

Package ‘LumReader’

January 27, 2017

Type Package

Title TL/OSL Reader Simulator

Version 0.1.0

Date 2017-01-25

Maintainer David Strebler <david.strebler@uni-koeln.de>

Description A series of functions to estimate the detection windows of a luminescence reader based on the filters and the photomultiplier (PMT) selected. These functions also allow to simulate a luminescence experiment based on the thermoluminescence (TL) or the optically stimulated luminescence (OSL) properties of a material.

License GPL-3

Encoding UTF-8

Depends R (>= 3.3.2)

Imports tools, methods, lattice, gridExtra, plotly, shiny

Collate 'Filter-class.R' 'create_Filter.R' 'import_Filter.R'
'export_Filter.R' 'plot_Filter.R' 'combine_Filters.R'
'FilterStack-class.R' 'create_FilterStack.R' 'add_Filter.R'
'remove_Filter.R' 'plot_FilterStack.R' 'default_Filters.R'
'PMT-class.R' 'create_PMT.R' 'import_PMT.R' 'export_PMT.R'
'default_PMT.R' 'plot_PMT.R' 'Stimulation-class.R'
'create_Stimulation.R' 'import_Stimulation.R'
'export_Stimulation.R' 'default_Stimulation.R'
'plot_Stimulation.R' 'Reader-class.R' 'create_Reader.R'
'plot_Reader.R' 'Material-class.R' 'create_Material.R'
'import_Material.R' 'export_Material.R' 'default_Material.R'
'plot_Material.R' 'Experiment-class.R' 'create_Experiment.R'
'plot_Experiment.R' 'LumReader-package.R'
'call_ShinyLumReader.R'

RoxygenNote 5.0.1

NeedsCompilation no

Author David Strebler [aut, cre],
Dominik Brill [ths],
Helmut Brückner [ths]

Repository CRAN

Date/Publication 2017-01-27 10:17:39

R topics documented:

LumReader-package	3
add_Filter	4
call_ShinyLumReader	5
combine_Filters	5
create_Experiment	6
create_Filter	7
create_FilterStack	8
create_Material	9
create_PMT	11
create_Reader	12
create_Stimulation	13
default_Filters	14
default_Material	15
default_PMT	16
default_Stimulation	16
Experiment-class	17
export_Filter	18
export_Material	19
export_PMT	20
export_Stimulation	21
Filter-class	22
FilterStack-class	23
import_Filter	24
import_Material	25
import_PMT	25
import_Stimulation	26
Material-class	27
plot_Experiment	28
plot_Filter	29
plot_FilterStack	29
plot_Material	30
plot_PMT	31
plot_Reader	31
plot_Stimulation	32
PMT-class	33
Reader-class	34
remove_Filter	34
Stimulation-class	35

Index

36

LumReader-package *Tools to simulate a TL/OSL Reader*

Description

A series of functions to estimate the detection windows of a luminescence reader based on the filters and the photomultiplier (PMT) selected. These functions also allow to simulate a luminescence experiment based on the thermoluminescence (TL) or the optically stimulated luminescence (OSL) properties of a material. For those only interested by filter combination, the package can provide the transmission property of a filter stack. The user can use the filters, detection units, stimulation units and materials already included in the package or create its own.

Details

Package: TLdating
Type: Package
Version: 0.1.0
Date: 2017-01-25
License: GPL-3

Author(s)

Authors

David Strebler, University of Cologne, Germany

Supervisor

Helmut Brückner, University of Cologne, Germany
Dominik Brill, University of Cologne, Germany

Project source code repository

<https://github.com/dstreble/LumReader>

Package maintainer

David Strebler, Geographisches Institut, Universität zu Köln, Cologne, Germany.
<david.strebler@uni-koeln.de>

Acknowledgement

This project is realized in the context of the CRC 806 “Our Way to Europe” (<http://www.sfb806.uni-koeln.de/>) which is funded by the German Research Foundation (DFG).

add_Filter	<i>Function to add a filter to a stack</i>
------------	--

Description

This function adds a filter to a stack

Usage

```
add_Filter(object, filter)
```

Arguments

object	FilterStack from which the filter has to be added
filter	Filter that has to be added

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
#create filters
filter1 <- default_Filters('example')[[1]]
filter2 <- default_Filters('example2')[[1]]

#Create filterStack
name <- "example"
description <- "non realistic filterStack"

filters <- list(filter1)
filterStack <- create_FilterStack(name, description, filters)

plot_FilterStack(filterStack)

#add_Filter
filterStack <- add_Filter(filterStack, filter2)

plot_FilterStack(filterStack)
```

call_ShinyLumReader *ShinyLumReader application*

Description

This function calls the 'shinyLumReader' application.

Usage

```
call_ShinyLumReader()
```

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
## Not run:  
call_ShinyLumReader()  
  
## End(Not run)
```

combine_Filters *Function to combine two filters*

Description

This function creates a new filter out of 2 filters.

Usage

```
combine_Filters(filter1, filter2)
```

Arguments

filter1 **Filter** First of the two filters to combine.
filter2 **Filter** Second of the two filters to combine.

Value

The function creates a new **Filter** object.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
filter

#create filters
filter1 <- default_Filters('example')[[1]]
filter2 <- default_Filters('example2')[[1]]

new.filter <- combine_Filters (filter1, filter2)

plot_Filter(new.filter)
```

create_Experiment *Function to create a Experiment.*

Description

This function create a new Experiment.

Usage

```
create_Experiment(name, description, reader, material, type = NULL,
  interval = NULL)
```

Arguments

name	character : Name of the Experiment.
description	character : Description of the Experiment.
reader	Reader : Reader used for the Experiment.
material	Material : Material used for the Experiment.
type	character : type of experiment ('TL' or 'OSL').
interval	numeric : Temperature or wavelength on which the experiment focuses on.

Value

This function return a new Experiment.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
# Create info
name <- 'example'
description <- 'example'

# Create reader components
filter <- default_Filters('example')
filterStack <- create_FilterStack(name, description, filter)
stimulation <- default_Stimulation('example')
PMT <- default_PMT('example')

# Create reader
reader <- create_Reader(name, description, stimulation, filterStack, PMT)

# Create material
material <- default_Material('example')

experiment <- create_Experiment(name, description, reader, material, 'OSL')

plot_Experiment(experiment)
```

create_Filter *Function to create a filter.*

Description

This function create a new filter.

Usage

```
create_Filter(name, description, reference.thickness, thickness = NULL,
             reflexion, transmission)
```

Arguments

name	character: Name of the filter.
description	character: Description of the filter.
reference.thickness	numeric: Reference thickness of the filter.
thickness	numeric: Thickness of the filter (by default thickness = reference.thickness).
reflexion	numeric: Reflexion of the filter (1-P).
transmission	numeric: Transmission of the filter (T).

Value

This function return a new filter.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
#Data
wavelength <- seq(200,1000,10)

values <- dnorm(x = wavelength,mean = 400,sd = 50)*dnorm(x = wavelength,mean = 200,sd = 50)
values <- values/max(values)

name <- "example"
description <- "non realistic filter"
reference.thickness <- 1
reflexion <- 1
transmission <- matrix(data=c(wavelength,
                             values),
                       nrow = 81,
                       ncol = 2,
                       byrow = FALSE)

#Filter
filter <- create_Filter(name = name,
                       description = description,
                       reference.thickness = reference.thickness,
                       reflexion = reflexion,
                       transmission = transmission)

plot_Filter(filter)
```

create_FilterStack *Function to create a filter stack*

Description

This function create a new filter stack

Usage

```
create_FilterStack(name, description, filters)
```

Arguments

name **character:** Name of the filter stack.
description **character:** Description of the filter stack.
filters **list:** Filter included in the stack

Value

This function return a new filter stack.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
#Data
name <- "example"
description <- "non realistic filterStack"

filters <- default_Filters(c('example','example2'))

#Filterstack
filterstack <- create_FilterStack(name,description,filters)

plot_FilterStack(filterstack)
```

create_Material	<i>Function to create a Material.</i>
-----------------	---------------------------------------

Description

This function create a new Material.

Usage

```
create_Material(name, description.TL, description.OSL, TL, OSL)
```

Arguments

name	character: Name of the material.
description.TL	character: description of the material TL properties.
description.OSL	character: description of the material OSL properties.
TL	numeric: TL response of the Material [a.u.].
OSL	numeric: TL of the Material [a.u.].

Value

This function return a new Material.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
#Data
name <- "example"

# TL
description.TL <- "example"

TL.wavelength <- seq(200,1000,10)
TL.temperatures <- seq(0,800,10)

# TL peak
TL.peak.x <- dnorm(TL.wavelength,400,50)
TL.peak.x <- rep(TL.peak.x,each=length(TL.temperatures))

TL.peak.x <- TL.peak.x/max(TL.peak.x)

TL.peak.y <- dnorm(TL.temperatures,400,25)
TL.peak.y <- rep(TL.peak.y,times=length(TL.wavelength))
TL.peak.y <- TL.peak.y/max(TL.peak.y)

TL.signal <- TL.peak.x*TL.peak.y

TL <- matrix(data=c(rep(TL.wavelength,
                      each=length(TL.temperatures)),
                  rep(TL.temperatures,
                      times=length(TL.wavelength)),
                  TL.signal),
             nrow = length(TL.signal),
             ncol = 3,
             byrow = FALSE)

# OSL
description.OSL <- "example"

OSL.wavelength <- seq(200,1000,10)
OSL.color <- seq(200,1000,10)

OSL.peak.x <- dnorm(OSL.wavelength,300,100)
OSL.peak.x <- rep(OSL.peak.x,each=length(OSL.color))

OSL.peak.x <- OSL.peak.x/max(OSL.peak.x)

OSL.peak.y <- dnorm(OSL.color,500,50)
OSL.peak.y <- rep(OSL.peak.y,times=length(OSL.wavelength))
OSL.peak.y <- OSL.peak.y/max(OSL.peak.y)

OSL.signal <- OSL.peak.x*OSL.peak.y

OSL <- matrix(data=c(rep(OSL.wavelength,
                        each=length(OSL.color)),
                    rep(OSL.color,
```

```
        times=length(OSL.wavelength)),
        OSL.signal),
nrow = length(OSL.signal),
ncol = 3,
byrow = FALSE)

# Material

material <- create_Material(name = name,
                           description.TL = description.TL,
                           TL = TL,
                           description.OSL = description.OSL,
                           OSL = OSL)

plot_Material(material)
```

create_PMT

Function to create a PMT.

Description

This function create a new PMT.

Usage

```
create_PMT(name, description, efficiency)
```

Arguments

name **character**: Name of the PMT.
description **character**: Description of the PMT.
efficiency **numeric**: efficiency of the PMT [mA/W].

Value

This function return a new PMT.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
# Data
name <- "example"
description <- "non realistic PMT"

wavelength <- seq(200,1000,10)

values <- dnorm(x = wavelength,mean = 300,sd = 150)
values <- values/max(values)

efficiency <- matrix(data=c(wavelength,
                             values),
                      nrow = 81,
                      ncol = 2,
                      byrow = FALSE)

#PMT
PMT <- create_PMT(name = name,
                  description = description,
                  efficiency = efficiency)

plot_PMT(PMT)
```

create_Reader

Function to create a Reader.

Description

This function create a new Reader.

Usage

```
create_Reader(name, description, stimulation, filterStack, PMT)
```

Arguments

name	character: Name of the Reader.
description	character: Description of the Reader.
stimulation	numeric: Excitation source of the Reader.
filterStack	numeric: filter stack of the Reader.
PMT	numeric: PMT of the Reader.

Value

This function return a new Reader.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
# Data
name <- 'example'
description <- 'example'

# Create reader components
filter <- default_Filters('example')
filterStack <- create_FilterStack(name, description, filter)
stimulation <- default_Stimulation('example')
PMT <- default_PMT('example')

reader <- create_Reader(name, description, stimulation, filterStack, PMT)

plot_Reader(reader)
```

create_Stimulation *Function to create a Stimulation.*

Description

This function create a new Stimulation.

Usage

```
create_Stimulation(name, description, type, emission)
```

Arguments

name	character: Name of the Stimulation.
description	character: Description of the Stimulation.
type	character: Type of the Stimulation ('TL' or 'OSL').
emission	numeric: emission of the Stimulation [u.a].

Value

This function return a new Stimulation.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
# Data
name <- "example"
description <- "non realistic stimulation"

wavelength <- seq(200,1000,10)

values <- dnorm(x = wavelength,mean = 500,sd = 25)
values <- values/max(values)

emission <- matrix(data=c(wavelength,
                          values),
                   nrow = 81,
                   ncol = 2,
                   byrow = FALSE)

stimulation <- create_Stimulation(name = name,
                                 description = description,
                                 type = 'OSL',
                                 emission = emission)

plot_Stimulation(stimulation)
```

default_Filters *Fonction to upload the default filters.*

Description

This fonction generates a list containing the filters included in the packages

Usage

```
default_Filters(names, thickness = NULL)
```

Arguments

names **character** (with default): Names of the filters to import.
thickness **character** (with default): thickness of the filters to import.

Value

This function return a list containing the filters included in the package.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_Filters('example')[[1]]  
  
plot_Filter(example)
```

<code>default_Material</code>	<i>Fonction to upload the default Material.</i>
-------------------------------	---

Description

This fonction generates a list containing the Material included in the packages

Usage

```
default_Material(name)
```

Arguments

name **character** (with default): name of the Material to import.

Value

This function return a list containing the Material included in the package.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_Material('example')  
  
plot_Material(example)
```

default_PMT *Fonction to upload the default PMT.*

Description

This fonction generates a list containing the PMT included in the packages

Usage

```
default_PMT(name)
```

Arguments

name **character** (with default): name of the PMT to import.

Value

This function return a list containing the PMT included in the package.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_PMT('example')  
  
plot_PMT(example)
```

default_Stimulation *Fonction to upload the default Stimulation.*

Description

This fonction generates a list containing the Stimulation included in the packages

Usage

```
default_Stimulation(name)
```

Arguments

name **character** (with default): name of the Stimulation to import.

Value

This function return a list containing the Stimulation included in the package.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_Stimulation('example')  
  
plot_Stimulation(example)
```

Experiment-class	<i>Class Experiment</i>
------------------	-------------------------

Description

Class Experiment contains the characteristics of a luminescence experiment.

Method setExperiment

Method getExperiment

Usage

```
## S4 method for signature 'Experiment'  
show(object)  
  
setExperiment(name, description, reader, material, type, interval)  
  
## S4 method for signature  
## 'character,character,Reader,Material,character,numeric'  
setExperiment(name,  
  description, reader, material, type, interval)  
  
getExperiment(object, ref)  
  
## S4 method for signature 'Experiment,character'  
getExperiment(object, ref)
```

Arguments

object	Experiment : Experiment.
name	character : Name of the luminescence experiment.
description	character : Description of the experiment.

reader	Reader: TL/OSL reader unit used for the experiment.
material	Material: Material analysed by the experiment.
type	character: Type of experiment ('TL' or 'OSL'). By default, defined by the stimulation unit of the reader.
interval	numeric: vector defining the stimulation temperature (for 'TL') or wavelength (for 'OSL') interval on which the experiment focuses.
ref	character: Slot reference.

Slots

name	character: Name of the luminescence experiment.
description	character: Description of the experiment.
reader	Reader: TL/OSL reader unit used for the experiment.
material	Material: Material analysed by the experiment.
type	character: Type of experiment ('TL' or 'OSL'). By default, defined by the stimulation unit of the reader.
interval	numeric: vector defining the stimulation temperature (for 'TL') or wavelength (for 'OSL') interval on which the experiment focuses.
emission	Stimulation: Material emission spectra for the temperature or wavelength interval on which the experiment focuses.
detected	Stimulation: Emission spectra which is detected by the reader.

Author(s)

David Strebler

export_Filter *Function to export a filter*

Description

This function exports a filter as a .FLT file.

Usage

```
export_Filter(object, file.name)
```

Arguments

object	Filter to export
file.name	character name of the .FLT file that will contain the filter properties.

Value

The function creates a new txt file containing the filter properties.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_Filters('example')[[1]]
file.name <- 'example'

## Not run:
export_Filter(example, file.name) # uncomment

## End(Not run)

# There is now an 'example.FLT' file in the 'working directory'.
# This file is a classical .txt file despite the extension.

file <- paste(getwd(), '/', file.name, '.FLT', sep="")
print(file)
# readLines(file)
```

export_Material	<i>Function to export a Material</i>
-----------------	--------------------------------------

Description

This function exports a Material as a .Material file.

Usage

```
export_Material(object, file.name)
```

Arguments

object [Material](#) to export
file.name [character](#) name of the .FLT file that will contain the Material properties.

Value

The function creates a new txt file containing the Material properties.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_Material('example')
file.name <- 'example'

## Not run:
export_Material(example, file.name) # uncomment

## End(Not run)

# There is now an 'example.TL' and an 'example.OSL' file in the 'working directory'.
# These file are classical .txt files despite the extensions.

file.TL <- paste(getwd(), '/', file.name, '.TL', sep="")
print(file.TL)
# readLines(file.TL)

file.OSL <- paste(getwd(), '/', file.name, '.OSL', sep="")
print(file.OSL)
# readLines(file.OSL)
```

export_PMT

Function to export a PMT

Description

This function exports a PMT as a .PMT file.

Usage

```
export_PMT(object, file.name)
```

Arguments

object [PMT](#) to export
file.name [character](#) name of the .FLT file that will contain the PMT properties.

Value

The function creates a new txt file containing the PMT properties.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_PMT('example')
file.name <- 'example'

## Not run:
export_PMT(example, file.name) # uncomment

## End(Not run)

# There is now an 'example.PMT' file in the 'working directory'.
# This file is a classical .txt file despite the extension.

file <- paste(getwd(), '/', file.name, '.PMT', sep="")
print(file)
# readLines(file)
```

export_Stimulation *Function to export a Stimulation*

Description

This function exports a Stimulation as a .STI file.

Usage

```
export_Stimulation(object, file.name)
```

Arguments

object [Stimulation](#) to export
file.name [character](#) name of the .FLT file that will contain the Stimulation properties.

Value

The function creates a new txt file containing the Stimulation properties.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
example <- default_Stimulation('example')
file.name <- 'example'

## Not run:
export_Stimulation(example, file.name) # uncomment
```

```
## End(Not run)

# There is now an 'example.EXI' file in the 'working directory'.
# This file is a classical .txt file despite the extension.

file <- paste(getwd(), '/', file.name, '.EXI', sep="")
print(file)
# readLines(file)
```

Filter-class

Class Filter

Description

Class Filter contains the properties of a optical filter.

Method setFilter

Method getFilter

Usage

```
setFilter(name, description, reference.thickness, thickness, reflexion,
          reference.transmission)
```

```
## S4 method for signature 'character,character,numeric,numeric,numeric,matrix'
setFilter(name,
          description, reference.thickness, thickness, reflexion,
          reference.transmission)
```

```
getFilter(object, ref)
```

```
## S4 method for signature 'Filter,character'
getFilter(object, ref)
```

Arguments

name	character: Name of the filter
description	character: Description of the filter
reference.thickness	numeric: Reference thickness for the filter.
thickness	numeric: Actual filter thickness (by default, the reference thickness).
reflexion	numeric: Reflection of the filter (1-P) (between 0, which means that the signal is completely reflected, and 1, which means there is no reflection of the signal).
reference.transmission	matrix: Transmission matrix of the filter. The first column contains the wavelength [nm] and the second the transmission [0-1] at these wavelengths.
object	Filter: Filter
ref	character: Slot reference.

Slots

name **character**: Name of the filter

description **character**: Description of the filter

reference.thickness **numeric**: Reference thickness for the filter.

thickness **numeric**: Actual filter thickness (by default, the reference thickness).

reflexion **numeric**: Reflection of the filter (1-P) (between 0, which means that the signal is completely reflected, and 1, which means there is no reflection of the signal).

reference.transmission **matrix**: Transmission matrix of the filter for the reference.thickness. The first column contains the wavelength [nm] and the second the transmission [0-1] at these wavelengths.

transmission **matrix**: Transmission matrix of the filter. The first column contains the wavelength [nm] and the second the transmission [0-1] at these wavelengths.

Author(s)

David Strebler

FilterStack-class	<i>Class</i> FilterStack
-------------------	--------------------------

Description

Object class containing a combination of filters.

Method setFilterStack

Method getFilterStack

Usage

```
setFilterStack(name, description, filters)
```

```
## S4 method for signature 'character,character'
setFilterStack(name, description, filters)
```

```
getFilterStack(object, ref)
```

```
## S4 method for signature 'FilterStack,character'
getFilterStack(object, ref)
```

Arguments

name **character**: Name of the filter stack.

description **character**: Description of the filter stack.

filters **list**: List of the **Filter** which are in the filter stack.

object **FilterStack**: FilterStack

ref **character**: FilterStack slot.

Slots

name **character**: Name of the filter stack.
description **character**: Description of the filter stack.
filters **list**: List of the **Filter** which are in the filter stack.
bunch **Filter**: Properties of the complete filter stack.

Author(s)

David Strebler

import_Filter	<i>Function to import a filter</i>
---------------	------------------------------------

Description

This function import the properties of a filter previously saved in a .FLT file.

Usage

```
import_Filter(file.name, thickness = NULL)
```

Arguments

file.name **character** (required): name of the .FLT file containing the filter properties.
thickness **numeric** (default): Thickness of the filter (by default thickness = reference thickness).

Value

The function creates a new **Filter** object.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
folder <- system.file("extdata", package="LumReader")  
file.name <- 'example' # !!! no extension !!! #  
file <- paste(folder, '/', file.name, sep="")  
example <- import_Filter(file)  
plot_Filter(example)
```

import_Material	<i>Function to import a Material</i>
-----------------	--------------------------------------

Description

This function import the properties of a Material previously saved in a .Material file.

Usage

```
import_Material(file.name)
```

Arguments

file.name **character** name of the .FLT file containing the Material properties.

Value

The function creates a new **Material** object.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
folder <- system.file("extdata", package="LumReader")  
file.name <- 'example' # !!! no extension !!! #  
file <- paste(folder, '/', file.name, sep="")  
example <- import_Material(file)  
plot_Material(example)
```

import_PMT	<i>Function to import a PMT</i>
------------	---------------------------------

Description

This function import the properties of a PMT previously saved in a .PMT file.

Usage

```
import_PMT(file.name)
```

Arguments

file.name **character** name of the .FLT file containing the PMT properties.

Value

The function creates a new **PMT** object.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
folder <- system.file("extdata", package="LumReader")  
  
file.name <- 'example' # !!! no extension !!! #  
  
file <- paste(folder, '/', file.name, sep="")  
  
example <- import_PMT(file)  
  
plot_PMT(example)
```

import_Stimulation *Function to import a Stimulation*

Description

This function import the properties of a Stimulation previously saved in a .EXI file.

Usage

```
import_Stimulation(file.name)
```

Arguments

file.name **character** name of the .FLT file containing the Stimulation properties.

Value

The function creates a new **Stimulation** object.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```

folder <- system.file("extdata", package="LumReader")

file.name <- 'example' # !!! no extension !!! #

file <-paste(folder, '/', file.name, sep="")

example <- import_Stimulation(file)

plot_Stimulation(example)

```

Material-class	<i>Class Material</i>
----------------	-----------------------

Description

Class *Material* contains the luminescence properties of a material.

Method `setMaterial`

Method `getMaterial`

Usage

```
setMaterial(name, description.TL, description.OSL, TL, OSL)
```

```
## S4 method for signature 'character,character,character,matrix,matrix'
setMaterial(name,
  description.TL, description.OSL, TL, OSL)
```

```
getMaterial(object, ref)
```

```
## S4 method for signature 'Material,character'
getMaterial(object, ref)
```

Arguments

`name` **character**: Name of the material.

`description.TL` **character**: Description of the TL properties of the material.

`description.OSL`

character: Description of the OSL properties of the material.

`TL` **matrix**: TL emission properties of the material. The first column contains the emission wavelength [nm], the second column contains the stimulation temperature [°C] and the third column contains the emission intensity [a.u].

`OSL` **matrix**: OSL emission properties of the material. The first column contains the emission wavelength [nm], the second column contains the stimulation wavelength [nm] and the third column contains the emission intensity [a.u].

`object` **Material**: Material.

`ref` **character**: Material slot.

Slots

name **character**: Name of the material.

description.TL **character**: Description of the TL properties of the material.

description.OSL **character**: Description of the OSL properties of the material.

TL **matrix**: TL emission properties of the material. The first column contains the emission wavelength [nm], the second column contains the stimulation temperature [°C] and the third column contains the emission intensity [a.u].

OSL **matrix**: OSL emission properties of the material. The first column contains the emission wavelength [nm], the second column contains the stimulation wavelength [nm] and the third column contains the emission intensity [a.u].

Author(s)

David Strebler

plot_Experiment *Function to plot a Experiment*

Description

This function plots the selected Experiment.

Usage

```
plot_Experiment(object)
```

Arguments

object **Experiment** to plot

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
# Create info
name <- 'example'
description <- 'example'

# Create reader components
filter <- default_Filters('example')
filterStack <- create_FilterStack(name, description, filter)
stimulation <- default_Stimulation('example')
PMT <- default_PMT('example')

# Create reader
```

```
reader <- create_Reader(name, description, stimulation, filterStack, PMT)

# Create material
material <- default_Material('example')

experiment <- create_Experiment(name, description, reader, material, 'OSL')

plot_Experiment(experiment)
```

plot_Filter *Function to plot a filter*

Description

This function plots the selected filter.

Usage

```
plot_Filter(object)
```

Arguments

object [Filter](#) to plot

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
filters <- default_Filters(c('example', 'example2'))

plot_Filter(filters[[1]])
```

plot_FilterStack *Function to plot a filter stack.*

Description

This function plots a filter stack.

Usage

```
plot_FilterStack(object)
```

Arguments

object [FilterStack](#) to plot.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
#Data
name <- "example"
description <- "non realistic filterStack"

filters <- default_Filters(c('example','example2'))

#Filterstack
filterstack <- create_FilterStack(name,description,filters)

plot_FilterStack(filterstack)
```

plot_Material

Function to plot a Material

Description

This function plots the selected Material.

Usage

```
plot_Material(object)
```

Arguments

object [Material](#) to plot

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
material <- default_Material('example')

plot_Material(material)
```

plot_PMT	<i>Function to plot a PMT</i>
----------	-------------------------------

Description

This function plots the selected PMT.

Usage

```
plot_PMT(object)
```

Arguments

object [PMT](#) to plot

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
PMT <- default_PMT('example')  
plot_PMT(PMT)
```

plot_Reader	<i>Function to plot a filter stack.</i>
-------------	---

Description

This function plots a filter stack.

Usage

```
plot_Reader(object)
```

Arguments

object [Reader](#) to plot.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
# Data
name <- 'example'
description <- 'example'

# Create reader components
filter <- default_Filters('example')
filterStack <- create_FilterStack(name, description, filter)
stimulation <- default_Stimulation('example')
PMT <- default_PMT('example')

reader <- create_Reader(name, description, stimulation, filterStack, PMT)

plot_Reader(reader)
```

plot_Stimulation	<i>Function to plot a Stimulation</i>
------------------	---------------------------------------

Description

This function plots the selected Stimulation.

Usage

```
plot_Stimulation(object)
```

Arguments

object [Stimulation](#) to plot

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
stimulation <- default_Stimulation('example')

plot_Stimulation(stimulation)
```

PMT-class

Class PMT

Description

Object class containing the properties of a photomultiplier tube (PMT).

Method setPMT

Method getPMT

Usage

```
setPMT(name, description, efficiency)
```

```
## S4 method for signature 'character,character,matrix'  
setPMT(name, description, efficiency)
```

```
getPMT(object, ref)
```

```
## S4 method for signature 'PMT,character'  
getPMT(object, ref)
```

Arguments

name **character**: Name of the PMT

description **character**: Description of the PMT

efficiency **matrix**: Quantum efficiency of the PMT. The first column contains the wavelength [nm] and the second column the corresponding quantum efficiency [0-1].

object **PMT**: PMT.

ref **character**: Material slot.

Slots

name **character**: Name of the PMT

description **character**: Description of the PMT

efficiency **matrix**: Quantum efficiency of the PMT. The first column contains the wavelength [nm] and the second column the corresponding quantum efficiency [0-1].

Author(s)

David Strebler

Reader-class	<i>Class Reader</i>
--------------	---------------------

Description

Object class containing the properties of aPhotomultiplier tube.

Author(s)

David Strebler

remove_Filter	<i>Function to remove a filter from a stack</i>
---------------	---

Description

This function removes a filter from a stack

Usage

```
remove_Filter(object, filter)
```

Arguments

object	FilterStack from which the filter has to be removed.
filter	Filter that has to be removed.

Author(s)

David Strebler, University of Cologne (Germany).

Examples

```
#Create filterStack
name <- "example"
description <- "non realistic filterStack"

filters <- default_Filters(c('example','example2'))

filterStack <- create_FilterStack(name, description, filters)

plot_FilterStack(filterStack)

#Removal of the 2nd filter
filterStack <- remove_Filter(filterStack, filters[[2]])

plot_FilterStack(filterStack)
```

Stimulation-class *Class Stimulation*

Description

Class Stimulation contains the properties of a stimulation unit.

Method setStimulation

Method getStimulation

Usage

```
setStimulation(name, description, type, emission)
```

```
## S4 method for signature 'character,character,character,matrix'  
setStimulation(name,  
  description, type, emission)
```

```
getStimulation(object, ref)
```

```
## S4 method for signature 'Stimulation,character'  
getStimulation(object, ref)
```

Arguments

name	character : name of the stimulation unit.
description	character : description of the stimulation unit.
type	character : type of the stimulation ('TL' or 'OSL').
emission	matrix : Emission spectra of the stimulation unit. The first column contains the wavelength [nm] and the second the intensity of the signal [a.u].
object	Stimulation : Stimulation unit
ref	character : Slot reference.

Slots

name	character : name of the stimulation unit.
description	character : description of the stimulation unit.
type	character : type of the stimulation ('TL' or 'OSL').
emission	matrix : Emission spectra of the stimulation unit. The first column contains the wavelength [nm] and the second the intensity of the signal [a.u].

Author(s)

David Strebler

Index

*Topic **classes**

Experiment-class, [17](#)
Filter-class, [22](#)
Material-class, [27](#)
PMT-class, [33](#)
Reader-class, [34](#)

*Topic **package**

LumReader-package, [3](#)

[add_Filter](#), [4](#)

[call_ShinyLumReader](#), [5](#)
[character](#), [6–9](#), [11–28](#), [33](#), [35](#)
[combine_Filters](#), [5](#)
[create_Experiment](#), [6](#)
[create_Filter](#), [7](#)
[create_FilterStack](#), [8](#)
[create_Material](#), [9](#)
[create_PMT](#), [11](#)
[create_Reader](#), [12](#)
[create_Stimulation](#), [13](#)

[default_Filters](#), [14](#)
[default_Material](#), [15](#)
[default_PMT](#), [16](#)
[default_Stimulation](#), [16](#)

[Experiment](#), [17](#), [28](#)
[Experiment-class](#), [17](#)
[export_Filter](#), [18](#)
[export_Material](#), [19](#)
[export_PMT](#), [20](#)
[export_Stimulation](#), [21](#)

[Filter](#), [4](#), [5](#), [18](#), [22–24](#), [29](#), [34](#)
[Filter-class](#), [22](#)
[FilterStack](#), [4](#), [23](#), [30](#), [34](#)
[FilterStack-class](#), [23](#)

[getExperiment \(Experiment-class\)](#), [17](#)

[getExperiment, Experiment, character-method \(Experiment-class\)](#), [17](#)
[getExperiment, Experiment-method \(Experiment-class\)](#), [17](#)
[getFilter \(Filter-class\)](#), [22](#)
[getFilter, Filter, character-method \(Filter-class\)](#), [22](#)
[getFilter, Filter-method \(Filter-class\)](#), [22](#)
[getFilterStack \(FilterStack-class\)](#), [23](#)
[getFilterStack, FilterStack, character-method \(FilterStack-class\)](#), [23](#)
[getFilterStack, Material-method \(FilterStack-class\)](#), [23](#)
[getMaterial \(Material-class\)](#), [27](#)
[getMaterial, Material, character-method \(Material-class\)](#), [27](#)
[getMaterial, Material-method \(Material-class\)](#), [27](#)
[getPMT \(PMT-class\)](#), [33](#)
[getPMT, PMT, character-method \(PMT-class\)](#), [33](#)
[getPMT, PMT-method \(PMT-class\)](#), [33](#)
[getReader, Reader-method \(Reader-class\)](#), [34](#)
[getStimulation \(Stimulation-class\)](#), [35](#)
[getStimulation, Stimulation, character-method \(Stimulation-class\)](#), [35](#)
[getStimulation, Stimulation-method \(Stimulation-class\)](#), [35](#)

[import_Filter](#), [24](#)
[import_Material](#), [25](#)
[import_PMT](#), [25](#)
[import_Stimulation](#), [26](#)

[list](#), [8](#), [23](#), [24](#)
[LumReader \(LumReader-package\)](#), [3](#)
[LumReader-package](#), [3](#)

[Material](#), [6](#), [18](#), [19](#), [25](#), [27](#), [30](#)

Material-class, 27
 matrix, 22, 23, 27, 28, 33, 35
 numeric, 6, 7, 9, 11–13, 18, 22–24

 plot_Experiment, 28
 plot_Filter, 29
 plot_FilterStack, 29
 plot_Material, 30
 plot_PMT, 31
 plot_Reader, 31
 plot_Stimulation, 32
 PMT, 20, 26, 31, 33
 PMT-class, 33

 Reader, 6, 18, 31
 Reader-class, 34
 remove_Filter, 34

 setExperiment (Experiment-class), 17
 setExperiment, character, character, Reader, Material, character, numeric-method
 (Experiment-class), 17
 setExperiment, Experiment-method
 (Experiment-class), 17
 setFilter (Filter-class), 22
 setFilter, character, character, numeric, numeric, numeric, matrix-method
 (Filter-class), 22
 setFilter, Filter-method (Filter-class),
 22
 setFilterStack (FilterStack-class), 23
 setFilterStack, character, character-method
 (FilterStack-class), 23
 setFilterStack, FilterStack-method
 (FilterStack-class), 23
 setMaterial (Material-class), 27
 setMaterial, character, character, character, matrix, matrix-method
 (Material-class), 27
 setMaterial, Material-method
 (Material-class), 27
 setPMT (PMT-class), 33
 setPMT, character, character, matrix-method
 (PMT-class), 33
 setPMT, PMT-method (PMT-class), 33
 setReader, Reader-method (Reader-class),
 34
 setStimulation (Stimulation-class), 35
 setStimulation, character, character, character, matrix-method
 (Stimulation-class), 35
 setStimulation, Stimulation-method
 (Stimulation-class), 35

 show (Experiment-class), 17
 show, Experiment-method
 (Experiment-class), 17
 show, Reader-method (Reader-class), 34
 Stimulation, 18, 21, 26, 32, 35
 Stimulation-class, 35