMzXml . . . . . 2
readBrukerFlexDir . . . . . 3
readBrukerFlexFile . . . . . 4

**Index** **8**

readBrukerFlexData-package *The readBrukerFlexData Package*

## Description

The readBrukerFlexData package reads data files acquired by MALDI-TOF MS on Bruker Daltonics machines of the *flex series. (autoflex, microflex, ultraflex).
The package was developed *without* any knowledge nor even support by Bruker Daltonics.

All trademarks are owned by or licensed to Bruker Daltonics.

## Author(s)

Sebastian Gibb <mail@sebastiangibb.de>

## References

https://github.com/sgibb/readBrukerFlexData

cpSpecHpcMzXml *Mass spectrum generated by Bruker Daltonics CompassXport*

## Description

This dataset was generated by Bruker Daltonics CompassXport and imported by readMzXmlFile to R. It is only needed for comparison between Bruker Daltonics' HPC and .hpc.

## Format

A list containing a mass and an intensity vector.

## Source

Examples/hpc/mzXML/hpc.mzXML

## See Also

.hpc, readMzXmlFile

readBrukerFlexDir *Reads recursively mass spectrometry data in Bruker Daltonics XMASS format.*

## Description

This function leads recursively all mass spectrometry data in Bruker Daltonics XMASS format in a specified directory.

## Usage

```
readBrukerFlexDir(
  brukerFlexDir,
  removeCalibrationScans = TRUE,
  removeMetaData = FALSE,
  useHpc = TRUE,
  useSpectraNames = TRUE,
  filterZeroIntensities = FALSE,
  verbose = FALSE
)
```

## Arguments

| | |
|---|---|
| brukerFlexDir | character, path to *directory* which should be read recursively. |
| removeCalibrationScans | logical, if TRUE all scans in directories called [Cc]alibration will be ignored. |
| removeMetaData | logical, to calculate mass data a lot of meta data are needed. To save memory they could be deleted after calculation. |
| useHpc | logical, should Bruker Daltonics' High Precision Calibration be used if available? (see also: .hpc) |
| useSpectraNames | logical, if TRUE all list elements get an unique name from metaData otherwise file path is used. (If 'removeMetaData' is TRUE 'useSpectraNames' has no effect.) |
| filterZeroIntensities | logical, don't change it. If TRUE all intensities equal 0.0 are removed. (see also: readBrukerFlexFile) |
| verbose | logical, print verbose messages? |

## Details

See readBrukerFlexFile.

## Value

A `list` of spectra.

- `[[1]]$spectrum$mass`: A vector of calculated mass.
- `[[1]]$spectrum$intensity`: A vector of intensity values.
- `[[1]]$metaData`: A list of metaData depending on read spectrum.

## See Also

`importBrukerFlex`, `readBrukerFlexFile`, `.hpc`

## Examples

```
## load library
library("readBrukerFlexData")

## get examples directory
exampleDirectory <- system.file("Examples", package="readBrukerFlexData")

## read example spectra
spec <- readBrukerFlexDir(file.path(exampleDirectory,
  "2010_05_19_Gibb_C8_A1"))

## plot spectra
plot(spec[[1]]$spectrum$mass, spec[[1]]$spectrum$intensity, type="n")

l <- length(spec)
legendStr <- character(l)
for (i in seq(along=spec)) {
  lines(spec[[i]]$spectrum$mass, spec[[i]]$spectrum$intensity, type="l",
        col=rainbow(l)[i])
  legendStr[i] <- spec[[i]]$metaData$fullName
}

## draw legend
legend(x="topright", legend=legendStr, col=rainbow(l), lwd=1)
```

---

# readBrukerFlexFile *Reads mass spectrometry data in Bruker Daltonics XMASS format.*

---

## Description

This function reads mass spectrometry data in Bruker Daltonics XMASS format used by Bruker Daltonics mass spectrometer of *flex series (autoflex, microflex, ultraflex).

## Usage

```
readBrukerFlexFile(
  fidFile,
  removeMetaData = FALSE,
  useHpc = TRUE,
  filterZeroIntensities = FALSE,
  keepNegativeIntensities = FALSE,
  verbose = FALSE
)
```

## Arguments

| | |
|---|---|
| `fidFile` | `character`, path to *fid* file which should be read. |
| `removeMetaData,` | `logical`, to calculate mass data a lot of meta data are needed. To save memory they could be deleted after calculation. |
| `useHpc` | `logical`, should Bruker Daltonics' High Precision Calibration be used if available? (see also: `.hpc`) |
| `filterZeroIntensities` | `logical`, don't change it. If `TRUE` all intensities equal `0.0` are removed. (see also: 'Details' section) |
| `keepNegativeIntensities` | `logical`, don't change it. If `FALSE` all intensities less than zero are replaced by zero. (see also: 'Details' section) |
| `verbose` | `logical`, print verbose messages? |

## Details

`readBrukerFlexFile` has to import the following data to calculating mass from *acqu* file:

| acqu-value | becomes metaData | description |
|---|---|---|
| $BYTORDA | metaData$byteOrder | endianness of fid file |
| $TD | metaData$number | total number of measured time periods |
| $DELAY | metaData$timeDelay | first measured intensity after *metaData$timeDelay* ns |
| $DW | metaData$timeDelta | ns between measured time periods |
| $ML1 | metaData$calibrationConstants[1] | mass calibration constant |
| $ML2 | metaData$calibrationConstants[2] | mass calibration constant |
| $ML3 | metaData$calibrationConstants[3] | mass calibration constant |

If High Precision Calibration (HPC) is used, `readBrukerFlexFile` needs:

| acqu-value | becomes metaData | description |
|---|---|---|
| $HPClBHi | metaData$hpc$limits["maxMass"] | upper mass threshold |
| $HPClBLo | metaData$hpc$limits["minMass"] | lower mass threshold |
| $HPClOrd | metaData$hpc$order | polynomial order |

| | | |
|---|---|---|
| $HPClUse | metaData$hpc$use | maybe using of HPC? (seems to be always “yes” in our test data) |
| $HPCStr | metaData$hpc$coefficients | polynomial coefficients in a string |

`readBrukerFlexFile` tries also to import [optional]:

| acqu-value | becomes metaData | description |
|---|---|---|
| DATATYPE | metaData$dataType | e.g CONTINUOUS MASS |
| SPECTROMETER/DATASYSTEM | metaData$dataSystem | e.g. Bruker Flex Series |
| .SPECTROMETER TYPE | metaData$spectrometerType | e.g. TOF |
| .INLET | metaData$inlet | DIRECT |
| .IONIZATION MODE | metaData$ionizationMode | e.g. LD+ |
| $DATE | metaData$date | same as $AQ_DATE but of |
| $ACQMETH | metaData$acquisitionMethod | path to method file |
| $AQ_DATE | metaData$acquisitionDate | acquisition date |
| $AQ_mod | metaData$acquisitionMode | acquisition mode |
| $AQOP_m | metaData$acquisitionOperatorMode, metaData$tofMode | LINEAR / REFLECTOR |
| $ATTEN | metaData$laserAttenuation | laser beam attenuation |
| $CMT[1:4] | metaData$comments | comments |
| $DEFLON | metaData$deflection | deflection ON/OFF |
| $DIGTYP | metaData$digitizerType | type of digitizer |
| $DPCAL1 | metaData$deflectionPulserCal1 | deflection pulser cal 1 |
| $DPMASS | metaData$deflectionPulserMass | deflection pulser mass |
| $FCVer | metaData$flexControlVersion | Version of Bruker Daltonic |
| $ID_raw | metaData$id | spectrum id |
| $INSTRUM | metaData$instrument | e.g. AUTOFLEX |
| $InstrID | metaData$instrumentId | ID of mass spectrometer |
| $InstTyp | metaData$instrumentType | instrument type |
| $Lift1 | metaData$lift[1] | LIFT constant? |
| $Lift2 | metaData$lift[2] | LIFT constant? |
| $Masserr | metaData$massError | initial mass error in ppm |
| $NoSHOTS | metaData$laserShots | number of applied laser sh |
| $PATCHNO | metaData$patch | sample postion on target |
| $PATH | metaData$path | original file path (on Bruke |
| $REPHZ | metaData$laserRepetition | laser repetition rate in Hz |
| $SPOTNO | metaData$spot | same as *$PATCHNO* (in ol |
| $SPType | metaData$spectrumType | e.g. TOF |
| $TgIDS | metaData$target$id | target ids |
| $TgCount | metaData$target$count | number of measurements v |
| $TgSer | metaData$target$serialNumber | target serial number |
| $TgTyp | metaData$target$typeNumber | target type number |
| $TLift | metaData$tlift | LIFT constant? |

import from file path:

| value | becomes metaData | description |
|---|---|---|

| | | |
|---|---|---|
| full current path to fid file | metaData$file | path on local machine |
| sample name | metaData$sampleName | - |

`filterZeroIntensities`: Change default value is **not recommended**! If `TRUE` all intensities equal zero are removed. This parameter exists only to be compatible to Bruker Daltonics CompassXport's mzXML export function. For details see: 'Release Notes for CompassXport 3.0.3', cap. 6 'Filtering of Zero Intensities': "Bruker Daltonics' Acquisition Software will compress Analysis raw data. To save on operation time and to keep export file sizes small, CompassXport 3.0.3 will filter out zero (0.0) intensities when exporting to mzXML or mzData ..."

`keepNegativeIntensities`: Change default value is **not recommended**! If `TRUE` negative intensity values are not replaced by zero. This parameter exists only to be compatible to Bruker Daltonics CompassXport.

## Value

A `list` of spectra and metadata.

- `spectrum$mass`: A vector of calculated mass.
- `spectrum$tof`: A vector of time-of-flight data.
- `spectrum$intensity`: A vector of intensity values.
- `metaData`: A list of metaData depending on read spectrum.

## See Also

https://github.com/sgibb/readBrukerFlexData/wiki, importBrukerFlex, readBrukerFlexDir, .hpc

## Examples

```
## load library
library("readBrukerFlexData")

## get examples directory
exampleDirectory <- system.file("Examples", package="readBrukerFlexData")

## read example spectrum
spec <- readBrukerFlexFile(file.path(exampleDirectory,
  "2010_05_19_Gibb_C8_A1/0_A1/1/1SLin/fid"))

## print metaData
print(spec$metaData)

## plot spectrum
plot(spec$spectrum$mass, spec$spectrum$intensity, type="l", col="red")
```

# Index

∗ **IO**
    readBrukerFlexDir, 3
    readBrukerFlexFile, 4
∗ **datasets**
    cpSpecHpcMzXml, 2
∗ **package**
    readBrukerFlexData-package, 2
.hpc, *2–5*, *7*

cpSpecHpcMzXml, 2

importBrukerFlex, *4*, *7*

readBrukerFlexData-package, 2
readBrukerFlexDir, 3, *7*
readBrukerFlexFile, *3*, *4*, 4
readMzXmlFile, *2*