Package: xlum (via r-universe)

November 16, 2026

Type Package

Title Read, Write, and Convert XLUM Data

Version 0.2.0.9000

Date 2026-10-20

Maintainer Sebastian Kreutzer

<maintainer_luminescence@r-luminescence.org>

Description Support for the XLUM file format (Kreutzer et al. (2023) <doi:10.5194/gchron-5-271-2023>) for exchange and long-term preservation of luminescence data. The package facilitates the import and export of XLUM files and the conversion from other ASCII and binary formats commonly used to store luminescence (dating) data.

License GPL-3

URL https://r-lum.github.io/xlum/

BugReports https://github.com/R-Lum/xlum/issues

Depends R (>= 4.5)

Imports base64enc (>= 0.1), Luminescence (>= 1.3.1.9000-4), methods, utils, xml2 (>= 1.6.0)

Suggests spelling (>= 2.3.2), testthat (>= 3.3.2)

Remotes R-Lum/Luminescence

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

Config/pak/sysreqs cmake make libxml2-dev libssl-dev

Repository https://r-lum.r-universe.dev

Date/Publication 2026-11-16 15:14:11 UTC

RemoteUrl https://github.com/r-lum/xlum

RemoteRef HEAD

RemoteSha 9bd727edc5319226c8974d8802446b59ee2d17c7

Contents

xlum-package 2
convert_2xlum 3
convert_rlum2xlum 5
read_xlum 6
validate_xlum 7
write_RLum2XLUM,RLum-method 8
write_xlum 9

Index **11**

`xlum-package` *Read, write, and convert XLUM data*

Description

Support for the XLUM file format for exchange and long-term preservation of luminescence data. The package facilitates the import and export of XLUM files and the conversion from other ASCII and binary formats commonly used to store luminescence (dating) data.

Details

Support contact

- https://github.com/R-Lum/xlum/discussions

Bug reporting

- https://github.com/R-Lum/xlum/issues

Project source code repository

- https://github.com/R-Lum/xlum

Package maintainer

Sebastian Kreutzer, F2.1 Geophysical Parametrisation/Regionalisation, LIAG - Institute for Applied Geophysics (Germany)

Funding

- The initial development of xlum received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 844457 (project: CREDit)

- Since 03/2023: Sebastian Kreutzer as maintainer of the package receives funding from the DFG Heisenberg programme No 505822867
- The contribution by Marco Colombo was enabled through the DFG programme "REPLAY: REProducible Luminescence Data AnalYses" No 528704761

Author(s)

Maintainer: Sebastian Kreutzer <maintainer_luminescence@r-luminescence.org> (ORCID)

Authors:

- Sebastian Kreutzer <maintainer_luminescence@r-luminescence.org> (ORCID)
- Marco Colombo (ORCID)

Other contributors:

- Christoph Burow (ORCID) [contributor]

References

Kreutzer, S., Grehl, S., Höhne, M., Simmank, O., Dornich, K., Adamiec, G., Burow, C., Roberts, H. M., Duller, G. A. T., 2023. XLUM: an open data format for exchange and long-term preservation of luminescence data. Geochronology, 5, 271-284, doi:10.5194/gchron52712023

See Also

Useful links:

- https://r-lum.github.io/xlum/
- Report bugs at https://github.com/R-Lum/xlum/issues

`convert_2xlum` *Convert input files to XLUM*

Description

These functions convert several types of input files to XLUM format. They are implemented as shallow wrappers around an appropriate `Luminescence` read function and convert_rlum2xlum.

Usage

```
convert_binx2xlum(file, out_file = NULL, ...)

convert_bin2xlum(file, out_file = NULL, ...)

convert_xsyg2xlum(file, out_file = NULL)

convert_Daybreak2xlum(file, out_file = NULL)

convert_psl2xlum(file, out_file = NULL)
```

Arguments

`file`	character (**required**): input file.
`out_file`	character (*optional*): path to an output file where the XLUM file should be written to. If `NULL` (default), no file is written, but an xml2::xml_document object is returned.
`...`	support for a limited number of additional arguments (currently only `position` and `n.records`) to be forwarded to Luminescence::read_BIN2R.

Value

When `out_file` is `NULL` (default), an xml2::xml_document object is returned, otherwise the function attempts to write an `.xlum` file with the path provided.

Functions

- `convert_binx2xlum()`: Convert Risø BIN/BINX files to XLUM format. This is a shallow wrapper around Luminescence::read_BIN2R and convert_rlum2xlum.
- `convert_xsyg2xlum()`: Convert Freiberg Instruments XSYG files to `xlum` format. This is a shallow wrapper around Luminescence::read_XSYG2R and convert_rlum2xlum.
- `convert_Daybreak2xlum()`: Convert Daybreak TXT and DAT files to XLUM format. This is a shallow wrapper around Luminescence::read_Daybreak2R and convert_rlum2xlum.
- `convert_psl2xlum()`: Convert SUERC PSL files to XLUM format. This is a shallow wrapper around Luminescence::read_PSL2R and convert_rlum2xlum.

Function version

0.1.0

Author(s)

Sebastian Kreutzer, F2.1 Geophysical Parametrisation/Regionalisation, LIAG - Institute for Applied Geophysics (Germany)

See Also

convert_rlum2xlum, Luminescence::read_BIN2R, Luminescence::read_Daybreak2R, Luminescence::read_PSL2R, Luminescence::read_XSYG2R

Examples

```
## load BINX example data
file <- system.file("extdata/BINfile_V8.binx", package = "Luminescence")
convert_binx2xlum(file)

## load XSYG example data
file <- system.file("extdata/XSYG_file.xsyg", package = "Luminescence")
convert_xsyg2xlum(file)
```

```
## load Daybreak example data
file <- system.file("extdata/Daybreak_TestFile.txt", package = "Luminescence")
convert_Daybreak2xlum(file)

## load PSL example data
file <- system.file("extdata", "DorNie_0016.psl", package = "Luminescence")
convert_psl2xlum(file)
```

`convert_rlum2xlum` *Convert RLum-class objects to the XLUM format*

Description

Converts Luminescence::RLum objects to XLUM format.

Usage

```
convert_rlum2xlum(rlum, out_file = NULL)
```

Arguments

`rlum`	Luminescence::RLum or a list of it (**required**): input object.
`out_file`	character (*optional*): path to an output file where the XLUM file should be written to. If `NULL` (default), no file is written, but an xml2::xml_document object is returned.

Details

The function tries to make the best possible conversion of an Luminescence::RLum object to the XLUM format. Supported are Luminescence::RLum.Analysis objects as well as the Luminescence::RLum.Data objects they may contain. Luminescence::RLum.Results objects are not supported.

The input may be a single object or a list of objects. All objects in a list must be of the same class, either Luminescence::RLum.Analysis or Luminescence::RLum.Data. A list of data objects is combined into a single sequence, whereas a list of Luminescence::RLum.Analysis objects produces one `<sequence/>` per element.

Each analysis record is mapped to a `<record/>` node holding one or more `<curve/>` nodes. The measured values are stored as the text content of the `<curve/>` node, while `tValues`, `xValues`, and `yValues` are kept as node attributes. Additional record attributes, such as `curveType` and `startDate`, are transferred where available.

As Luminescence::RLum objects do not distinguish between different samples, the `<xlum/>` and `<sample/>` nodes are created automatically, so all converted objects end up in a single sample.

Value

When `out_file` is `NULL` (default), an xml2::xml_document object is returned. Otherwise the document is written to `out_file` (with the `.xlum` extension added automatically if needed) and `NULL` is returned invisibly.

Function version

0.1.0

Author(s)

Sebastian Kreutzer, F2.1 Geophysical Parametrisation/Regionalisation, LIAG - Institute for Applied Geophysics (Germany)

See Also

convert_2xlum, write_xlum, read_xlum, Luminescence::RLum.Analysis

Examples

```
## load example data
data(ExampleData.RLum.Analysis, envir = environment(), package = "Luminescence")

## convert to an in-memory XLUM document
xlum <- convert_rlum2xlum(IRSAR.RF.Data)

## convert and write to a file
convert_rlum2xlum(IRSAR.RF.Data, tempfile(fileext = ".xlum"))
```

`read_xlum` *Import XLUM files*

Description

Imports XLUM files using xml2::read_xml. Comes with a few convenience features, such as the transformation of string lists to numeric vectors where meaningful.

Usage

```
read_xlum(file, verify = TRUE, output = "xlum_list")
```

Arguments

`file`	character (**required**): path to file.
`verify`	logical (*with default*): enable/disable format validation.
`output`	character (*with default*): output object of the import, one of `"xlum_list"` (default), `"list"`, or `"xml_document"` (see Details).

Details

Supported output options

- `"xlum_list"` (default): imports XML data and coerces it to a list. Numeric values for level `<curve/>` are converted into numeric vectors for better handling.
- `"list"`: imports XML data and coerces it to a list, no further treatment of data.
- `"xml_document"`: unprocessed output from xml2::read_xml.

Value

The output depends on the setting selected in `output`. It will be either an xml2::xml_document, a plain list, or a list of class `xlum_list`.

Function version

0.1.0

Author(s)

Sebastian Kreutzer, F2.1 Geophysical Parametrisation/Regionalisation, LIAG - Institute for Applied Geophysics (Germany)

See Also

xml2::read_xml, validate_xlum, write_xlum

Examples

```
file <- system.file("extdata/xlum_example.xlum", package = "xlum")
read_xlum(file)
```

---

validate_xlum *Validate XLUM format against an XSD reference schema*

Description

A convenience wrapper around xml2::xml_validate to check an XLUM file against the reference format description shipped with xlum-package.

Usage

```
validate_xlum(file, xsd_version = NULL)
```

Arguments

`file`	character or xml2::xml_document (**required**): path to an XLUM file, or an in-memory xml2::read_xml document to test.
`xsd_version`	numeric (*optional*): format version to be used for validation, e.g., 1. If `NULL` (default), the most recent version will be used.

Value

A logical value, with the validation errors in its attributes.

Function version

0.1.0

Author(s)

Sebastian Kreutzer, F2.1 Geophysical Parametrisation/Regionalisation, LIAG - Institute for Applied Geophysics (Germany)

See Also

xml2::xml_validate, read_xlum, write_xlum

Examples

```
file <- system.file("extdata/xlum_prototype.xlum", package = "xlum")
validate_xlum(file)
```

`write_RLum2XLUM,RLum-method`

Write an `RLum` *object into XLUM format*

Description

The function writes an Luminescence::RLum object into XLUM format. The function is a simple wrapper around convert_rlum2xlum and write_xlum.

Usage

```
## S4 method for signature 'RLum'
write_RLum2XLUM(object, file, encode_base64 = FALSE, ...)

## S4 method for signature 'list'
write_RLum2XLUM(object, file, encode_base64 = FALSE, split_samples = NULL, ...)
```

Arguments

`object`	Luminescence::RLum or list (**required**): S4 object of class `RLum` or a list of such objects.
`file`	character (**required**): path to the output file.
`encode_base64`	logical (*with default*): enable/disable base64 encoding for values in the `<curve/>` node (`FALSE` by default).
`...`	currently ignored.
`split_samples`	list (**optional**): list of index vectors defining how the `object` list should be split into different `<sample>` nodes. If `NULL` (default), only one `<sample>` node is generated.

Value

The document is written to `file`, with the `.xlum` extension added automatically if needed; `NULL` is returned invisibly.

Function version

0.1.0

Author(s)

Marco Colombo, Institute of Geography, Heidelberg University (Germany)

See Also

convert_rlum2xlum, write_xlum

Examples

```
data(ExampleData.RLum.Analysis, envir = environment(), package = "Luminescence")
out_file <- tempfile()
write_RLum2XLUM(IRSAR.RF.Data, out_file)
```

`write_xlum` *Write XLUM files*

Description

Create XLUM files using xml2::write_xml.

Usage

```
write_xlum(x, file, encode_base64 = FALSE, ...)
```

Arguments

x	xlum_list, list, xml2::xml_document (**required**): input object.
file	character (**required**): path to the output file, with the .xlum extension added automatically if needed.
encode_base64	logical (*with default*): enable/disable base64 encoding for values in the <curve/> node. Please note that this parameter works only if the input is of class xlum_list.
...	further arguments to be passed to xml2::write_xml.

Value

The document is written to file; NULL is returned invisibly.

Function version

0.1.0

Author(s)

Sebastian Kreutzer, F2.1 Geophysical Parametrisation/Regionalisation, LIAG - Institute for Applied Geophysics (Germany)

See Also

xml2::write_xml, read_xlum

Examples

```
file <- system.file("extdata/xlum_prototype.xlum", package = "xlum")
xlum_data <- read_xlum(file)
write_xlum(xlum_data, tempfile())
```

Index

∗ **IO**
    read_xlum, 6
    write_RLum2XLUM,RLum-method, 8
    write_xlum, 9
∗ **datagen**
    convert_rlum2xlum, 5
∗ **package**
    xlum-package, 2

character, *4–6*, *8–10*
convert_2xlum, 3, *6*
convert_bin2xlum *(*convert_2xlum*)*, 3
convert_binx2xlum *(*convert_2xlum*)*, 3
convert_Daybreak2xlum *(*convert_2xlum*)*, 3
convert_psl2xlum *(*convert_2xlum*)*, 3
convert_rlum2xlum, *3, 4*, 5, *8, 9*
convert_xsyg2xlum *(*convert_2xlum*)*, 3

list, *5*, *7*, *9, 10*
logical, *6*, *8–10*
Luminescence::read_BIN2R, *4*
Luminescence::read_Daybreak2R, *4*
Luminescence::read_PSL2R, *4*
Luminescence::read_XSYG2R, *4*
Luminescence::RLum, *5*, *8, 9*
Luminescence::RLum.Analysis, *5, 6*
Luminescence::RLum.Data, *5*
Luminescence::RLum.Results, *5*

numeric, *7, 8*

read_xlum, 6, *6*, *8*, *10*

validate_xlum, 7, *7*

write_RLum2XLUM,list-method
        *(*write_RLum2XLUM,RLum-method*)*, 8
write_RLum2XLUM,RLum-method, 8
write_xlum, *6–8*, 9, *9*
xlum *(*xlum-package*)*, 2
xlum-package, 2, *7*
xml2::read_xml, *6–8*
xml2::write_xml, *9, 10*
xml2::xml_document, *4–8*, *10*
xml2::xml_validate, *7, 8*