Package ‘PamBinaries’

October 9, 2021

Title Read and Process ‘Pamguard’ Binary Data

Version 1.6.0

Description Functions for easily reading and processing binary data files created by ‘Pamguard’ (<https://www.pamguard.org/>). All functions for directly reading the binary data files are based on ‘MATLAB’ code written by Michael Oswald.

Depends R (>= 3.4.0)

License GNU General Public License

Encoding UTF-8

RoxygenNote 7.1.1

Imports ggplot2, dplyr

Suggests testthat

NeedsCompilation no

Author Taiki Sakai [aut, cre], Michael Oswald [aut], Douglas Gillespie [ctb]

Maintainer Taiki Sakai <taiki.sakai@noaa.gov>

Repository CRAN

Date/Publication 2021-10-09 05:50:02 UTC

R topics documented:

  contourToFreq .................................................. 2
  convertPgDate .................................................. 3
  countChannels .................................................. 4
  dateNumToMillis ............................................... 5
  loadBackgroundNoise ............................................ 5
  loadPamguardBinaryFile ....................................... 6
  millisToDateNum ............................................... 8
  pamBinRead .................................................... 8
  pbToDf .......................................................... 9
  plotWMD ......................................................... 10
### contourToFreq

**Add Frequency and Time to Pamguard Whistle Binaries**

#### Description

Adds items freq and time to a Pamguard binary file from the Whistle & Moan Detector.

#### Usage

```r
contourToFreq(data, verbose = FALSE)
```

#### Arguments

- **data**: either a PamBinary class object or just the $data from a PamBinary object.
- **verbose**: logical flag to print calculated parameters.
Value

data with items \texttt{freq} and \texttt{time} added. These use the calculated FFT window length, hop size, and sample rate to compute the frequency and time values of the saved whistle contour.

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

Examples

# load example whistle file
wmFile <- system.file("extdata", "WM.pgdf", package="PamBinaries")
wmData <- loadPamguardBinaryFile(wmFile)
# converts contour and FFT slice numbers to frequency and time values
wmData <- contourToFreq(wmData)
wmData$data[[1]]$contour
wmData$data[[1]]$freq
wmData$data[[1]]$time

---

### convertPgDate

**Convert Pamguard Numeric Date to POSIXct**

Description

a simple helper to convert Pamguard’s numeric date to POSIXct format.

Usage

convertPgDate(dateNum)

Arguments

dateNum 

date as a numeric, seconds since 1970-01-01 per standard Pamguard output. Timezone is UTC.

Value

A POSIXct date in UTC

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>
countChannels

Examples

# load the example click binary data, leaving date as numeric
clickFile <- system.file('extdata', 'Click.pgdf', package='PamBinaries')
clickData <- loadPamguardBinaryFile(clickFile, convertDate = FALSE)
# convert date to POSIXct
convertPgDate(clickData$data[[1]]$date)

countChannels | Count Number of Active Channels

Description

Counts the number of active channels given a channel mapping

Usage

countChannels(channelMap)

Arguments

channelMap | Mapping of channels as a binary number

Value

The number of active channels (number of ones)

Note

Altered from original script to loop through 30 instead 32 because R stores only 32 bit integers. Should not ever have enough channels for this to matter.

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>
**dateNumToMillis**  
*Convert Date Number to Milliseconds*

**Description**
Converts numeric date to millisecond date.

**Usage**
dateNumToMillis(datenum)

**Arguments**
datenum Numeric value of a date.

**Value**
Date as milliseconds

**Note**
Conversion to milliseconds to match how Java stores dates. Doesn’t appear to ever be used.

**Author(s)**
Taiki Sakai <taiki.sakai@noaa.gov>

---

**loadBackgroundNoise**  
*Load and Format Background Noise Data*

**Description**
Reads and formats background noise data from Pamguard binary files or if not present in the original file will try to read the accompanying .pgnf noise file if it exists.

**Usage**
loadBackgroundNoise(x)
plotBackgroundNoise(x)
combineBackgroundNoise(x, forPlot = FALSE)
loadPamguardBinaryFile

Arguments

- **x** character pointing to a Pamguard binary file, or a PamBinary object created by loadPamguardBinaryFile. For plotting or combining, either of these or the output from loadBackgroundNoise
- **forPlot** logical flag when combining noise data. If used for plotting purposes this will insert NA columns into background data so that images show up with time gaps as expected. Leave as FALSE unless you are sure you want this.

Value

A list with times storing the POSIXct time of each background measurement, and background a matrix of background values. For binary data based on spectrogram measurements, there will also be freq the frequency in Hertz for each column of background measurement

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

Examples

```r
# load the example click binary data, leaving date as numeric
gplFile <- system.file('extdata', 'GPL.pgd', package='PamBinaries')
gplNoise <- loadBackgroundNoise(gplFile)
plotBackgroundNoise(gplNoise)
```

loadPamguardBinaryFile

Load Pamguard Binary File

Description

This function will load in the data from a Pamguard binary file. It will figure out the type of data being read based on the header of the file. All functions based on Matlab code written by Michael Oswald.

Usage

```r
loadPamguardBinaryFile(
  fileName,
  skipLarge = FALSE,
  skipData = FALSE,
  debug = FALSE,
  keepUIDs = NULL,
  convertDate = FALSE,
  ...
)
```
loadPamguardBinaryFile

Arguments

fileName
The name of the binary file to be read

skipLarge
Should we skip large parts of binaries? Currently only applicable to whistle, click, and DIFAR data

skipData
Should we skip all data and only read headers and footers?

debug
logical flag to show more info on errors

keepUIDs
If not NULL, a vector of UIDs to read. All UIDs not in this vector will not be read.

convertDate
logical flag to convert date from numeric to POSIXct. Defaults to FALSE for speed, can reduce time by

... Arguments passed to other functions

Value

This function returns a list containing two objects. Data contains all the binary data read. fileInfo contains metadata information for the file.

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

Examples

# read example whistle data
wmFile <- system.file('extdata', 'WM.pgdf', package='PamBinaries')
whistleData <- loadPamguardBinaryFile(wmFile)

# works the same for different kinds of binary files

clickFile <- system.file('extdata', 'Click.pgdf', package='PamBinaries')
clickData <- loadPamguardBinaryFile(clickFile)

# convert date to POSIXct (default does not because it is faster)

clickPOSIX <- loadPamguardBinaryFile(clickFile, convertDate = TRUE)
clickData$[1]$date
clickPOSIX$[1]$date

# read only the fileInfo portion, has empty $data item

clickInfo <- loadPamguardBinaryFile(clickFile, skipData = TRUE)

# skip reading the large click waveforms, much faster if you dont need them

clickLess <- loadPamguardBinaryFile(clickFile, skipLarge = TRUE)

object.size(clickData)
oct.size(clickLess)

# only read specific UID numbers


clickSpecific <- loadPamguardBinaryFile(clickFile, keepUIDs = c(4000006, 4000007))

names(clickSpecific$data)
pamBinRead

Converting Java Millisecond Time to R

Description

Converts Java millisecond time into numeric time that R uses.

Usage

millisToDateNum(millis)

Arguments

millis

Millisecond time from Java

Value

Numeric time used by R.

Note

Original function was more relevant as Matlab and Java use different time origins. Java & R both use 1970-01-01, but Java stores as milliseconds vs seconds in R.

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

pamBinRead

Read Pamguard Binary Data

Description

A wrapper for reading various types of binary data.

Usage

pamBinRead(
  fid,
  what = c("int8", "int16", "int32", "int64", "uint8", "uint16", "float", "double",
           "character"),
  n,
  seek = FALSE
)
pbToDf

Arguments

fid  The binary file being read
what The type of data to read. Int64 is not handled natively by R, see note.
n  The number of objects to read.
seek  Whether or not to just seek instead of reading

Value

Data of the type and number specified.

Note

R does not natively support 64-bit integers. Current implementation is to read an int64 as 8 separate 1-byte raw pieces. These are converted from hexadecimal, shifted by the appropriate power of 2, then summed. Currently cannot read more than one int64 at a time, shouldn’t be necessary.

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

---

pbToDf  Convert a PamBinary Object to Data Frame

Description

Converts a PamBinary object into a data frame. The data.frame will combine all of the data from the data part of the PamBinary object, but will not include annotations data, click waveforms, DIFAR demux data, or contours from the WMD detector. These are skipped because they are either inconsistent in their size, or are large objects. The function pbToDf is also called when as.data.frame is called on a PamBinary class object.

Usage

pbToDf(pb, templateNames = NULL)

Arguments

pb  a PamBinary class object created by loadPamguardBinaryFile

templateNames  if using the click template classifier, the names of the species for the click templates. These will be used as the names of the columns in the data frame, and the length of this must exactly match the number of templates used. Will add columns for the threshold, match, and reject correlation values for each template name provided
plotWMD

Plot Whistle Contour

Description

Plots the entire whistle contour saved in a Pamguard Whistle & Moan Detector binary file, highlighting the selected contour

Usage

plotWMD(data, id = 1, ...)

Arguments

data: either a PamBinary class object, or just the $data from a PamBinary object, or a single detection from the $data

id: the id of the whistle to plot, either an index or Pamguard UID

...: parameters to pass to other functions

Value

A ggplot object
readAISData

Author(s)
Taiki Sakai <taiki.sakai@noaa.gov>

Examples

# load example whistle file
wmFile <- system.file('extdata', 'WM.pgdf', package='PamBinaries')
wmData <- loadPamguardBinaryFile(wmFile)
plotWMD(wmData, 1)
plotWMD(wmData, 2)

Description
Reads binary data stored by the AIS Processing module.

Usage
readAISData(fid, fileInfo, data, debug = FALSE, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fid</td>
<td>binary file identifier</td>
</tr>
<tr>
<td>fileInfo</td>
<td>structure holding the file header and module header</td>
</tr>
<tr>
<td>data</td>
<td>a structure containing standard data</td>
</tr>
<tr>
<td>debug</td>
<td>logical flag to show more info on errors</td>
</tr>
</tbody>
</table>

Value
a structure containing data from a single object, and a logical flag if an error has occurred

Author(s)
Taiki Sakai <taiki.sakai@noaa.gov>
readClickBackground

*Read Click Detector Background data*

**Description**

Reads in the background data saved by the Click Detector

**Usage**

```
readClickBackground(fid, fileInfo, data)
```

**Arguments**

- **fid**: binary file identifier
- **fileInfo**: structure holding the file header, module header, and the appropriate function to read module specific data
- **data**: a structure containing standard data

**Value**

a structure containing data from a single object

**Author(s)**

Michael Oswald <taiki.sakai@noaa.gov>

---

readClickClsfrAnnotation

*Read Click Classifier Annotation*

**Description**

Reads binary data stored by Click Classifier annotations

**Usage**

```
readClickClsfrAnnotation(fid, fileInfo, debug = FALSE)
```

**Arguments**

- **fid**: binary file identifier
- **fileInfo**: structure holding the file header and module header
- **debug**: logical flag to show more info on errors
**readClickData**

**Value**

a vector of click classifiers, represented by the click type flag

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>

---

**readClickData**  
*Read Click Data*

**Description**

Reads binary data stored by the Click Detector module.

**Usage**

```r
readClickData(
  fid,
  fileInfo,
  data,
  skipLarge = FALSE,
  debug = FALSE,
  getWave,
  onlyWave
)
```

**Arguments**

- **fid**  
  binary file identifier
- **fileInfo**  
  structure holding the file header and module header
- **data**  
  a structure containing standard data
- **skipLarge**  
  a flag for whether or not to skip reading large wave file
- **debug**  
  logical flag to show more info on errors
- **getWave**  
  DEPRECATED: see skipLarge
- **onlyWave**  
  DEPRECATED: see skipLarge

**Value**

a structure containing data from a single object, and a logical flag if an error has occurred

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>
**readClickFooter**  
*Read Click Footer*

**Description**

Reads module footer information for the Click Detector module. Note that sometimes there is no additional footer information, so check first whether or not the binaryLength variable is 0.

**Usage**

```r
readClickFooter(file)
```

**Arguments**

- `file`  
  Binary file to be read

**Value**

Footer information for Click Detector module

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>

---

**readClickTriggerData**  
*Read Click Trigger Level*

**Description**

Reads binary data stored by the click detector trigger

**Usage**

```r
readClickTriggerData(fid, fileInfo, data, debug = FALSE, ...)
```

**Arguments**

- `fid`  
  Binary file identifier
- `fileInfo`  
  Structure holding the file header and module header
- `data`  
  A structure containing standard data
- `debug`  
  Logical flag to show more info on errors
- `...`  
  Arguments passed to other functions

**Value**

A structure containing data from a single object, and a logical flag if an error has occurred
**readClickTriggerHeader**

*Read Click Trigger Header*

**Description**
Reads file header information specific to the click trigger module.

**Usage**
readClickTriggerHeader(file)

**Arguments**
- file: binary file to be read

**Value**
header information for the click trigger

**Author(s)**
Taiki Sakai <taiki.sakai@noaa.gov>

---

**readClipData**

*Read Clip Data*

**Description**
Reads binary data stored by the Clip Generator module.

**Usage**
readClipData(fid, fileInfo, data, debug = FALSE, ...)

**Arguments**
- fid: binary file identifier
- fileInfo: structure holding the file header and module header
- data: a structure containing standard data
- debug: logical flag to show more info on errors
- ...: Arguments passed to other functions
readDbHtData

Value

a structure containing data from a single object, and a logical flag if an error has occurred

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

Description

Reads binary data stored by the DbHt module.

Usage

readDbHtData(fid, fileInfo, data, debug = FALSE, ...)

Arguments

- **fid**: binary file identifier
- **fileInfo**: structure holding the file header and module header
- **data**: a structure containing standard data
- **debug**: logical flag to show more info on errors
- ... Arguments passed to other functions

Value

a structure containing data from a single object, and a logical flag if an error has occurred

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>
**readDifarData**

**Description**
Reads binary data stored by the Difar Processing module.

**Usage**
```r
code = readDifarData(fid, fileInfo, data, skipLarge = FALSE, debug = FALSE)
```

**Arguments**
- `fid`: binary file identifier
- `fileInfo`: structure holding the file header and module header
- `data`: a structure containing standard data
- `skipLarge`: a flag of whether or not to skip reading the waveform
- `debug`: logical flag to show more info on errors

**Value**
a structure containing data from a single object, and a logical flag if an error has occurred

**Author(s)**
Taiki Sakai <taiki.sakai@noaa.gov>

---

**readFileFooterInfo**

**Description**
Reads in the binary file footer. The input variable version is the file format read in from the file header. As of version 3, the file footer includes the lowest and highest UID values in the file.

**Usage**
```r
results = readFileFooterInfo(fid, version)
```

**Arguments**
- `fid`: binary file to be read
- `version`: binary file version
readGPLDetections

Value
footer information common to all files

Author(s)
Taiki Sakai <taiki.sakai@noaa.gov>

readFileHeader  
Read File Header

Description
Reads file header information common to all files

Usage
readFileHeader(file, readExtra = FALSE)

Arguments
file binary file to be read
readExtra flag if there is extra information to read

Value
header information common to all files

Author(s)
Taiki Sakai <taiki.sakai@noaa.gov>

readGPLDetections  
Read GPL Detections

Description
Reads binary data stored by the GPL Module.

Usage
readGPLDetections(fid, fileInfo, data, debug = FALSE, ...)
**Arguments**

- **fid**  
  binary file identifier
- **fileInfo**  
  structure holding the file header and module header
- **data**  
  a structure containing standard data
- **debug**  
  logical flag to show more info on errors
- **...**  
  Arguments passed to other functions

**Value**

a structure containing data from a single object, and a logical flag if an error has occurred

**Author(s)**

Michael Oswald <mo55@st-andrews.ac.uk>

---

**Description**

Reads a Java UTF-8 string. The first 2 bytes are the length of the string, then the string itself.

**Usage**

```plaintext
readJavaUTFString(file)
```

**Arguments**

- **file**  
  binary file to be read

**Value**

the string and its length

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>
### readLTSAData  
**Read LTSA Data**

**Description**

Reads binary data stored by the LTSA module.

**Usage**

`readLTSAData(fid, fileInfo, data, debug = FALSE, ...)`

**Arguments**

- `fid`: binary file identifier
- `fileInfo`: structure holding the file header and module header
- `data`: a structure containing standard data
- `debug`: logical flag to show more info on errors
- `...`: Arguments passed to other functions

**Value**

a structure containing data from a single object, and a logical flag if an error has occurred

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>

---

### readLTSAHeader  
**Read LTSA Header**

**Description**

Reads file header information specific to the LTSA module

**Usage**

`readLTSAHeader(file)`

**Arguments**

- `file`: binary file to be read

**Value**

header information for the LTSA module
Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

---

**Description**

Reads annotations from the matched click classifier. The matched click classifier annotates click detections with a threshold, matchcorr and rejectcorr values. The threshold value is used in the binary classification process. If it exceeds a hard value then the click is classified with the set type. The matchcorr and rejectcorr values are simply the correlation values of the match and reject templates with the click.

**Usage**

```r
readMatchClsfrAnnotation(fid, fileInfo, anVersion, debug = FALSE)
```

**Arguments**

- `fid` binary file identifier
- `fileInfo` structure holding the file header and module header
- `anVersion` version id of annotation module
- `debug` logical flag to show more info on errors

**Value**

A vector with the threshold, matchcorr, and rejectcorr values. See description.

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>
### readNoiseBandData  
**Read Noise Band Data**

**Description**

Reads binary data stored by the Noise Band Monitor.

**Usage**

```r
readNoiseBandData(fid, fileInfo, data, debug = FALSE, ...)
```

**Arguments**

- `fid`  
  binary file identifier

- `fileInfo`  
  structure holding the file header and module header

- `data`  
  a structure containing standard data

- `debug`  
  logical flag to show more info on errors

- `...`  
  Arguments passed to other functions

**Value**

a structure containing data from a single object, and a logical flag if an error has occurred

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>

### readNoiseMonData  
**Read Noise Monitor Data**

**Description**

Reads binary data stored by the Noise Monitor.

**Usage**

```r
readNoiseMonData(fid, fileInfo, data, debug = FALSE, ...)
```

**Arguments**

- `fid`  
  binary file identifier

- `fileInfo`  
  structure holding the file header and module header

- `data`  
  a structure containing standard data

- `debug`  
  logical flag to show more info on errors

- `...`  
  Arguments passed to other functions
readNoiseMonHeader

Value

a structure containing data from a single object, and a logical flag if an error has occurred

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

readNoiseMonHeader  Read Noise Monitor Header

Description

Reads file header information specific to the Noise Monitor module

Usage

readNoiseMonHeader(file)

Arguments

file  binary file to be read

Value

header information for the Noise Monitor module

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

readPamData  Read Pamguard Data

Description

Reads in the object data that is common to all modules. This reads up to (but not including) the object binary length, and then calls a function to read the module-specific data.

Usage

readPamData(fid, fileInfo, skipLarge, debug = FALSE, keepUIDs, ...)

readRWEDetectorData

Arguments

- `fid` binary file identifier
- `fileInfo` structure holding the file header, module header, and the appropriate function to read module specific data
- `skipLarge` Should we skip large parts of binaries? Currently only applicable to whistle, click, and DIFAR data
- `debug` logical flag to show more info on errors
- `keepUIDs` If not NULL, a vector of UIDs to read. All UIDs not in this vector will not be read.
- ... Arguments passed to other functions

Value

a structure containing data from a single object

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

Description

Reads binary data stored by the Right Whale Edge Detector.

Usage

`readRWEDetectorData(fid, fileInfo, data, debug = FALSE, ...)`

Arguments

- `fid` binary file identifier
- `fileInfo` structure holding the file header and module header
- `data` a structure containing standard data
- `debug` logical flag to show more info on errors
- ... Arguments passed to other functions

Value

a structure containing data from a single object, and a logical flag if an error has occurred

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>
readSpectralBackground

*Read Spectral Background data*

**Description**

Reads in the background data saved by various detectors (e.g. WMD, Right Whale Edge Detector, etc) EXCEPT FOR the Click Detector

**Usage**

readSpectralBackground(fid, fileInfo, data)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fid</td>
<td>binary file identifier</td>
</tr>
<tr>
<td>fileInfo</td>
<td>structure holding the file header, module header, and the appropriate function to read module specific data</td>
</tr>
<tr>
<td>data</td>
<td>a structure containing standard data</td>
</tr>
</tbody>
</table>

**Value**

a structure containing data from a single object

**Author(s)**

Michael Oswald <taiki.sakai@noaa.gov>

readStdModuleFooter

*Read Standard Module Footer*

**Description**

Reads the module footer information common to all modules. Differs from the legacy code in that it does not read in or skip any information specific to a module.

**Usage**

readStdModuleFooter(file)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>binary file to be read</td>
</tr>
</tbody>
</table>
Value

footer information common to all modules

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

---

readStdModuleHeader  Read Standard Module Header

Description

Reads the module header information common to all modules. Differs from the legacy code in that it does not read in or skip any information specific to a module.

Usage

readStdModuleHeader(file)

Arguments

file binary file to be read

Value

header information common to all modules

Author(s)

Taiki Sakai <taiki.sakai@noaa.gov>

---

readWMDData  Read Whistle and Moan Data

Description

Reads binary data stored by the Whistle & Moan Detector

Usage

readWMDData(fid, fileInfo, data, skipLarge = FALSE, debug = FALSE)
**Arguments**

- **fid**
  - Binary file identifier

- **fileInfo**
  - Structure holding the file header and module header

- **data**
  - A structure containing standard data

- **skipLarge**
  - A flag for whether or not to skip reading large contours

- **debug**
  - Logical flag to show more info on errors

**Value**

A structure containing data from a single object, and a logical flag if an error has occurred

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>

---

**readWMDHeader**

*Read Whistle & Moan Detector Header*

**Description**

Reads file header information specific to the Whistle & Moan Detector module

**Usage**

```plaintext
readWMDHeader(file)
```

**Arguments**

- **file**
  - Binary file to be read

**Value**

Header information for the Whistle & Moan Detector module

**Author(s)**

Taiki Sakai <taiki.sakai@noaa.gov>
Index

combineBackgroundNoise (loadBackgroundNoise), 5
contourToFreq, 2
convertPgDate, 3
countChannels, 4
dateNumToMillis, 5
loadBackgroundNoise, 5, 6
loadPanguardBinaryFile, 6, 6, 9
millisToDateNum, 8
pamBinRead, 8
pbToDf, 9
plotBackgroundNoise (loadBackgroundNoise), 5
plotWMD, 10
readAISData, 11
readClickBackground, 12
readClickClsfrAnnotation, 12
readClickData, 13
readClickFooter, 14
readClickTriggerData, 14
readClickTriggerHeader, 15
readClipData, 15
readDbHtData, 16
readDifarData, 17
readFileFooterInfo, 17
readFileHeader, 18
readGPLDetections, 18
readJavaUTFString, 19
readLTSAData, 20
readLTSAHeader, 20
readMatchClsfrAnnotation, 21
readNoiseBandData, 22
readNoiseMonData, 22
readNoiseMonHeader, 23
readPamData, 23
readRWEDetectorData, 24
readSpectralBackground, 25
readStdModuleFooter, 25
readStdModuleHeader, 26
readWMDData, 26
readWMDHeader, 27