

# Package ‘RLumSTARR’

May 10, 2022

**Type** Package

**Title** Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE)

**Version** 0.1.0.9000-132

**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@uni-heidelberg.de>

**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 (>= 3.2.0),  
coda (>= 0.19-1),  
data.table (>= 1.14),  
khroma (>= 1.8.0),  
Luminescence (>= 0.9.19),  
matrixStats (>= 0.61.0),  
methods,  
runjags (>= 2.2.1)

**Suggests** rmarkdown (>= 2.11),  
rjags (>= 4-12),  
testthat (>= 3.1.2),  
knitr (>= 1.37)

**Encoding** UTF-8

**Language** en-GB

**LazyData** true

**VignetteBuilder** knitr

**RoxygenNote** 7.1.2

## R topics documented:

RLumSTARR-package . . . . .	2
create_RFCurveArray . . . . .	2
extract_TRUELight . . . . .	3

get_MCMCParameters . . . . .	5
melt_RLumSTARR . . . . .	7
run_ImageJ . . . . .	8
run_TRUELightExtraction . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

RLumSTARR-package	<i>SpaTiAlly Resolved Radiofluorescence</i>
-------------------	---

---

## 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, Institute of Geography, Heidelberg University (Germany)  
Dirk Mittelstrass (Germany) , RLum Developer Team

## References

##TODO

---

create_RFCurveArray	<i>Create Multidimensional Curve Arrays from RF File input</i>
---------------------	--

---

## 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	<b>list (required):</b> list of .rf files to be imported. Alternatively you can pass an object created by <code>Luminescence::extract_ROI</code> . In this case only the first list element RF_nat is filled.
-------	---

## Value

Returns a list of class `RLumSTARR_RFCurveArray` with two arrays for the RF\_nat and the RF\_reg curve

**Function version**

0.1.0

**How to cite**

Kreutzer, S., 2022. create\_RFCurveArray(): Create Multidimensional Curve Arrays from RF File input. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2022. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-132.

**Author(s)**

Sebastian Kreutzer, Institute of Geography, Heidelberg University (Germany) , RLum Developer Team

**See Also**

[Luminescence::read\\_RF2R](#), [Luminescence::extract\\_ROI](#)

**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	<i>Extract True Light from the Camera Measurements using a Bayesian Approach</i>
-------------------	--

---

**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	<a href="#">array</a> ( <b>required</b> ): object created by <a href="#">create_RFCurveArray</a>
element	<a href="#">character</a> ( <i>with default</i> ): element from the input to be analysed, 'supported are only RF_nat or RF_reg
ROI	<a href="#">numeric</a> ( <i>optional</i> ): ROI to be analysed, if nothing is given all ROIs are analysed, however, the first ROIS is discarded!
stepping	<a href="#">numeric</a> ( <i>with default</i> ): 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	<a href="#">list</a> ( <i>optional</i> ): parameter to be passed to rjags. Supported are n.chain, n.iter, thin, variable.names, model, see details for more.
verbose	<a href="#">logical</a> ( <i>with default</i> ): 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 [runjags::run.jags](#) and [rjags::coda.samples](#).

## PARAMETER

n.chain

thin

burnin

sample

adapt

summarise

method

jags.refresh\tab [numeric] \tab0.1\tab refresh rate of update of the iteration process, select larger

*Note: The argument `model` allows to heavily modified the underlying model. To avoid crashes the paramters passed by `variable.names` will always be cross-checked against parameteres present in the model. Unknown parameters will be skipped!*

## 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

0.1.1

**How to cite**

Kreutzer, S., 2022. `extract_TRUELight()`: Extract True Light from the Camera Measurements using a Bayesian Approach. Function version 0.1.1. In: Kreutzer, S., Mittelstrass, D., 2022. `RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE)`. R package version 0.1.0.9000-132.

**Author(s)**

Sebastian Kreutzer, Institute of Geography, Heidelberg University (Germany) , `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 = 2,
    sample = 100,
    thin = 20))
```

---

<code>get_MCMCParameters</code>	<i>Extracts a parameter from an MCMC list</i>
---------------------------------	---

---

**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	<a href="#">coda::mcmc</a> or <a href="#">coda::mcmc.list</a> ( <b>required</b> ): input object, if created by <a href="#">extract_TRUELight</a> the correct object is extracted automatically
parameter	<a href="#">character</a> ( <i>optional</i> ): 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	<a href="#">numeric</a> ( <i>with default</i> ): probability for the HPD calculation (cf. <a href="#">coda::HPDInterval</a> )
unlist	<a href="#">logical</a> ( <i>with default</i> ): if TRUE the output is a <a href="#">matrix</a> 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 <a href="#">list</a> as returned by <a href="#">coda::HPDInterval</a>

**Value**

Returns a matrix with the parameter value or a named [list](#) with such matrices if parameters has a length > 1.

**Function version**

0.1.1

**How to cite**

Kreutzer, S., 2022. `get_MCMCParameters()`: Extracts a parameter from an MCMC list. Function version 0.1.1. In: Kreutzer, S., Mittelstrass, D., 2022. *RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE)*. R package version 0.1.0.9000-132.

**Author(s)**

Sebastian Kreutzer, Institute of Geography, Heidelberg University (Germany) , 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,
    sample = 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

<code>x</code>	( <b>required</b> ): input object of class <code>RLumSTARR_RFCurveArray</code> or <code>RLumSTARR_TRUELight</code>
<code>...</code>	further parameters, currently not used

### Value

[data.frame](#)

### Function version

0.1.0

### How to cite

Kreutzer, S., 2022. `melt_RLumSTARR()`: Melt output from RLumSTARR into simple data frames. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2022. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-132.

### 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	<b>character (required)</b> : path to files to be analysed
RF_nat	<b>character (default)</b> : name of the RF_nat file
RF_reg	<b>character (default)</b> : name of the RF_reg file
bg_rm	<b>character (with default)</b> : 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	<b>numeric (with default)</b> : grouping value for running median to remove outliers
image_alignment	<b>logical (with default)</b> : enable/disable image alignment
first_slices_rm	<b>logical (with default)</b> : remove first slice of each curve set
noise_tolorance	<b>numeric (with default)</b> : noise tolerance parameter
ROI_size	<b>numeric (with default)</b> : ROI size in pixel



center_x	<b>numeric</b> ( <i>with default</i> ): aliquot ROI centre x-coordinate
center_y	<b>numeric</b> ( <i>with default</i> ): aliquot ROI centre y-coordinate
diameter	<b>numeric</b> ( <i>with default</i> ): relative diameter aliquot ROI
use_predefined_ROIs	<b>logical</b> ( <i>with default</i> ): use pre-defined ROIs imported from a file ROIs.zip found in the same folder as the files
channel_time	<b>numeric</b> ( <i>with default</i> ): channel time, this parameter was set the moment the sequence was written
save_workflow_images	<b>logical</b> ( <i>with default</i> ): enable/disable writing of additional workflow images
save_additional_results	<b>logical</b> ( <i>with default</i> ): enable/disable writing of additional workflow images
save_signal_decay_videos	<b>logical</b> ( <i>with default</i> ): enable/disable writing of additional workflow videos
offset_time	<b>numeric</b> ( <i>with default</i> ): offset time for the time axis
.ImageJ	<b>numeric</b> ( <i>with default</i> ): 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., 2022. run\_ImageJ(): Run ImageJ SR-RF macro. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2022. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-132.

### Author(s)

Sebastian Kreutzer, Institute of Geography, Heidelberg University (Germany) , 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](#) 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	<a href="#">array</a> ( <b>required</b> ): object created by <a href="#">create_RFCurveArray</a>
ROI	<a href="#">numeric</a> ( <i>optional</i> ): ROIs to be analysed, if nothing is given all ROIs are analysed, however, the first ROIS is discarded!
stepping	<a href="#">numeric</a> ( <i>with default</i> ): the stepping parameter from and to be passed to <a href="#">extract_TRUELight</a>
mc.cores	<a href="#">numeric</a> ( <i>with default</i> ): number of cores used for the processing, passed to <a href="#">parallel::mclapply</a>
method_control	<a href="#">list</a> ( <i>optional</i> ): parameters to be passed to <a href="#">extract_TRUELight</a>
include_jags_output	<a href="#">logical</a> ( <i>with default</i> ): allows to include the output from JAGS as info object in the <a href="#">Luminescence::RLum.Analysis</a> 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 <a href="#">extract_TRUELight</a> is recommended instead.
verbose	<a href="#">logical</a> ( <i>with default</i> ): enable/disable verbose mode. The output of the MCMC sampling using <code>rjags</code> is <b>always</b> silent.

## Value

The output is a [list](#) containing [Luminescence::RLum.Analysis](#) objects with two [Luminescence::RLum.Data.Curve](#) objects for RF\_nat and RF\_reg respectively.

## Function version

0.1.0

**How to cite**

Kreutzer, S., 2022. run\_TRUELightExtraction(): Run Automated TRUE-Light Extraction over multiple datasets. Function version 0.1.0. In: Kreutzer, S., Mittelstrass, D., 2022. RLumSTARR: Spatially Resolved Radiofluorescence Analysis (EXPERIMENTAL PACKAGE). R package version 0.1.0.9000-132.

**Author(s)**

Sebastian Kreutzer, Institute of Geography, Heidelberg University (Germany) , 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,
    sample = 0,
    thin = 20))
```

# Index

- \* **datagen**
  - create\_RFCurveArray, 2
  - extract\_TRUELight, 3
- \* **package**
  - RLumSTARR-package, 2
- array, 4, 10
- character, 4, 6, 8
- coda::HPDinterval, 6
- coda::mcmc, 6
- coda::mcmc.list, 6
- create\_RFCurveArray, 2, 4, 5, 10
- data.frame, 7
- extract\_TRUELight, 3, 5, 6, 10
- get\_MCMCParameters, 5, 5
- list, 2, 4, 6, 10
- logical, 4, 6, 8–10
- Luminescence::extract\_ROI, 2, 3
- Luminescence::read\_RF2R, 3
- Luminescence::RLum.Analysis, 10
- Luminescence::RLum.Data.Curve, 4, 10
- matrix, 6
- melt\_RLumSTARR, 7
- numeric, 4, 6, 8–10
- parallel::mclapply, 10
- rjags::coda.samples, 4
- RLumSTARR (RLumSTARR-package), 2
- RLumSTARR-package, 2
- run\_ImageJ, 8
- run\_TRUELightExtraction, 5, 10
- runjags::run.jags, 4
- writeLines, 4