

reml

An R package for reading, writing, integrating and publishing data using the Ecological Metadata Language (EML) format.

- **Note:** This package is still in active development and not yet submitted to CRAN. Functions and documentation may be incomplete and subject to change.
- Maintainer: Carl Boettiger
- Authors: Carl Boettiger, Duncan Temple Lang, Karthik Ram, Matt Jones
- License: [CC0](#)
- [Issues](#): Bug reports, feature requests, and development discussion.

This is a dynamic document, generated by [knitr](#) using [vignette.Rmd](#) as the source.

Quick Start

Install the R package:

```
library("devtools")
install_github("reml", "ropensci")
```

Load the package:

```
library("reml")
```

Writing R data into EML

`reml` now extends R's `data.frame` class by introducing the `data.set` class which includes additional metadata required by EML. A `data.set` can be created much like a `data.frame` by specifying additional arguments

```
dat = data.set(river = c("SAC", "SAC", "AM"),
               spp   = c("king", "king", "ccho"),
               stg   = c("smolt", "parr", "smolt"),
               ct     = c(293L, 410L, 210L),
               col.defs = c("River site used for collection",
                           "Species common name",
                           "Life Stage",
                           "count of live fish in traps"),
```

```
unit.defs = list(c(SAC = "The Sacramento River",
  AM = "The American River"),
  c(king = "King Salmon",
    ccho = "Coho Salmon"),
  c(parr = "third life stage",
    smolt = "fourth life stage"),
  "number"))
```

- **col.defs:** These are usually just plain text definitions, though a URI to a semantic definition can be particularly powerful. See “Advanced Use” for details on adding richer information, such as the method used to collect the data or set the geographic, taxonomic, or temporal coverage of an individual column.
- **unit.defs:** For factors, this is a definition of the levels involved. For numeric data, specify the units from [this list](#). For dates, specify the format, (e.g. YYYY or MM-DD-YY). For character strings, a definition of the kind of string can be given, (e.g. species scientific name), otherwise the column description will be used.

Alternatively, annotations can be added to an existing data frame, `data.set(my.data.frame, col.defs = ..., unit.defs = ...)`.

```
eml_write(dat,
  title = "reml example",
  description = "An example, intended for
    illustrative purposes only.",
  creator = "Carl Boettiger <cboettig@gmail.com>",
  file = "reml_example.xml")
```

```
[1] "reml_example.xml"
```

for convenience, *dat* could simply be a *data.frame* and *reml* will launch its metadata wizard to assist in constructing the metadata based on the *data.frame* provided. While this may be helpful starting out, regular users will find it faster to define the columns and units directly in the format above.

See the [EML generated](#) by this example. Note that if the `metadata` argument providing the definitions for the `data.frame` is not specified, *reml* will launch the wizard to prompt the user.

Configuration

Configure general metadata you may want to frequently reuse, avoiding having to specify things like the name of the creator or contact.

```
eml_config(creator = list("Carl Boettiger <cboettig@ropensci.org>", "Karthik Ram"), contact =
```

- See [Open issues in writing EML](#)

Publish EML

Once we have generated a basic EML file, we can publish it to a variety of academic repositories where it will be indexed, permanently archived, more easily citable, and publicly discoverable.

In this example, we will publish the above EML to [figshare](#) using the [rfigshare](#) package. Like most academic repositories, figshare requires some basic metadata for any entry. Much of this we can automatically extract from our existing EML file. Additional metadata unique to figshare (e.g. matching it's allowed "categories" can also be provided.

*Note: this example assumes **rfigshare** is installed, even though it is not required to use **reml** and run the examples in the other sections*

```
id = eml_publish("reml_example.xml", description="Example EML file from reml", categories =
```

Error: Could not resolve host: api.figshare.com

This creates a draft file visible only to the user configured in **rfigshare**. The document can be made (permanently) public using either the figshare web interface, the **rfigshare** package (using `fs_make_public(id)`) or just by adding the argument `visibility = TRUE` to the above `eml_publish` command.

<- Too detailed for this vignette

For the sake of completeness, the explicitly figshare metadata is also written into a dedicated `<additionalMetadata>` node with the attribute `id = figshare`. This allows us to distinguish between metadata that is an is not available to figshare's indexing, e.g. if we use the figshare search tool.

->

In development See [Open issues in publishing EML](#)

Reading EML

```
obj <- eml_read("reml_example.xml")
```

We can use various accessor functions to return the data and metadata elements in native R formats.

```
dat <- dataTable(obj)

metadata <- attributeList(obj)

contact(obj)

[1] "Carl Boettiger <cboettig@gmail.com>"
```

Note that the contact has been coerced into R's built-in 'person' object:

```
class(contact(obj))

[1] "person"

citationInfo(obj)
```

```
Boettiger C (2013-10-31). _reml example_.
```

In development See [Open issues in reading EML](#)

Integrating multiple EML files into a single data frame

In development See [Open issues in integrating EML](#)

Testing and development

While the dependencies for basic functionality are kept to a minimum, to access all the functions and tests implemented in **reml** you'll need several additional packages.

```
install.packages(c("yaml", "knitr", "rfigshare", "testthat", "RCurl"))
```

Some of these additional packages are not yet on CRAN and may not be stable.

```
install.packages("uuid")
install_github("RHTMLForms", "omegahat")
install_github("XMLSchema", "omegahat")
```

Successful installation of these packages will require the **devtools** package (for **install_github**) and the ability to build packages from source.