Package 'RLumSTARR'

August 29, 2021

Type Package

Title Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE)

Version 0.1.0.9000-113

Author Sebastian Kreutzer [aut, cre] (<https://orcid.org/0000-0002-0734-2199>),
Dirk Mittelstrass [aut] (<https://orcid.org/0000-0002-9567-8791>)

Maintainer Sebastian Kreutzer <sebastian.kreutzer@aber.ac.uk>

Description Analysing spatially resolved radiofluorescence data using ImageJ in combination with Bayesian number-crunching.
A collection of functions to support early work on the subject.

License GPL-3

Depends R (>= 4.0),
utils,

Imports cli (>= 3.0.1),
coda (>= 0.19-1),
data.table (>= 1.14),
khroma (>= 1.6),
Luminescence (>= 0.9.8),
matrixStats (>= 0.60.1),
methods,
rjags (>= 4-10),

Suggests rmarkdown,
knitr

Encoding UTF-8

Language en-GB

LazyData true

VignetteBuilder knitr

RoxygenNote 7.1.1

R topics documented:

`RLumSTARR-package` *SpaTiAlly Resolved Radiofluorescence*

Description

A collection of functions to analyse spatially resolved radiofluorescence data

Details

Funding

- Sebastian Kreutzer received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 844457 (CREDit).

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom)
Dirk Mittelstrass (Germany) , RLum Developer Team

References

##TODO

`create_RFCurveArray` *Create Multidimensional Curve Arrays from RF File input*

Description

Helper function to create a multidimensional curve array based on RF-file input imported using the function `Luminescence::read_RF2R()` to prepare the Bayesian modelling process

Usage

```
create_RFCurveArray(files)
```

Arguments

`files`	list	(**required**): list of `.rf` files to be imported

Value

Returns a list of class `RLumSTARR_RFCurveArrary` with two arrays for the `RF_nat` and the `RF_reg` curve

Function version

0.1.0

How to cite

Kreutzer, S., 2021. create_RFCurveArray(): Create Multidimensional Curve Arrays from RF File input. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2021. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-113.

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom) , RLum Developer Team

See Also

Luminescence::read_RF2R

Examples

```
## list files using package external data
files <- list.files(system.file("extdata", "", package="RLumSTARR"), full.names=TRUE)

## create curve array
create_RFCurveArray(files = files)
```

---

extract_TRUELight — *Extract True Light from the Camera Measurements using a Bayesian Approach*

Description

A Bayesian modelling approach to extract the true light using the expanding region-of-interest (ROI) approach proposed by Cunningham and Clark-Balzan (2017). The function will return the results for a **single** curve from a **single** ROI

Usage

```
extract_TRUELight(
  data,
  element = c("RF_nat", "RF_reg"),
  ROI = 2,
  stepping = 1,
  method_control = list(),
  verbose = TRUE
)
```

Arguments

`data`	array (**required**): object created by create_RFCurveArray
`element`	character (*with default*): element from the input to be analysed, 'supported are only `RF_nat` or `RF_reg`

ROI	numeric (*optional*): ROI to be analysed, if nothing is given all ROIs are analysed, however, the first ROIS is discarded!
stepping	numeric (*with default*): stepping parameter that allows you to model only every xth (the value in stepping). This option can be extremely useful to play with data because it dramatically improves the modelling speed because less data are considered.
method_control	list (*optional*): parameter to be passed to rjags. Supported are n.chain, n.iter, thin, variable.names, model, see details for more.
verbose	logical (*with default*): enable/disable terminal feedback

Details

Method control

Supported options to be passed via the parameter method_control, most of them are used internally for the calls to rjags::jags.model and rjags::coda.samples.

PARAMETER	TYPE	DESCRIPTION
n.chain	numeric	the number of MCMC chains
n.iter	numeric	number of iterations for the MC runs
thin	numeric	thinning interval used for the monitoring
variable.names	character	variable names to monitor, alpha is always monitored
model	character	the bugs model

Value

Returns a list of class RLumSTARR_TRUELight with an the following elements:

...$RF_curve: Luminescence::RLum.Data.Curve object (the RF curve with the true light)

...$rjags_output: rjags::coda.samples output for further processing. *Note: Regardless the observed variable, the parameter alpha will always be used to create the curve*

...$model: the model used to run the Bayesian process, use writeLines to have nicely formatted terminal output

Function version

0.1.0

How to cite

Kreutzer, S., 2021. extract_TRUELight(): Extract True Light from the Camera Measurements using a Bayesian Approach. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2021. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-113.

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom) , RLum Developer Team

References

Cunningham, A.C., Clark-Balzan, L., 2017. Overcoming crosstalk in luminescence images of mineral grains. Radiation Measurements 106, 498–505. doi:10.1016/j.radmeas.2017.06.004

See Also

create_RFCurveArray, get_MCMCParameters

Examples

```
## list files using package external data
files <- list.files(system.file("extdata", "", package="RLumSTARR"), full.names=TRUE)

## create curve array
dat <- create_RFCurveArray(files = files)
output <- extract_TRUELight(
 data = dat,
 ROI = c(4),
 stepping = 10,
 verbose = FALSE,
 method_control = list(
   n.chain = 1,
   n.iter = 50,
   thin = 20))
```

---

get_MCMCParameters *Extracts a parameter from an MCMC list*

Description

Short cut to extract a parameter from an MCMC list or an object of class `RLumSTARR_TRUELight` created by the functions extract_TRUELight or run_TRUELightExtraction. In the latter case, the row names of the extracted matrices have the same dimension as the RF curve. If more processing is wanted, `'coda'` package should be used.

Usage

```
get_MCMCParameters(mcmc, parameter = NULL, prob = 0.95, unlist = TRUE)
```

Arguments

`mcmc`	coda::mcmc or coda::mcmc.list (**required**): input object, if created by extract_TRUELight the correct object is extracted automatically
`parameter`	character (*optional*): name of the parameter to be extracted. If `NULL` (the default) all found parameters are extracted and the result is a list with matrices of those parameter. If `parameter` is a vector, the function will try to extract the names parameters.
`prob`	numeric (*with default*): probability for the HPD calculation (cf. coda::HPDinterval)
`unlist`	logical (*with default*): if `TRUE` the output is a matrix of the means of the lower and upper intervals of the parameter. If the parameter was estimated based on multiple chains, this chains are also subject to an average calculation. If `FALSE` the output is a list as returned by coda::HPDinterval

Value

Returns a matrix with the parameter value or a named list with such matrices if `parameters` has a length > 1.

Function version

0.1.0

How to cite

Kreutzer, S., 2021. get_MCMCParameters(): Extracts a parameter from an MCMC list. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2021. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-113.

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom) , RLum Developer Team

See Also

coda::HPDinterval, extract_TRUELight

Examples

```
## load example files
files <- list.files(system.file("extdata", "", package="RLumSTARR"), full.names=TRUE)

##prepare data and run model
dat <- create_RFCurveArray(files = files)
output <-
 extract_TRUELight(
   data = dat,
   ROI = c(4),
   stepping = 60,
   verbose = FALSE,
   method_control = list(
   n.chain = 1,
   n.iter = 50,
   thin = 20))

##extract parameters
get_MCMCParameters(output)
```

`melt_RLumSTARR` *Melt output from RLumSTARR into simple data frames*

Description

The function provides a convenient way to convert the objects created by RLumSTARR into simple data.frames that can be processed conveniently by other functions for example the package `ggplot2`.

Usage

```
melt_RLumSTARR(x, ...)
```

Arguments

x	(**required**): input object of class RLumSTARR_RFCurveArray or RLumSTARR_TRUELight
...	further parameters, currently not used

Value

data.frame

Function version

0.1.0

How to cite

Kreutzer, S., 2021. melt_RLumSTARR(): Melt output from RLumSTARR into simple data frames. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2021. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-113.

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom) , RLum Developer Team

---

run_ImageJ *Run ImageJ SR-RF macro*

Description

The script runs the SR-RF ImageJ macro in batch mode out of R

Usage

```
run_ImageJ(
  path,
  RF_nat = "default",
  RF_reg = "default",
  bg_rm = "take_from_RF_reg",
  image_group_size = 5,
  image_alignment = TRUE,
  first_slices_rm = FALSE,
  noise_tolorance = 10,
  ROI_size = 10,
  center_x = 0.5,
  center_y = 0.5,
  diameter = 0.9,
  use_predefined_ROIs = FALSE,
  channel_time = 5,
```

```
  save_workflow_images = FALSE,
  save_additional_results = FALSE,
  save_signal_decay_videos = FALSE,
  offset_time = 0,
  .ImageJ = "/Applications/Fiji.app/Contents/MacOS/ImageJ-macosx"
)
```

Arguments

path	character (**required**): path to files to be analysed
RF_nat	character (*default*): name of the RF_nat file
RF_reg	character (*default*): name of the RF_reg file
bg_rm	character (*with default*): background subtraction options. Allowed are none (no background subtraction), take_from_RF_reg (takes the last 100 channels from the RF_reg signal: dangerous) or <your file name> (this does not work in batch mode)
image_group_size	numeric (*with default*): grouping value for running median to remove outliers
image_alignment	logical (*with default*): enable/disable image alignment
first_slices_rm	logical (*with default*): remove first slice of each curve set
noise_tolorance	numeric (*with default*): noise tolerance parameter
ROI_size	numeric (*with default*): ROI size in pixel
center_x	numeric (*with default*): aliquot ROI centre x-coordinate
center_y	numeric (*with default*): aliquot ROI centre y-coordinate
diameter	numeric (*with default*): relative diameter aliquot ROI
use_predefined_ROIs	logical (*with default*): use pre-defined ROIs imported from a file ROIs.zip found in the same folder as the files
channel_time	numeric (*with default*): channel time, this parameter was set the moment the sequence was written
save_workflow_images	logical (*with default*): enable/disable writing of additional workflow images
save_additional_results	logical (*with default*): enable/disable writing of additional workflow images
save_signal_decay_videos	logical (*with default*): enable/disable writing of additional workflow videos
offset_time	numeric (*with default*): offset time for the time axis
.ImageJ	numeric (*with default*): Path to ImageJ (the macro is shipped with the package)

Value

This functions returns the path of the analysed data

Function version

0.1.0

How to cite

Kreutzer, S., 2021. run_ImageJ(): Run ImageJ SR-RF macro. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2021. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-113.

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom) , RLum Developer Team

Examples

```
##TODO
```

---

run_TRUELightExtraction

Run Automated TRUE-Light Extraction over multiple datasets

Description

Runs [extract_TRUELight] automatically in a parallel processing mode using [parallel::mclapply]. In essence, the function calls [extract_TRUELight] and extracts the RF curves from the results and constructs an [Luminescence::RLum.Analysis-class] for further analyses.

Usage

```
run_TRUELightExtraction(
  data,
  ROI,
  stepping = 1,
  mc.cores = max(c(1, parallel::detectCores() - 2)),
  method_control = list(),
  include_jags_output = FALSE,
  verbose = TRUE
)
```

Arguments

`data`	[array] (**required**): object created by [create_RFCurveArray]
`ROI`	[numeric] (*optional*): ROIs to be analysed, if nothing is given all ROIs are analysed, however, the first ROIS is discarded!
`stepping`	[numeric] (*with default*): the stepping parameter from and to be passed to [extract_TRUELight]
`mc.cores`	[numeric] (*with default*): number of cores used for the processing, passed to [parallel::mclapply]
`method_control`	[list] (*optional*): parameters to be passed to [extract_TRUELight]

include_jags_output
: [logical] (*with default*): allows to include the output from JAGS as info object in the [Luminescence::RLum.Analysis-class] output objects for further diagnostics. Setting this option to 'TRUE' is not recommended for large datasets, since it will tremendously inflate the size of output and consume a lot of memory. If in-depth diagnostics are required, the function [extract_TRUELight] is recommended instead.

verbose
: [logical] (*with default*): enable/disable verbose mode. The output of the MCMC sampling using 'rjags' is **always** silent.

Value

The output is a [list] containing [Luminescence::RLum.Analysis-class] objects with two [Luminescence::RLum.Data.Cur class] objects for 'RF_nat' and 'RF_reg' respectively.

Function version

0.1.0

How to cite

Kreutzer, S., 2021. run_TRUELightExtraction(): Run Automated TRUE-Light Extraction over multiple datasets. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2021. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-113.

Author(s)

Sebastian Kreutzer, Geography & Earth Sciences, Aberystwyth University (United Kingdom) , RLum Developer Team

Examples

```
## list files using package external data
files <- list.files(system.file("extdata", "", package="RLumSTARR"), full.names=TRUE)
## create curve array
dat <- create_RFCurveArray(files = files)
output <- run_TRUELightExtraction(
data = dat,
stepping = 15,
mc.cores = 1,
ROI = 5,
verbose = TRUE,
method_control = list(
 n.chain = 1,
 n.iter = 50,
 thin = 20))
```

Index