

Package ‘timsr’

May 11, 2021

Title Easily Access timsTOF Data

Version 0.0.3

Maintainer Mateusz Krzysztof Łacki <matteo.lacki@gmail.com>

Description Access 'timsTOF' mass spectrometry data, as described <https://sso.bruker.com/auth/realms/bruker/protocol/openid-connect/auth?client_id=aem-bruker.com&redirect_uri=https%3A%2F%2Fwww.bruker.com%2Fen.login.html%3FtargetUrl%3Dhttps%3A%2F%2Fwww.downloads%2Fmass-spectrometry.html&response_type=id_token%20token&scope=openid%20profile&state=4f9d225e92f341cca3b03a555330ter%20registering>, using the 'OpenTIMS' C++ reader and save all into 'data.tables'.

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.1

Depends R (>= 3.0.0)

Imports opentimsr, data.table, methods

NeedsCompilation no

Author Mateusz Krzysztof Łacki [aut, cre, cph]
(<<https://orcid.org/0000-0001-7415-4748>>),
Michał Piotr Startek [aut, cph]
(<<https://orcid.org/0000-0001-5227-3447>>)

Repository CRAN

Date/Publication 2021-05-11 17:30:03 UTC

R topics documented:

cleanMem	2
download_bruker_proprietary_code	3
intensity_per_frame	4
min_max_measurements	4
MS1	5
peaks_per_frame_cnts	6
plot_TIC	6

query	7
retention_times	8
rt_query	8
setup_braker_so	9
table2dt	10
tables_names	10
tdf.tables	11
TimsR	11
TimsR-class	12
[,TimsR,ANY,ANY-method	12
[,TimsR,ANY,character-method	13

Index 14

cleanMem	<i>Clean memory.</i>
----------	----------------------

Description

Run garbage collection, by default - 10 times.

Usage

```
cleanMem(times = 10)
```

Arguments

times	Number of times to run garbage collection.
-------	--

Details

Check <https://stackoverflow.com/questions/1467201/forcing-garbage-collection-to-run-in-r-with-the-g>

Value

No return value, called for side effects.

Examples

```
## Not run:
cleanMem()

## End(Not run)
```

`download_bruker_proprietary_code`

Get Bruker's code needed for running proprietary time of flight to mass over charge and scan to drift time conversion.

Description

By using this function you agree to terms of license precised in "https://github.com/MatteoLacki/opentims_bruker_bridge". The conversion, due to independent code-base restrictions, are possible only on Linux and Windows operating systems. Works on full open-source solution are on the way.

Usage

```
download_bruker_proprietary_code(  
    target.folder,  
    net_url = paste0("https://github.com/MatteoLacki/opentims_bruker_bridge/",  
        "raw/main/opentims_bruker_bridge/"),  
    mode = "wb",  
    ...  
)
```

Arguments

<code>target.folder</code>	Folder where to store the 'dll' or 'so' file.
<code>net_url</code>	The url with location of all files.
<code>mode</code>	Which mode to use when downloading a file?
<code>...</code>	Other parameters to 'download.file'.

Value

character, path to the output 'timsdata.dll' on Windows and 'libtimsdata.so' on Linux.

Examples

```
## Not run:  
download_bruker_proprietary_code("your/prefered/destination/folder")  
  
## End(Not run)
```

intensity_per_frame *Get sum of intensity per each frame (retention time).*

Description

Get sum of intensity per each frame (retention time).

Usage

```
intensity_per_frame(timsr, recalibrated = TRUE)
```

Arguments

timsr	Instance of TimsR
recalibrated	Use Bruker recalibrated total intensities or calculate them from scratch with OpenTIMS?

Value

integer vector: total intensity per each frame.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(intensity_per_frame(D))  
print(intensity_per_frame(D, recalibrated=FALSE))  
  
## End(Not run)
```

min_max_measurements *Get border values for measurements.*

Description

Get the min-max values of the measured variables (except for TOFs, that would require iteration through data rather than parsing metadata).

Usage

```
min_max_measurements(timsr)
```

Arguments

timsr	Instance of TimsR.
-------	--------------------

Value

data.frame Limits of individual extracted quantities.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
min_max_measurements(D) # this gives a small data-frame with min and max values.  
  
## End(Not run)
```

MS1

Get MS1 frame numbers.

Description

Get MS1 frame numbers.

Usage

```
MS1(timsr)
```

Arguments

timsr Instance of TimsR

Value

Numbers of frames corresponding to MS1, i.e. precursor ions.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(MS1(D))  
  
## End(Not run)
```

peaks_per_frame_cnts *Get the number of peaks per frame.*

Description

Get the number of peaks per frame.

Usage

```
peaks_per_frame_cnts(timsr)
```

Arguments

timsr Instance of TimsR.

Value

Number of peaks in each frame.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(peaks_per_frame_cnts(D))  
  
## End(Not run)
```

plot_TIC *Plot intensity per retention time.*

Description

Plot will split 'MS1' and 'MS2'.

Usage

```
plot_TIC(timsr, recalibrated = TRUE)
```

Arguments

timsr Instance of TimsR
recalibrated Use Bruker recalibrated total intensities or calculate them from scratch with OpenTIMS?

Value

No return value, called for side effects.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
plot_TIC(D)  
plot_TIC(D, recalibrated=FALSE)  
  
## End(Not run)
```

query

Query for raw data.

Description

Get the raw data from Bruker's 'tdf_bin' format. Defaults to both raw data ('frame', 'scan', 'tof', 'intensity') and its transformations into physical units ('mz', 'inv_ion_mobility', 'retention_time').

Usage

```
query(timsr, frames, columns = all_columns)
```

Arguments

timsr	Instance of TimsR.
frames	Vector of frame numbers to extract.
columns	Vector of columns to extract. Defaults to all columns.

Value

data.frame with selected columns.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(query(D, c(1,20, 53)) # extract all columns  
print(query(D, c(1,20, 53), columns=c('scan','intensity')) # only 'scan' and 'intensity'  
  
## End(Not run)
```

retention_times	<i>Get the retention time for each frame.</i>
-----------------	---

Description

Get the retention time for each frame.

Usage

```
retention_times(timsr)
```

Arguments

timsr	Instance of TimsR.
-------	--------------------

Value

Retention times corresponding to each frame.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(retention_times(D))  
  
## End(Not run)
```

rt_query	<i>Get the retention time for each frame.</i>
----------	---

Description

Get the retention time for each frame.

Usage

```
rt_query(timsr, min_retention_time, max_retention_time, columns = all_columns)
```

Arguments

timsr	Instance of TimsR
min_retention_time	Lower boundry on retention time.
max_retention_time	Upper boundry on retention time.
columns	Vector of columns to extract. Defaults to all columns.

Value

data.frame with selected columns.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(query_slice(D, 10, 200, 4)) # extract every fourth frame between 10 and 200.  
print(query_slice(D, 10, 200, 4, columns=c('scan','intensity')) # only 'scan' and 'intensity'  
  
## End(Not run)
```

setup Bruker SO	<i>Dynamically link Bruker's DLL to enable tof-mz and scan- inv_ion_mobility conversion.</i>
-----------------	--

Description

By using this function you agree to terms of license precised in "<https://github.com/MatteoLacki/opentims Bruker bridge>". The conversion, due to independent code-base restrictions, are possible only on Linux and Windows operating systems. Works on full open-source solution are on the way.

Usage

```
setup Bruker SO(path)
```

Arguments

path	Path to the 'libtimsdata.so' on Linux or 'timsdata.dll' on Windows, as produced by 'download Bruker proprietary code'.
------	--

Value

No return value, called for side effects.

Examples

```
## Not run:  
so_path = download Bruker proprietary code("your/prefered/destination/folder")  
setup Bruker SO(so_path)  
  
## End(Not run)
```

table2dt	<i>Extract tables from sqlite database analysis.tdf.</i>
----------	--

Description

Export a table from sqlite.

Usage

```
table2dt(timsr, names)
```

Arguments

timsr	Instance of TimsR
names	Names to extract from the sqlite database.

Value

data.table

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(head(table2dt(D, "Frames"))) # Extract table "Frames".  
  
## End(Not run)
```

tables_names	<i>Extract tables from sqlite database analysis.tdf.</i>
--------------	--

Description

Extract tables from sqlite database analysis.tdf.

Usage

```
tables_names(timsr)
```

Arguments

timsr	Instance of TimsR
-------	-------------------

Value

character, names of tables.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(tables_names(D))  
  
## End(Not run)
```

tdf.tables	<i>Explore the contents of the sqlite .tdf database.</i>
------------	--

Description

Explore the contents of the sqlite .tdf database.

Usage

```
tdf.tables(timsr)
```

Arguments

timsr	Instance of TimsR
-------	-------------------

Value

List of data.tables filled with data from 'analysis.tdf'.

Examples

```
## Not run:  
D = TimsR('path/to/your/folder.d')  
print(tdf.tables(D))  
  
## End(Not run)
```

TimsR	<i>Get TimsR data accessor.</i>
-------	---------------------------------

Description

Get TimsR data accessor.

Usage

```
TimsR(path.d)
```

Arguments

path.d Path to the TimsTOF '*.d' folder containing the data (requires the folder to contain only 'analysis.tdf' and 'analysis.tdf_bin').

Value

instance of TimsR class that represents raw data

Examples

```
## Not run:
D = TimsR(path_to_.d_folder)
D[1] # First frame.

## End(Not run)
```

TimsR-class

Advanced TimsTOF data accessor.

Description

S4 class that facilitates data queries for TimsTOF data.

[,TimsR,ANY,ANY-method

Get some frames of data.

Description

Get some frames of data.

Usage

```
## S4 method for signature 'TimsR,ANY,ANY'
x[i]
```

Arguments

x OpenTIMS data instance.
i An array of nonzero indices to extract.

Value

data.table

Examples

```
## Not run:
D = TimsR('path/to/your/folder.d')
print(head(D[10]))
print(head(D[10:100]))

## End(Not run)
```

[,TimsR,ANY,character-method
Get some frames of data.

Description

Get some frames of data.

Usage

```
## S4 method for signature 'TimsR,ANY,character'
x[i, j]
```

Arguments

x OpenTIMS data instance.
i An array of nonzero frame numbers to extract.
j A vector of strings with column names to extract.

Value

data.table

Examples

```
## Not run:
D = TimsR('path/to/your/folder.d')
all_cols = c('frame','scan','tof','intensity',
             'mz','inv_ion_mobility','retention_time')
print(D[c(1,20, 53), all_cols])
# extracted all columns

print(D[c(1,20, 53), c('scan','intensity')])
# only 'scan' and 'intensity'

## End(Not run)
```

Index

[,TimsR,ANY,ANY-method, [12](#)
[,TimsR,ANY,character-method, [13](#)

cleanMem, [2](#)

download_bruker_proprietary_code, [3](#)

intensity_per_frame, [4](#)

min_max_measurements, [4](#)
MS1, [5](#)

peaks_per_frame_cnts, [6](#)
plot_TIC, [6](#)

query, [7](#)

retention_times, [8](#)
rt_query, [8](#)

setup_bruker_so, [9](#)

table2dt, [10](#)
tables_names, [10](#)
tdf.tables, [11](#)
TimsR, [11](#)
TimsR-class, [12](#)