

Package ‘mzID’

October 8, 2014

Type Package

Title An mzIdentML parser for R

Version 1.2.1

Date 2013-04-22

Author Thomas Lin Pedersen, Vladislav A Petyuk with contributions from
Laurent Gatto and Sebastian Gibb.

Maintainer Thomas Lin Pedersen <thomasp85@gmail.com>

Description A parser for mzIdentML files implemented using the XML package.
The parser tries to be general and able to handle all types of mzIdentML
files with the drawback of having less 'pretty' output than a vendor
specific parser. Please contact the maintainer with any problems and supply
an mzIdentML file so the problems can be fixed quick.

License GPL (>= 2)

LazyLoad yes

biocViews DataImport, MassSpectrometry, Proteomics

VignetteBuilder knitr

Depends methods

Imports XML, plyr, parallel, doParallel, foreach, iterators

Suggests knitr

Collate 'aaa.R' 'generics.R' 'mzIDpsm.R' 'mzIDpeptides.R'
'mzIDparameters.R' 'mzIDevidence.R' 'mzIDdatabase.R' 'mzID.R'
'mzIDCollection.R' 'mzID-constructor.R' 'mzID-package.R'

R topics documented:

mzID-package	3
attrExtract	3
attrExtractNameValuePair	4
countChildren	4
flatten	6
getAdditionalPar	7
getDatabaseFile	8
getDatabaseFilter	8
getDatabaseTranslation	9
getEnzymes	10
getFragmentTolerance	10
getMassTable	11
getModifications	12
getParentTolerance	12
getPath	13
getRawFile	14
getSearchType	14
getSoftware	15
getThreshold	16
getVersion	16
isRemote	17
length	17
length,mzID-method	18
length,mzIDCollection-method	19
length,mzIDdatabase-method	19
length,mzIDevidence-method	20
length,mzIDparameters-method	20
length,mzIDpeptides-method	21
length,mzIDpsm-method	21
mzID	22
mzID-class	23
mzIDCollection	24
mzIDCollection-class	25
mzIDdatabase	25
mzIDdatabase-class	26
mzIDevidence	27
mzIDevidence-class	28
mzIDparameters	28
mzIDpeptides	29
mzIDpeptides-class	30
mzIDpsm	30
mzIDpsm-class	31
prepareXML	32
show	32
show,mzID-method	33
show,mzIDCollection-method	34

show,mzIDdatabase-method 34
 show,mzIDevidence-method 35
 show,mzIDparameters-method 35
 show,mzIDpeptides-method 36
 show,mzIDpsm-method 37
 type.convert 37
 Utilities 38

Index 39

mzID-package *A parser for the mzIdentML file format*

Description

Using the mzID function this package is able to parse mzIdentML files into an mzID class instance.

Author(s)

Thomas Dybdal Pedersen with contributions from Laurent Gatto

attrExtract *Extract and properly format node attributes*

Description

This is a function to mainly handle correct formatting of the output from `xmlAttrs`. In addition it allows for multiple named child nodes to be specified.

Usage

```
attrExtract(doc, ns, path, child, addFinalizer = FALSE)
```

Arguments

doc an XMLInternalDocument created using `xmlInternalTreeParse`
 ns A string with name x givin the namespace for the doc
 path An xpath expression giving the nodes to search in
 child A character vector giving the name of the children to search for
 addFinalizer Logical Sets whether reference counting should be turned on

Value

If `length(child) == 1` a data.frame with a column for each attribute name and rows giving the value in each node. If `length(child) > 1` a named list containing a data.frame for each child. NA values are inserted if an attribute is missing from a node.

attrExtractNameValuePair

Extract and format attributes from nodes with name and value attributes

Description

This function handles the special case of extracting information from nodes with the attributes name and value. This function treats the value of the name attribute as the category for the value of the value attribute, so that it in essence equals an XML node of the form `<node category=value/>`. Several child names can be specified and the result will in the end be merged to a single data.frame rows for each node in the path argument.

Usage

```
attrExtractNameValuePair(doc, ns, path, child, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	A string with name x givin the namespace for the doc
path	An xpath expression giving the nodes to search in
child	A character vector giving the name of the children to search for
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with columns for each unique value in the name attribute of the nodes search in. The data.frame will have rows for each node in the path expression, regardless of whether it contains children.

countChildren	<i>Report the number of children with specific name per node in an XML document</i>
---------------	---

Description

This function takes an xpath argument and reports the number of children with given names at each node. Several children can be specified

Usage

```
countChildren(doc, ns, path, child, withPar, simplify = TRUE,
  addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	A string with name x givin the namespace for the doc
path	An xpath expression giving the nodes to search in
child	A character vector giving the name of the children to search for
withPar	(optional) A character string giving a name of an attribute that must exist in the child for it to be considred in counting.
simplify	(default=TRUE) A logical value giving whether the output should be returned as a dataframe if only one child is given
addFinalizer	Logical Sets whether reference counting should be turned on

Details

Given an xmlInternalDocument, a namespace, an xpath argument and a number of names this function will count, for each node in the xpath argument, how many children with the specified name(s) exists. It is optimized for several child names, so that the performance hit with additional names are neglectible.

An additional constraint on the match can be given with the withPar argument, which requests that each child, in addition to having the correct name, also should contain an attribute with the name given in the withPar argument, in order to be counted. This search requires addition look- ups in the XML document and will thus give a performance hit.

If our XML document had the following structure, and doc and ns was set up correctly

```
<node/>
__<child1 value=5/>
__<child1 value=1/>
__<child2/>
</node>
<node/>
__<child1/>
__<child2/>
__<child2/>
</node>
```

```
countChildren(doc, ns, path=./x:node,child=child1)
```

would return:

```
[1] 2 1
```

```
countChildren(doc, ns, path=./x:node,child=child1, withPar=value)
```

would return:

```
[1] 2 0
```

```
countChildren(doc, ns, path=./x:node,child=child1, simplify=FALSE)
```

would return:

```
$child1
```

```
[1] 2 1
```

```
countChildren(doc, ns, path=./x:node,child=c(child1, child2)
would return:
$child1
[1] 2 1

$child2
[1] 1 2
```

Value

If `length(child) == 1` and `simplify == TRUE` a vector giving the number of matches per node.
 If `length(child) > 1` or `simplify == FALSE` a named list with an element per child argument containing a vector giving the number of matches per node.

flatten	<i>Flatten an mzID related class into a table</i>
---------	---

Description

This function flattens the content of the object into a table by merging the content intelligently (it knows the links between the different objects).

Usage

```
flatten(object, ...)

## S4 method for signature mzID
flatten(object, no.redundancy = FALSE)

## S4 method for signature mzIDCollection
flatten(object, no.redundancy = FALSE)

## S4 method for signature mzIDpeptides
flatten(object)

## S4 method for signature mzIDpsm
flatten(object)
```

Arguments

object	The object to be flattened
...	Additional parameters (currently unused)
no.redundancy	Logical Should duplicate peptides be removed. Default is FALSE as identical peptides from different proteins should normally be kept.

Value

A data.frame with the flattened result

See Also

[mzID-class](#)

Examples

```
mzResults <- mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/55merge_tandem.mzid")
head(flatten(mzResults))
```

getAdditionalPar	<i>Extracts the additional, often software specific parameters</i>
------------------	--

Description

This function extracts and formats the information stored in the cvParam and userParam nodes in /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/AdditionalSearchParams

Usage

```
getAdditionalPar(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A list with names corresponding to the name attribute of the node, and content corresponding to the value attribute. If the node haven't got a a value attribute the content is set to TRUE

See Also

[mzIDparameters-class](#)

getDatabaseFile	<i>Retrive information on the database used in the analysis</i>
-----------------	---

Description

This function collects the information regarding the database used in the analysis stored in the mzIdentML file

Usage

```
getDatabaseFile(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with a at least the location of the database stored

See Also

[mzIDparameters-class](#)

getDatabaseFilter	<i>Looks for any database filters applied during the peptide search</i>
-------------------	---

Description

This function is still unimplemented, as no proper use of the /MzIdentML/AnalysisProtocolCollection/SpectrumIdentification have been found

Usage

```
getDatabaseFilter(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

NULL

See Also

[mzIDparameters-class](#)

getDatabaseTranslation

Looks whether a translation table is present in the mzIdentML file and reports it if found

Description

This function checks for the existence of /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/Database? and if found formats these for easy reading.

Usage

```
getDatabaseTranslation(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with a translation table or NULL if none exists

See Also

[mzIDparameters-class](#)

getEnzymes	<i>Extracts the enzymes used for digestion specified as specified in the software analysis</i>
------------	--

Description

The function looks for information on the enzymes used in the protocol and if present extracts it

Usage

```
getEnzymes(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with names of the enzymes as well as other settings related to the cleavage of proteins or NULL if no information is present

See Also

[mzIDparameters-class](#)

getFragmentTolerance	<i>Looks for presence of fragment tolerance settings and reports these if found</i>
----------------------	---

Description

This function checks for the existence of /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/Fragment and if found reports these.

Usage

```
getFragmentTolerance(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with the settings used for fragment tolerance or NULL if none exists

See Also

[mzIDparameters-class](#)

getMassTable

Search for the presence of a mass table used during the search

Description

This function checks for the existence of /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/MassTable and if found formats these for easy reading.

Usage

```
getMassTable(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with the masses used for each amino acid at the different ms levels NULL if none exists

See Also

[mzIDparameters-class](#)

getModifications	<i>Looks for presence of amino acid modification settings and reports these if found</i>
------------------	--

Description

This function checks for the existence of /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/Modifications and if found formats these for easy reading.

Usage

```
getModifications(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with information on the modification settings in the search or NULL if none exists

See Also

[mzIDparameters-class](#)

getParentTolerance	<i>Looks for presence of parent tolerance settings and reports these if found</i>
--------------------	---

Description

This function checks for the existence of /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol/ParentTolerance and if found reports these.

Usage

```
getParentTolerance(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with the settings used for parent tolerance or NULL if none exists

See Also

[mzIDparameters-class](#)

getPath

Get the correct namespace path depending on version

Description

This function resolves an inconsistency between version 1.0 and 1.1 where the namespace is changed in case.

Usage

```
getPath(ns)
```

Arguments

ns	The namespace of the mzIdentML file
----	-------------------------------------

Value

A textstring with the top parent node of the DOM

getRawFile	<i>Retrive information on the raw files used in the analysis</i>
------------	--

Description

This function collects the information regarding the raw data files related to the analysis stored in the mzIdentML file

Usage

```
getRawFile(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with a row for each raw data file used in the analysis and at least a column for the location and the ID format of the file.

See Also

[mzIDparameters-class](#)

getSearchType	<i>Gets the search type as specified in the mzIdentML file</i>
---------------	--

Description

This function simply extracts the information stored in /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtoc

Usage

```
getSearchType(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A character with the name of the search type (e.g. 'ms-ms search' or 'de novo search')

See Also

[mzIDparameters-class](#)

getSoftware

Retrive information on the software used in the analysis

Description

This function collects the information regarding the analysis software used during creation of the data stored in the mzIDentML file

Usage

```
getSoftware(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with a row for each software used, and columns with at least name and id of the software

See Also

[mzIDparameters-class](#)

getThreshold	<i>Finds the psm threshold used in the file</i>
--------------	---

Description

This function simply extracts the attributes stored in /MzIdentML/AnalysisProtocolCollection/SpectrumIdentificationProtocol

Usage

```
getThreshold(doc, ns, addFinalizer = FALSE)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A data.frame with columns 'name' and 'value' storing the thresholds used

See Also

[mzIDparameters-class](#)

getVersion	<i>Get the mzIdentML version and check that the version is supported</i>
------------	--

Description

Currently version 1.1 and 1.0 are supported.

Usage

```
getVersion(ns)
```

Arguments

ns	The namespace of the mzIdentML file
----	-------------------------------------

Value

A textstring giving the version of the mzIdentML file if it is supported. In case of missing support it throws an error.

isRemote	<i>Test whether a given string is a remote resource</i>
----------	---

Description

Test whether a given string is a remote resource

Usage

```
isRemote(x)
```

Arguments

x	the path to the xml file
---	--------------------------

Value

logical

length	<i>Report the length of an mzIDpsm object</i>
--------	---

Description

The length of an mzIDpsm object is defined as the number of psm's. An empty object has a length of 0

The length of an mzIDpeptides object is defined as the number of peptides in the @peptides slot. An empty object has a length of 0

The length of an mzIDparameters object is defined as the number of raw datafiles used in the analysis. An empty object has a length of 0

The length of an mzIDevidence object is defined as the number of peptide evidences. An empty object has a length of 0

The length of an mzIDdatabase object is defined as the number of entries in the databse. An empty object has a length of 0

The length of an mzID object is defined as the number of PSMs. An empty object has a length of 0

Arguments

x	An mzIDpsm object
x	An mzIDpeptides object
x	An mzIDparameters object
x	An mzIDevidence object
x	An mzIDdatabase object
x	An mzID object

Value

- A numeric giving the number of psm's in the mzIDpsm object
- A numeric giving the number of peptides in the mzIDpeptides object
- A numeric giving the number raw datafiles used in the analysis
- A numeric giving the number of peptide evidences in the mzIDEvidence object
- A numeric giving the number of entries in the mzIDdatabase object
- A numeric giving the number of PSMs in the mzID object

See Also

[mzIDpsm-class](#)
[mzIDpeptides-class](#)
[mzIDparameters-class](#)
[mzIDEvidence-class](#)
[mzIDdatabase-class](#)
[mzID-class](#)

length,mzID-method	<i>Report the length of an mzID object</i>
--------------------	--

Description

The length of an mzID object is defined as the number of PSMs. An empty object has a length of 0

Usage

```
## S4 method for signature mzID  
length(x)
```

Arguments

x An mzID object

Value

A numeric giving the number of PSMs in the mzID object

See Also

[mzID-class](#)

length,mzIDCollection-method
Get the length of an mzIDCollection object

Description

The length of an mzIDCollection object is defined as the number of mzID objects it contain.

Usage

```
## S4 method for signature mzIDCollection
length(x)
```

Arguments

x An mzIDCollection object

Value

An integer giving the number of mzID objects contained in the mzIDCollection

See Also

[mzIDCollection-class](#)

length,mzIDdatabase-method
Report the length of an mzIDdatabase object

Description

The length of an mzIDdatabase object is defined as the number of entries in the database. An empty object has a length of 0

Usage

```
## S4 method for signature mzIDdatabase
length(x)
```

Arguments

x An mzIDdatabase object

Value

A numeric giving the number of entries in the mzIDdatabase object

See Also[mzIDdatabase-class](#)

`length,mzIDevidence-method`*Report the length of an mzIDevidence object*

Description

The length of an mzIDevidence object is defined as the number of peptide evidences. An empty object has a length of 0

Usage

```
## S4 method for signature mzIDevidence
length(x)
```

Arguments

x An mzIDevidence object

Value

A numeric giving the number of peptide evidences in the mzIDevidence object

See Also[mzIDevidence-class](#)

`length,mzIDparameters-method`*Report the length of an mzIDparameters object*

Description

The length of an mzIDparameters object is defined as the number of raw datafiles used in the analysis. An empty object has a length of 0

Usage

```
## S4 method for signature mzIDparameters
length(x)
```

Arguments

x An mzIDparameters object

Value

A numeric giving the number raw datafiles used in the analysis

See Also

[mzIDparameters-class](#)

length,mzIDpeptides-method

Report the length of an mzIDpeptides object

Description

The length of an mzIDpeptides object is defined as the number of peptides in the @peptides slot. An empty object has a length of 0

Usage

```
## S4 method for signature mzIDpeptides
length(x)
```

Arguments

x An mzIDpeptides object

Value

A numeric giving the number of peptides in the mzIDpeptides object

See Also

[mzIDpeptides-class](#)

length,mzIDpsm-method *Report the length of an mzIDpsm object*

Description

The length of an mzIDpsm object is defined as the number of psm's. An empty object has a length of 0

Usage

```
## S4 method for signature mzIDpsm
length(x)
```

Arguments

x An mzIDpsm object

Value

A numeric giving the number of psm's in the mzIDpsm object

See Also

[mzIDpsm-class](#)

mzID

Parse an mzIdentML file

Description

This function takes a single mzIdentML file and parses it into an mzID object.

Usage

```
mzID(file, verbose = TRUE)
```

Arguments

file A character string giving the location of the mzIdentML file to be parsed
verbose Logical Should information be printed to the console? Default is TRUE

Details

The mzID function uses the XML package to read the content of an mzIdentML file and store it in an mzID object. Unlike how mzR handles mzML files, mzID parses everything in one chunk. Memory can thus be a problem for very big datasets, but as mzIdentML files are not indexed, it is inefficient to access the data dynamically.

If multiple filenames are passed to the function they will be processed in parallel using foreach and doParallel. The number of workers spawned is either the maximal number of available cores or the number of files to parse, whichever is smallest. The return value will in these cases be an mzIDCollection object. If some of the files cannot be parsed they will not be contained in the returned object and a warning will be issued. No errors will be thrown.

Value

An mzID object

See Also

[mzID-class](#) [mzIDCollection-class](#)

Examples

```
# Parsing of the example files provided by HUP0:
mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/55merge_tandem.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/55merge_omssa.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/Sequest_example_ver1.1.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/Mascot_NA_example.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/Mascot_top_down_example.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/MPC_example_Multiple_search_engines.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/mascot_pmf_example.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/Sequest_example_ver1.1.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_1examples/Mascot_MSMS_example.mzid")

mzID("http://psi-pi.googlecode.com/svn/trunk/examples/1_0examples/Mascot_MSMS_example.mzid")
```

mzID-class

A class to contain data from mzIdentML-files

Description

This class stores all parsed information from mzIdentML files

Details

The mzID class stores information in a subset of classes, each class having its own slot. While These classes should not need to be accessed directly, but descriptions of their content is delegated to each respective class.

Objects from the class

Objects can be created using the `mzID` constructor, which handles parsing of mzIdentML files

Slots

parameters: An instance of `mzIDparameters-class`. This object contains all information related to how the analysis was carried out.

psm: An instance of `mzIDpsm-class`. This object contains the meat of the analysis with all scans and their related PSMs recorded.

peptides: An instance of `mzIDpeptides-class`. This object contains a library of all peptides generated from the database along with possible modifications.

evidence: An instance of [mzIDEvidence-class](#). This object lists all peptides detected in the analysis with reference to the [mzIDpeptides](#) instance.

database: An instance of [mzIDdatabase-class](#). This object contains information on the proteins in the database. As the full database is not recorded in mzIdentML files the actual protein sequence is not recorded but there is sufficient information to retrieve it from the database file.

Methods

[length:](#) Reports the number of PSMs in the object.

[flatten:](#) ...

References

<http://www.psicodev.info/mzidentml>

See Also

[mzID](#) [mzIDparameters-class](#) [mzIDpsm-class](#) [mzIDpeptides-class](#) [mzIDEvidence-class](#) [mzIDdatabase-class](#)

mzIDCollection

Create a new mzIDCollection

Description

This function creates a new `mzIDCollection` object containing the supplied `mzID` object. As such the result is equivalent to passing a number of `mzID` objects to `c()`, except that an empty `mzIDCollection` object is returned if no `mzID` objects are supplied.

Usage

```
mzIDCollection(...)
```

Arguments

... An arbitrary number of `mzID` objects

Value

An `mzIDCollection` object

See Also

[mzID-class](#) [mzIDCollection-class](#)

mzIDCollection-class *A class to handle a set of mzID objects*

Description

This class is a container for multiple mzID objects. It is constructed such that the bulk data are not copied when passed around. It is the aim that this class have parity with the mzID class in the methods it exposes to the user, such that mzIDCollections can be thought of as vectors in the traditional R sense. Furthermore it accepts standard indexing and concatenation.

Details

Objects of this class is usually constructed be passing mulitple files to the mzID constructor, or by combining mulitple mzID objects.

mzIDdatabase *A constructor for the mzIDdatabase class*

Description

This function handles parsing of data and construction of an mzIDdatabase object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

Usage

```
mzIDdatabase(doc, ns, addFinalizer = FALSE, path)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on
path	If doc is missing the file specified here will be parsed

Value

An mzIDdatabase object

See Also

[mzIDdatabase-class](#)

mzIDdatabase-class *A class to store database information from an mzIdentML file*

Description

This class handles parsing and storage of database information from mzIdentML files, residing at the /MzIdentML/SequenceCollection/DBSequence node.

This class tries to collect the multitude of different analysis information required to rerun the analysis. The intended data to be stored are: The software used in the analysis of the data, the location and nature of the rawfile(s), the location and nature of the database file(s), the location of the mzIdentML file itself as well as all the parameters used during the analysis leading to the mzIdentML file. Information regarding how the LC-MS experiment was performed should be collected from the raw data file. As the parameters used in different software solutions can vary greatly, all these parameters are stored in a named list, which can thus be very different from pipeline to pipeline. It is the users responsibility to check conformity between samples.

Details

The content of the class is stored in a data.frame with columns depending on the content of the mzIdentML file. Required information for files conforming to the mzIdentML standard are: 'accession', 'searchDatabase_ref' and 'id', while additional information can fx be 'length' (number of residues), 'description' (from the fasta file) and 'sequence' (the actual sequence).

Objects from the class

Objects of mzIDdatabase are not meant to be created explicitly but as part of the [mzID-class](#). Still object can be created with the constructor [mzIDdatabase](#) (not exported).

Slots

database: A data.frame containing references to all the database sequences from the mzIdentML file

Methods

length: Reports the number of entries in the database

Objects from the class

Objects of mzIDparameters are not meant to be created explicitly but as part of the [mzID-class](#). Still object can be created with the constructor [mzIDparameters](#) (not exported).

Slots

software: A data frame with information retaining to the software used for the analysis. At least the name and an id is given, but optionally also version number and URI

rawFile: A data frame with information about the raw data file(s) used for the analysis. The data frame will contain at least the location and spectrum ID format.

databaseFile: A data frame containing at least the location and file format of the database file used in the search.

idFile: A character string containing the location of the mzIdentML file at the time of parsing.

parameters: A list containing containing the information stored in the MzIdentML/AnalysisProtocolCollection/SpectrumId node. SearchType and Threshold are the only required parameters given by the mzIdentML standard.

Methods

length: Reports the number of raw files used in the analysis.

See Also

[mzID-class mzIDdatabase](#)

[mzID-class mzIDparameters](#)

mzIDevidence

A constructor for the mzIDevidence class

Description

This function handles parsing of data and construction of an mzIDevidence object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

Usage

```
mzIDevidence(doc, ns, addFinalizer = FALSE, path)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on
path	If doc is missing the file specified here will be parsed

Value

An mzIDevidence object

See Also

[mzIDevidence-class](#)

mzIDevidence-class *A class to store peptide evidence information from an mzIdentML file*

Description

This class handles parsing and storage of peptide evidence information from mzIdentML files, residing at the `/*x:SequenceCollection/x:PeptideEvidence` node.

Details

The content of the class is stored in a `data.frame` with columns depending on the content of the mzIdentML file. Columns represent the attribute values of for each `PeptideEvidence` node. For files conforming to the HUPO standard, `dbSequence_ref`, `id` and `peptide_ref` is required while `start`, `end`, `pre`, `post`, `name`, `isDecoy`, `frame` and `translationTable_ref` are optional. Information residing in `cvParam` and `userParam` children are not parsed.

Objects from the class

Objects of `mzIDevidence` are not meant to be created explicitly but as part of the `mzID-class`. Still object can be created with the constructor `mzIDevidence` (not exported).

Slots

`evidence`: A `data.frame` containing all peptide evidence from the mzIdentML file

Methods

`length`: Reports the number of peptide evidences

See Also

[mzID-class mzIDevidence](#)

mzIDparameters *A constructor for the mzIDparameters class*

Description

This function handles parsing of data and construction of an `mzIDparameters` object. This function is not intended to be called explicitly but as part of an `mzID` construction. Thus, the function is not exported. It relies on a number of getter functions to retrieve the different information from around the document.

Usage

```
mzIDparameters(doc, ns, addFinalizer = FALSE, path)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on
path	If doc is missing the file specified here will be parsed

Value

An mzIDparameters object

See Also

[mzIDparameters-class](#)

mzIDpeptides	<i>A constructor for the mzIDpeptides class</i>
--------------	---

Description

This function handles parsing of data and construction of an mzIDpeptides object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

Usage

```
mzIDpeptides(doc, ns, addFinalizer = FALSE, path)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on
path	If doc is missing the file specified here will be parsed

Value

An mzIDpeptides object

See Also

[mzIDpeptides-class](#)

mzIDpeptides-class *A class to store peptide information from an mzIdentML file*

Description

This class handles parsing and storage of peptide information from mzIdentML files, residing at the /x:MzIdentML/x:SequenceCollection/x:Peptide node.

Details

The information is stored in a dataframe with an id, an optional name and the amino acid sequence of the peptide. Alongside a list is stored with modification information of each peptide. Each row in the dataframe has a corresponding entry in the list. If no modification of the peptide is present the entry is NULL, if a modification is present the entry is a dataframe, listing the different modifications of the peptide.

Objects from the class

Objects of mzIDpeptides are not meant to be created explicitly but as part of the [mzID-class](#). Still object can be created with the constructor [mzIDpeptides](#) (not exported).

Slots

peptides: A data.frame containing all peptides used in the search

modifications: A list containing possible modifications of the peptides listed in @peptides

Methods

length: Reports the number of peptides

See Also

[mzID-class mzIDpeptides](#)

mzIDpsm *A constructor for the mzIDpsm class*

Description

This function handles parsing of data and construction of an mzIDpsm object. This function is not intended to be called explicitly but as part of an mzID construction. Thus, the function is not exported.

Usage

```
mzIDpsm(doc, ns, addFinalizer = FALSE, path)
```

Arguments

doc	an XMLInternalDocument created using xmlInternalTreeParse
ns	The appropriate namespace for the doc, as a named character vector with the namespace named x
addFinalizer	Logical Sets whether reference counting should be turned on
path	If doc is missing the file specified here will be parsed

Value

An mzIDpsm object

See Also

[mzIDpsm-class](#)

mzIDpsm-class	<i>A class to store psm information from an mzIdentML file</i>
---------------	--

Description

This class handles parsing and storage of scan info and the related psm's. This information resides in the `/*x:DataCollection/x:AnalysisData/x:SpectrumIdentificationList/x:SpectrumIdentificationResult` node.

Details

The content of the class is stored as two data frames: One containing a row for each scan in the results, and one containing all psm's in the results. Additionally a list containing indexing from scan to psm is stored.

Objects from the class

Objects of mzIDpsm are not meant to be created explicitly but as part of the `mzID-class`. Still object can be created with the constructor `mzIDpsm` (not exported).

Slots

scans: A data.frame containing all reference to all scans with at least one psm. The columns gives at least an ID, a spectrumID and a reference to the file used

id: A data.frame containing all psm's from the analysis. The columns depend on the file but at least id, chargeState, experimentalMassToCharge, passThreshold and rank must exist according to the mzIdentML specifications

mapping: A list with an entry for each row in @scans. Each entry contains an integer vector pointing to the related rows in @id

Methods

length: Reports the number of psm's

See Also

[mzID-class mzIDpsm](#)

prepareXML	<i>Parses an xml file and defines the namespace</i>
------------	---

Description

Parses an xml file and defines the namespace

Usage

```
prepareXML(path, addFinalizer = FALSE)
```

Arguments

path	The path to the xml file
addFinalizer	Logical Sets whether reference counting should be turned on

Value

A list with the named elements: doc: the results of xmlInternalTreeParse and ns: the namespace as a vector.

show	<i>Show method for mzIDpsm objects</i>
------	--

Description

This function reports general information on the mzIDpsm object. It is called automatically when an object is queried.

This function reports general information on the mzIDpeptides object. It is called automatically when an object is queried.

This function reports general information on the mzIDparameters object. It is called automatically when an object is queried.

This function reports general information on the mzIDevidence object. It is called automatically when an object is queried.

This function reports general information on the mzIDdatabase object. It is called automatically when an object is queried.

This function reports general information on the mzID object. It is called automatically when an object is queried.

Arguments

object	An mzIDpsm object
object	An mzIDpeptides object
object	An mzIDparameters object
object	An mzIDevidence object
object	An mzIDdatabase object
object	An mzID object

Value

A description of the content of the mzIDpsm object
A description of the content of the mzIDpeptides object
A description of the content of the mzIDparameters object
A description of the content of the mzIDevidence object
A description of the content of the mzIDdatabase object
A description of the content of the mzID object

See Also

[mzIDpsm-class](#)
[mzIDpeptides-class](#)
[mzIDparameters-class](#)
[mzIDevidence-class](#)
[mzIDdatabase-class](#)
[mzID-class](#)

show,mzID-method	<i>Show method for mzID objects</i>
------------------	-------------------------------------

Description

This function reports general information on the mzID object. It is called automatically when an object is queried.

Usage

```
## S4 method for signature mzID  
show(object)
```

Arguments

object	An mzID object
--------	----------------

Value

A description of the content of the mzID object

See Also

[mzID-class](#)

show,mzIDCollection-method

Show method for mzIDCollection objects

Description

This function gives a very brief overview of the content of an mzIDCollection object.

Usage

```
## S4 method for signature mzIDCollection
show(object)
```

Arguments

object An mzIDCollection object

See Also

[mzIDCollection-class](#)

show,mzIDdatabase-method

Show method for mzIDdatabase objects

Description

This function reports general information on the mzIDdatabase object. It is called automatically when an object is queried.

Usage

```
## S4 method for signature mzIDdatabase
show(object)
```

Arguments

object An mzIDdatabase object

Value

A description of the content of the mzIDdatabase object

See Also

[mzIDdatabase-class](#)

show,mzIDevidence-method

Show method for mzIDevidence objects

Description

This function reports general information on the mzIDevidence object. It is called automatically when an object is queried.

Usage

```
## S4 method for signature mzIDevidence
show(object)
```

Arguments

object An mzIDevidence object

Value

A description of the content of the mzIDevidence object

See Also

[mzIDevidence-class](#)

show,mzIDparameters-method

Show method for mzIDparameters objects

Description

This function reports general information on the mzIDparameters object. It is called automatically when an object is queried.

Usage

```
## S4 method for signature mzIDparameters
show(object)
```

Arguments

object An mzIDparameters object

Value

A description of the content of the mzIDparameters object

See Also

[mzIDparameters-class](#)

show,mzIDpeptides-method

Show method for mzIDpeptides objects

Description

This function reports general information on the mzIDpeptides object. It is called automatically when an object is queried.

Usage

```
## S4 method for signature mzIDpeptides  
show(object)
```

Arguments

object An mzIDpeptides object

Value

A description of the content of the mzIDpeptides object

See Also

[mzIDpeptides-class](#)

show,mzIDpsm-method *Show method for mzIDpsm objects*

Description

This function reports general information on the mzIDpsm object. It is called automatically when an object is queried.

Usage

```
## S4 method for signature 'mzIDpsm'
show(object)
```

Arguments

object An mzIDpsm object

Value

A description of the content of the mzIDpsm object

See Also

[mzIDpsm-class](#)

type.convert *An extension of type.convert() to handle XML like logicals (lower case)*

Description

This function is a simple extension to the [type.convert](#) function. It handles the extra case where a vector of lower case true and false value would not be converted to logicals. As this is the standard format in XML files, this extension is needed

Usage

```
type.convert(...)
```

Arguments

... Parameters passed on to type.convert

See Also

[type.convert](#)

Description

These functions provide list-like handling of `mzIDCollection` object. Their behavior are fully in line with that of standard R lists.

Usage

```
## S4 method for signature mzIDCollection
names(x)

## S4 method for signature mzIDCollection
x names<- value

## S4 method for signature mzIDCollection,numeric,missing
x[[i, j, ...]]

## S4 method for signature mzIDCollection,character,missing
x[[i, j, ...]]

## S4 method for signature mzID
c(x, y, ..., recursive = FALSE)

## S4 method for signature mzIDCollection
c(x, y, ..., recursive = FALSE)
```

Arguments

<code>x</code>	An <code>mzIDCollection</code> object
<code>value</code>	A string giving the name(s) to set
<code>i</code>	An integer or a string giving the index or the name respectively
<code>j</code>	ignored
<code>...</code>	ignored
<code>y</code>	An <code>mzID</code> or <code>mzIDCollection</code> object
<code>recursive</code>	logical. If <code>recursive = TRUE</code> , the function recursively descends through lists (and pairlists) combining all their elements into a vector.

Index

[[,mzIDCollection,character,missing-method (Utilities), 38
[[,mzIDCollection,numeric,missing-method (Utilities), 38

attrExtract, 3
attrExtractNameValuePair, 4

c,mzID-method (Utilities), 38
c,mzIDCollection-method (Utilities), 38
countChildren, 4

flatten, 6, 24
flatten,mzID-method (flatten), 6
flatten,mzIDCollection-method (flatten), 6
flatten,mzIDpeptides-method (flatten), 6
flatten,mzIDpsm-method (flatten), 6

getAdditionalPar, 7
getDatabaseFile, 8
getDatabaseFilter, 8
getDatabaseTranslation, 9
getEnzymes, 10
getFragmentTolerance, 10
getMassTable, 11
getModifications, 12
getParentTolerance, 12
getPath, 13
getRawFile, 14
getSearchType, 14
getSoftware, 15
getThreshold, 16
getVersion, 16

isRemote, 17

length, 17, 24
length,mzID-method, 18
length,mzID-method (length), 17
length,mzIDCollection-method, 19
length,mzIDdatabase-method, 19
length,mzIDdatabase-method (length), 17
length,mzIDevidence-method, 20
length,mzIDevidence-method (length), 17
length,mzIDparameters-method, 20
length,mzIDparameters-method (length), 17
length,mzIDpeptides-method, 21
length,mzIDpeptides-method (length), 17
length,mzIDpsm-method, 21
length,mzIDpsm-method (length), 17

mzID, 22, 23, 24
mzID-class, 23
mzID-package, 3
mzIDCollection, 24
mzIDCollection-class, 25
mzIDdatabase, 25, 26, 27
mzIDdatabase-class, 26
mzIDevidence, 27, 28
mzIDevidence-class, 28
mzIDparameters, 26, 27, 28
mzIDparameters-class (mzIDdatabase-class), 26
mzIDpeptides, 29, 30
mzIDpeptides-class, 30
mzIDpsm, 30, 31, 32
mzIDpsm-class, 31

prepareXML, 32

show, 32
show,mzID-method, 33
show,mzIDCollection-method, 34
show,mzIDdatabase-method, 34
show,mzIDevidence-method, 35
show,mzIDparameters-method, 35
show,mzIDpeptides-method, 36
show,mzIDpsm-method, 37

type.convert, 37, 37

Utilities, [38](#)

xmlAttrs, [3](#)

xmlInternalTreeParse, [3–5](#), [7–16](#), [25](#), [27](#),
[29](#), [31](#)