# Package 'RaMS'

March 22, 2021

**Type** Package

**Title** R Access to Mass-Spec Data

**Version** 1.0.0

**Maintainer** William Kumler <wkumler@uw.edu>

**Description** R-based access to mass-spectrometry (MS) data. While many packages exist to process MS data, many of these make it difficult to access the underlying mass-to-charge ratio (m/z), intensity, and retention time of the files themselves. This package is designed to format MS data in a tidy fashion and allows the user perform the plotting and analysis.

**License** MIT + file LICENSE

**URL** https://github.com/wkumler/RaMS

**BugReports** https://github.com/wkumler/RaMS/issues

**Encoding** UTF-8

**LazyData** true

**Imports** xml2, base64enc, data.table, utils

**RoxygenNote** 7.1.1

**Suggests** testthat, knitr, rmarkdown, tidyverse, plotly, openxlsx, DBI, RSQLite, reticulate

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** William Kumler [aut, cre, cph]

**Repository** CRAN

**Date/Publication** 2021-03-22 16:00:02 UTC

## R topics documented:

checkOutputQuality . . . . . 2
grabMSdata . . . . . 3

grabMzmlBPC . . . 4
grabMzmlData . . . 5
grabMzmlEncodingData . . . 7
grabMzmlMetadata . . . 7
grabMzmlMS1 . . . 8
grabMzmlMS2 . . . 8
grabMzxmlBPC . . . 9
grabMzxmlData . . . 9
grabMzxmlEncodingData . . . 11
grabMzxmlMetadata . . . 11
grabMzxmlMS1 . . . 12
grabMzxmlMS2 . . . 12
grabMzxmlSpectraMzInt . . . 13
grabMzxmlSpectraPremz . . . 14
grabMzxmlSpectraRt . . . 14
grabMzxmlSpectraVoltage . . . 15
grabSpectraInt . . . 15
grabSpectraMz . . . 16
grabSpectraPremz . . . 16
grabSpectraRt . . . 17
grabSpectraVoltage . . . 17
pmppm . . . 18

**Index** **19**

---

`checkOutputQuality` *Check that the output data is properly formatted.*

---

## Description

This function checks that data produced by repeated calls to the 'grabMzmlData()' and 'grabMzxmlData()' functions is formatted properly before it's provided to the user. It checks that all of the requested data has been obtained and warns if data is found to be empty, misnamed, or has columns of the wrong type.

## Usage

```
checkOutputQuality(output_data, grab_what)
```

## Arguments

| | |
|---|---|
| `output_data` | The collected data resulting from repeated calls to 'grabMzmlData()', after being bound together. |
| `grab_what` | The names of the data requested by the user. |

## Value

NULL (invisibly). The goal of this function is its side effects, i.e. throwing errors and providing info when the files are not found.

grabMSdata *Grab mass-spectrometry data from file(s)*

## Description

The main ‘RaMS‘ function. This function accepts a list of the files that will be read into R’s working memory and returns a list of ‘data.table‘s containing the requested information. What information is requested is determined by the ‘grab_what‘ argument, which can include MS1, MS2, BPC, TIC, or metadata information. This function serves as a wrapper around both ‘grabMzmlData‘ and ‘grabMzxmlData‘ and handles multiple files, but those two have also been exposed to the user in case super-simple handling is desired. Retention times are reported in minutes, and will be converted automatically if they are encoded in seconds.

## Usage

```
grabMSdata(
  files,
  grab_what = "everything",
  verbosity = NULL,
  mz = NULL,
  ppm = NULL,
  rtrange = NULL
)
```

## Arguments

| | |
|---|---|
| files | A character vector of filenames to read into R’s memory. Both absolute and relative paths are acceptable. |
| grab_what | What data should be read from the file? Options include "MS1" for data only from the first spectrometer, "MS2" for fragmentation data, "BPC" for rapid access to the base peak chromatogram, and "TIC" for rapid access to the total ion chromatogram. These options can be combined (i.e. ‘grab_data=c("MS1", "MS2", "BPC")‘) or this argument can be set to "everything" to extract all of the above. Options "EIC" and "EIC_MS2" are useful when working with files whose total size exceeds working memory - they first extracts all relevant MS1 and MS2 data, then discard data outside of the mass range(s) calculated from the provided mz and ppm. |
| verbosity | Three levels of processing output to the R console are available, with increasing verbosity corresponding to higher integers. A verbosity of zero means that no output will be produced, useful when wrapping within larger functions. A verbosity of 1 will produce a progress bar using base R’s txtProgressBar function. A verbosity of 2 or higher will produce timing output for each individual file read in. The default, NULL, will select between 1 and 2 depending on the number of files being read: if a single file, verbosity is set to 2; if multiple files, verbosity is set to 1. |

| | |
|---|---|
| mz | A vector of the mass-to-charge ratio for compounds of interest. Only used when combined with ‘grab_what = "EIC"‘ (see above). Multiple masses can be provided. |
| ppm | A single number corresponding to the mass accuracy (in parts per million) of the instrument on which the data was collected. Only used when combined with ‘grab_what = "EIC"‘ (see above). |
| rtrange | Only available when parsing mzML files. A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object’s size. |

## Value

A list of ‘data.table‘s, each named after the arguments requested in grab_what. $MS1 contains MS1 information, MS2 contains fragmentation info, etc. MS1 data has four columns: retention time (rt), mass-to-charge (mz), intensity (int), and filename. MS2 data has six: retention time (rt), precursor m/z (premz), fragment m/z (fragmz), fragment intensity (int), collision energy (voltage), and filename. Data requested that does not exist in the provided files (such as MS2 data requested from MS1-only files) will return an empty (length zero) data.table. The data.tables extracted from each of the individual files are collected into one large table using data.table’s ‘rbindlist‘.

## Examples

```
library(RaMS)
# Extract MS1 data from a couple files
sample_dir <- system.file("extdata", package = "RaMS")
sample_files <- list.files(sample_dir, full.names=TRUE)
multifile_data <- grabMSdata(sample_files[2:4], grab_what="MS1")

# "Stream" data from the internet (i.e. Metabolights)
## Not run:
access_url <- "https://www.ebi.ac.uk/metabolights/MTBLS703/files"

# URL below obtained by right-clicking site download button and copying
# link address
sample_url <- paste0("https://www.ebi.ac.uk/metabolights/ws/studies/",
                     "MTBLS703/download/acefcd61-a634-4f35-9c3c-c572",
                     "ade5acf3?file=161024_Smp_LB12HL_AB_pos.mzXML")
file_data <- grabMSdata(sample_url, grab_what="everything", verbosity=2)

## End(Not run)
```

---

# grabMzmlBPC *Grab the BPC or TIC from a file*

---

## Description

The base peak intensity and total ion current are actually written into the mzML files and aren’t encoded, making retrieval of BPC and TIC information blazingly fast if parsed correctly.

### Usage

```
grabMzmlBPC(xml_data, rtrange, TIC = FALSE)
```

### Arguments

| | |
|---|---|
| xml_data | An ‘xml2‘ nodeset, usually created by applying ‘read_xml‘ to an mzML file. |
| rtrange | A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object's size. |
| TIC | Boolean. If TRUE, the TIC is extracted rather than the BPC. |

### Value

A ‘data.table‘ with columns for retention time (rt), and intensity (int).

---

## grabMzmlData *Get mass-spectrometry data from an mzML file*

---

### Description

This function handles the mzML side of things, reading in files that are written in the mzML format. Much of the code is similar to the mzXML format, but the xpath handles are different and the mz/int array is encoded as two separate entries rather than simultaneously. This function has been exposed to the user in case per-file optimization (such as peakpicking or additional filtering) is desired before the full data object is returned.

### Usage

```
grabMzmlData(
  filename,
  grab_what,
  verbosity = 0,
  mz = NULL,
  ppm = NULL,
  rtrange = NULL
)
```

### Arguments

| | |
|---|---|
| filename | A single filename to read into R's memory. Both absolute and relative paths are acceptable. |
| grab_what | What data should be read from the file? Options include "MS1" for data only from the first spectrometer, "MS2" for fragmentation data, "BPC" for rapid access to the base peak chromatogram, and "TIC" for rapid access to the total ion chromatogram. These options can be combined (i.e. ‘grab_data=c("MS1", "MS2", "BPC")‘) or this argument can be set to "everything" to extract all of the |

above. Option "EIC" is useful when working with files whose total size exceeds working memory - it first extracts all relevant MS1 and MS2 data, then discards data outside of the mass range(s) calculated from the provided mz and ppm.

`verbosity` Three levels of processing output to the R console are available, with increasing verbosity corresponding to higher integers. A verbosity of zero means that no output will be produced, useful when wrapping within larger functions. A verbosity of 1 will produce a progress bar using base R's txtProgressBar function. A verbosity of 2 or higher will produce timing output for each individual file read in.

`mz` A vector of the mass-to-charge ratio for compounds of interest. Only used when combined with 'grab_what = "EIC"' (see above). Multiple masses can be provided.

`ppm` A single number corresponding to the mass accuracy (in parts per million) of the instrument on which the data was collected. Only used when combined with 'grab_what = "EIC"' (see above).

`rtrange` A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object's size.

## Value

A list of 'data.table's, each named after the arguments requested in grab_what. $MS1 contains MS1 information, $MS2 contains fragmentation info, etc. MS1 data has three columns: retention time (rt), mass-to-charge (mz), and intensity (int). MS2 data has five: retention time (rt), precursor m/z (premz), fragment m/z (fragmz), fragment intensity (int), and collision energy (voltage). Data requested that does not exist in the provided files (such as MS2 data requested from MS1-only files) will return an empty (length zero) data.table.

## Examples

```
sample_file <- system.file("extdata", "LB12HL_AB.mzML.gz", package = "RaMS")
file_data <- grabMzmlData(sample_file, grab_what="MS1")
# Extract MS1 data and a base peak chromatogram
file_data <- grabMzmlData(sample_file, grab_what=c("MS1", "BPC"))
# Extract data from a retention time subset
file_data <- grabMzmlData(sample_file, grab_what=c("MS1", "BPC"),
                          rtrange=c(5, 7))
# Extract EIC for a specific mass
file_data <- grabMzmlData(sample_file, grab_what="EIC", mz=118.0865, ppm=5)
# Extract EIC for several masses simultaneously
file_data <- grabMzmlData(sample_file, grab_what="EIC", ppm=5,
                          mz=c(118.0865, 146.118104, 189.123918))

# Extract MS2 data
sample_file <- system.file("extdata", "DDApos_2.mzML.gz", package = "RaMS")
MS2_data <- grabMzmlData(sample_file, grab_what="MS2")
```

## grabMzmlEncodingData *Helper function to extract mzML file encoding data*

### Description

Helper function to extract mzML file encoding data

### Usage

```
grabMzmlEncodingData(xml_data)
```

### Arguments

| | |
|---|---|
| xml_data | mzML data as parsed by xml2 |

### Value

A list of values used by other parsing functions, currently compression, mz_precision, int_precision

## grabMzmlMetadata *Helper function to extract mzML file metadata*

### Description

Helper function to extract mzML file metadata

### Usage

```
grabMzmlMetadata(xml_data)
```

### Arguments

| | |
|---|---|
| xml_data | mzML data as parsed by xml2 |

### Value

A list of values corresponding to various pieces of metadata for each file

grabMzmlMS1 *Extract the MS1 data from an mzML nodeset*

## Description

Extract the MS1 data from an mzML nodeset

## Usage

```
grabMzmlMS1(xml_data, rtrange, file_metadata)
```

## Arguments

| | |
|---|---|
| xml_data | An ‘xml2‘ nodeset, usually created by applying ‘read_xml‘ to an mzML file. |
| rtrange | A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object’s size. |
| file_metadata | Information about the file used to decode the binary arrays containing m/z and intensity information. |

## Value

A ‘data.table‘ with columns for retention time (rt), m/z (mz), and intensity (int).

grabMzmlMS2 *Extract the MS2 data from an mzML nodeset*

## Description

Extract the MS2 data from an mzML nodeset

## Usage

```
grabMzmlMS2(xml_data, rtrange, file_metadata)
```

## Arguments

| | |
|---|---|
| xml_data | An ‘xml2‘ nodeset, usually created by applying ‘read_xml‘ to an mzML file. |
| rtrange | A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object’s size. |
| file_metadata | Information about the file used to decode the binary arrays containing m/z and intensity information. |

## Value

A ‘data.table‘ with columns for retention time (rt), precursor m/z (mz), fragment m/z (fragmz), collision energy (voltage), and intensity (int).

---

`grabMzxmlBPC` *Grab the BPC or TIC from a file*

---

## Description

The base peak intensity and total ion current are actually written into the mzXML files and aren't encoded, making retrieval of BPC and TIC information blazingly fast if parsed correctly.

## Usage

```
grabMzxmlBPC(xml_data, TIC = FALSE, rtrange)
```

## Arguments

| | |
|---|---|
| `xml_data` | An ‘xml2‘ nodeset, usually created by applying ‘read_xml‘ to an mzML file. |
| `TIC` | Boolean. If TRUE, the TIC is extracted rather than the BPC. |
| `rtrange` | A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object's size. |

## Value

A ‘data.table‘ with columns for retention time (rt), and intensity (int).

---

`grabMzxmlData` *Get mass-spectrometry data from an mzXML file*

---

## Description

This function handles the mzXML side of things, reading in files that are written in the mzXML format. Much of the code is similar to the mzXML format, but the xpath handles are different and the mz/int array is encoded simultaneously rather than as two separate entries. This function has been exposed to the user in case per-file optimization (such as peakpicking or additional filtering) is desired before the full data object is returned.

## Usage

```
grabMzxmlData(
  filename,
  grab_what,
  verbosity = 0,
  rtrange = NULL,
  mz = NULL,
  ppm = NULL
)
```

## Arguments

| | |
|---|---|
| filename | A single filename to read into R's memory. Both absolute and relative paths are acceptable. |
| grab_what | What data should be read from the file? Options include "MS1" for data only from the first spectrometer, "MS2" for fragmentation data, "BPC" for rapid access to the base peak chromatogram, and "TIC" for rapid access to the total ion chromatogram. These options can be combined (i.e. 'grab_data=c("MS1", "MS2", "BPC")') or this argument can be set to "everything" to extract all of the above. Option "EIC" is useful when working with files whose total size exceeds working memory - it first extracts all relevant MS1 and MS2 data, then discards data outside of the mass range(s) calculated from the provided mz and ppm. |
| verbosity | Three levels of processing output to the R console are available, with increasing verbosity corresponding to higher integers. A verbosity of zero means that no output will be produced, useful when wrapping within larger functions. A verbosity of 1 will produce a progress bar using base R's txtProgressBar function. A verbosity of 2 or higher will produce timing output for each individual file read in. |
| rtrange | Not supported for mzXML data. Only provided here so as to throw a friendly warning rather than an unexpected error. |
| mz | A vector of the mass-to-charge ratio for compounds of interest. Only used when combined with 'grab_what = "EIC"' (see above). Multiple masses can be provided. |
| ppm | A single number corresponding to the mass accuracy (in parts per million) of the instrument on which the data was collected. Only used when combined with 'grab_what = "EIC"' (see above). |

## Value

A list of 'data.table's, each named after the arguments requested in grab_what. $MS1 contains MS1 information, $MS2 contains fragmentation info, etc. MS1 data has three columns: retention time (rt), mass-to-charge (mz), and intensity (int). MS2 data has five: retention time (rt), precursor m/z (premz), fragment m/z (fragmz), fragment intensity (int), and collision energy (voltage). Data requested that does not exist in the provided files (such as MS2 data requested from MS1-only files) will return an empty (length zero) data.table.

## Examples

```
sample_file <- system.file("extdata", "LB12HL_AB.mzXML.gz", package = "RaMS")
file_data <- grabMzxmlData(sample_file, grab_what="MS1")
# Extract MS1 data and a base peak chromatogram
file_data <- grabMzxmlData(sample_file, grab_what=c("MS1", "BPC"))
# Extract EIC for a specific mass
file_data <- grabMzxmlData(sample_file, grab_what="EIC", mz=118.0865, ppm=5)
# Extract EIC for several masses simultaneously
file_data <- grabMzxmlData(sample_file, grab_what="EIC", ppm=5,
                           mz=c(118.0865, 146.118104, 189.123918))

# Extract MS2 data
sample_file <- system.file("extdata", "DDApos_2.mzXML.gz", package = "RaMS")
MS2_data <- grabMzxmlData(sample_file, grab_what="MS2")
```

---

# grabMzxmlEncodingData *Helper function to extract mzXML file metadata*

---

## Description

Helper function to extract mzXML file metadata

## Usage

```
grabMzxmlEncodingData(xml_data)
```

## Arguments

| | |
|---|---|
| xml_data | mzXML data as parsed by xml2 |

## Value

A list of values used by other parsing functions, currently compression, precision, and endian encoding (endi_enc)

---

# grabMzxmlMetadata *Helper function to extract mzXML file metadata*

---

## Description

Helper function to extract mzXML file metadata

## Usage

```
grabMzxmlMetadata(xml_data)
```

### Arguments

| | |
|---|---|
| xml_data | mzXML data as parsed by xml2 |

### Value

A list of values corresponding to various pieces of metadata for each file

---

## grabMzxmlMS1 *Extract the MS1 data from an mzXML nodeset*

---

### Description

Extract the MS1 data from an mzXML nodeset

### Usage

```
grabMzxmlMS1(xml_data, file_metadata, rtrange)
```

### Arguments

| | |
|---|---|
| xml_data | An ‘xml2‘ nodeset, usually created by applying ‘read_xml‘ to an mzXML file. |
| file_metadata | Information about the file used to decode the binary arrays containing m/z and intensity information. |
| rtrange | A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object’s size. |

### Value

A ‘data.table‘ with columns for retention time (rt), m/z (mz), and intensity (int).

---

## grabMzxmlMS2 *Extract the MS2 data from an mzXML nodeset*

---

### Description

Extract the MS2 data from an mzXML nodeset

### Usage

```
grabMzxmlMS2(xml_data, file_metadata, rtrange)
```

### Arguments

| | |
|---|---|
| xml_data | An 'xml2' nodeset, usually created by applying 'read_xml' to an mzXML file. |
| file_metadata | Information about the file used to decode the binary arrays containing m/z and intensity information. |
| rtrange | A vector of length 2 containing an upper and lower bound on retention times of interest. Providing a range here can speed up load times (although not enormously, as the entire file must still be read) and reduce the final object's size. |

### Value

A 'data.table' with columns for retention time (rt), precursor m/z (mz), fragment m/z (fragmz), collision energy (voltage), and intensity (int).

---

## grabMzxmlSpectraMzInt *Extract the mass-to-charge data from the spectra of an mzXML nodeset*

---

### Description

The mz and intensity information of mzXML files are encoded as a binary array, sometimes compressed via gzip or zlib or numpress. This code finds all the m/z-int binary arrays and converts them back to the original measurements. See https://github.com/ProteoWizard/pwiz/issues/1301

### Usage

```
grabMzxmlSpectraMzInt(xml_nodes, file_metadata)
```

### Arguments

| | |
|---|---|
| xml_nodes | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |
| file_metadata | Information about the file used to decode the binary arrays containing m/z and intensity information. Here, the compression and mz precision information is relevant. |

### Value

A numeric vector of masses, many for each scan.

## grabMzxmlSpectraPremz *Extract the precursor mass from the spectra of an mzXML nodeset*

### Description

Extract the precursor mass from the spectra of an mzXML nodeset

### Usage

```
grabMzxmlSpectraPremz(xml_nodes)
```

### Arguments

| | |
|---|---|
| xml_nodes | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |

### Value

A numeric vector of precursor masses, one for each scan

## grabMzxmlSpectraRt *Extract the retention time from the spectra of an mzXML nodeset*

### Description

Extract the retention time from the spectra of an mzXML nodeset

### Usage

```
grabMzxmlSpectraRt(xml_nodes)
```

### Arguments

| | |
|---|---|
| xml_nodes | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |

### Value

A numeric vector of retention times, one for each scan

grabMzxmlSpectraVoltage *Extract the collison energies from the spectra of an mzXML nodeset*

## Description

Although the collision energy is typically fixed per file, it's equally fast (afaik) to just grab them all individually here. Also, I'm worried about these rumors of "ramped" collision energies

## Usage

```
grabMzxmlSpectraVoltage(xml_nodes)
```

## Arguments

| | |
|---|---|
| xml_nodes | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |

## Value

A numeric vector of collision energies, one for each scan.

grabSpectraInt *Extract the intensity information from the spectra of an mzML nodeset*

## Description

The mz and intensity information of mzML files are encoded as binary arrays, sometimes compressed via gzip or zlib or numpress. This code finds all the intensity binary arrays and converts them back to the original measurements. See https://github.com/ProteoWizard/pwiz/issues/1301

## Usage

```
grabSpectraInt(xml_nodes, file_metadata)
```

## Arguments

| | |
|---|---|
| xml_nodes | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |
| file_metadata | Information about the file used to decode the binary arrays containing m/z and intensity information. Here, the compression and int precision information is relevant. |

### Value

A numeric vector of intensities, many for each scan.

---

`grabSpectraMz` *Extract the mass-to-charge data from the spectra of an mzML nodeset*

---

### Description

The mz and intensity information of mzML files are encoded as binary arrays, sometimes compressed via gzip or zlib or numpress. This code finds all the m/z binary arrays and converts them back to the original measurements. See https://github.com/ProteoWizard/pwiz/issues/1301

### Usage

```
grabSpectraMz(xml_nodes, file_metadata)
```

### Arguments

| | |
|---|---|
| `xml_nodes` | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |
| `file_metadata` | Information about the file used to decode the binary arrays containing m/z and intensity information. Here, the compression and mz precision information is relevant. |

### Value

A numeric vector of masses, many for each scan.

---

`grabSpectraPremz` *Extract the precursor mass from the spectra of an mzML nodeset*

---

### Description

Extract the precursor mass from the spectra of an mzML nodeset

### Usage

```
grabSpectraPremz(xml_nodes)
```

### Arguments

| | |
|---|---|
| `xml_nodes` | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying 'xml_find_all' to an MS1 or MS2 nodeset. |

### Value

A numeric vector of precursor masses, one for each scan

---

`grabSpectraRt` *Extract the retention time from the spectra of an mzML nodeset*

---

### Description

Extract the retention time from the spectra of an mzML nodeset

### Usage

```
grabSpectraRt(xml_nodes)
```

### Arguments

| | |
|---|---|
| `xml_nodes` | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying ‘xml_find_all‘ to an MS1 or MS2 nodeset. |

### Value

A numeric vector of retention times, one for each scan

---

`grabSpectraVoltage` *Extract the collison energies from the spectra of an mzML nodeset*

---

### Description

Although the collision energy is typically fixed per file, it’s equally fast (afaik) to just grab them all individually here. Also, I’m worried about these rumors of "ramped" collision energies

### Usage

```
grabSpectraVoltage(xml_nodes)
```

### Arguments

| | |
|---|---|
| `xml_nodes` | An xml_nodeset object corresponding to the spectra collected by the mass spectrometer, usually produced by applying ‘xml_find_all‘ to an MS1 or MS2 nodeset. |

### Value

A numeric vector of collision energies, one for each scan.

pmppm *Plus/minus parts per million*

## Description

It shouldn't be hard to translate a point mass into a mass window bounded by spectrometer accuracy.

## Usage

```
pmppm(mass, ppm = 4)
```

## Arguments

| | |
|---|---|
| mass | A length-1 numeric representing the mass of interest for which a mass range is desired. |
| ppm | The parts-per-million accuracy of the mass spectrometer on which the data was collected. |

## Value

A length-2 numeric representing the mass range requested

## Examples

```
pmppm(100, 5)
pmppm(1000000, 5)
pmppm(118.0865, 2.5)
pmppm(892.535313, 10)
```

# Index

checkOutputQuality, 2

grabMSdata, 3
grabMzmlBPC, 4
grabMzmlData, 5
grabMzmlEncodingData, 7
grabMzmlMetadata, 7
grabMzmlMS1, 8
grabMzmlMS2, 8
grabMzxmlBPC, 9
grabMzxmlData, 9
grabMzxmlEncodingData, 11
grabMzxmlMetadata, 11
grabMzxmlMS1, 12
grabMzxmlMS2, 12
grabMzxmlSpectraMzInt, 13
grabMzxmlSpectraPremz, 14
grabMzxmlSpectraRt, 14
grabMzxmlSpectraVoltage, 15
grabSpectraInt, 15
grabSpectraMz, 16
grabSpectraPremz, 16
grabSpectraRt, 17
grabSpectraVoltage, 17

pmppm, 18