createMetadata

cgries

July 4, 2016

This will create a basic EML file using the EML R package from gitHub https://github.com/ropensci/EML.

Disclaimer: Written on July 4, 2016 while the EML R package is not published and constantly being worked on. This code may not work after more work is done on the package.

Three word files need to be created, an Abstract for the data, the Methods description and Intellectual Rights.The word files will be read in and transformed into doctype XML.

```
library(EML)
library(rmarkdown)
```

`title`: be descriptive, a good title contains the what, where and when of the dataset

```
title <- "Sparkling Lake, Wisconsin, daily metabolism calculations 2014"
pubDate <- "2016"
```

`abstract`: general information about the data and caveats on what they can be used for.

```
abstract <- as(set_TextType("SparklingAbstract_NEP.docx"), "abstract")
```

Personnel involved in the study: `creator`, `associatedParty` (with roles like field tech, lab tec, etc.) `contact` information for questions regarding the data. Preferably, this is a generic address that will be valid for a long time. E.g., a generic e-mail address that forwards to different people.

```
#creator
hilary <- as.person("Hilary Dugan <hilarydugan@gmail.com>")

creator <- as(hilary, "creator")
```

```
## Warning: Person Hilary Dugan <hilarydugan@gmail.com> was not given any
## role.
```

```
#publisher
NTL_address <- new("address",
                   deliveryPoint = "Center for Limnology 680 North Park Str.",
                   city = "Madison",
                   administrativeArea = "WI",
                   postalCode = "53706",
                   country = "USA")

publisher <- new("publisher",
                 organizationName = "North Temperate Lakes LTER",
                 address = NTL_address)

#contact
NTL_contact <- new("individualName",
                   givenName = "NTL",
                   surName = "Information-Manager")

contact <- new("contact",
      individualName = NTL_contact,
      electronicMail = "infomgr@lter.limnology.wisc.edu",
      address = NTL_address,
      organizationName = "North Temperate Lakes LTER",
      phone = "608-890-3446")
```

`keywords`: should include higher level concepts that people my be searching for. Use words that don't already apear in the title or abstract.

```
keywordSet <- c(new("keywordSet",
                    keyword = c("gas flux",
                    "metabolism",
                    "lake")))
```

`intellectualRights`: use a license that regulates re-use from creative commons and a disclaimer for limiting your responsibility for any ramifications of possible errors in the data.

```
intellectualRights <- as(set_TextType("intellectualRights.docx"), "intellectualRights")
```

`methods`: Detailed description of how the date were generated. This may include an ID for input data, a citation for the script used to manipulate them.

```
methods<- set_methods("SparklingMethods_NEP.docx")
```

`coverage`: Begin and End dates of the data. Geographic location of where the data have been sampled.

```
begindate <- "2014-06-01"
enddate <- "2014-08-31"

geographicDescription <- "Sparkling Lake, Vilas County, Wisconsin USA"

coverage <- set_coverage(begin = begindate, end = enddate,
                         #sci_names = "Sarracenia purpurea",
                         geographicDescription = geographicDescription,
                         west = -89.704160, east = -89.695191,
                         north = 46.015468, south = 46.002189,
                         altitudeMin = 500, altitudeMaximum = 500,
                         altitudeUnits = "meter")
```

`attributeList`: describe each column in the data table. First create a template csv file for each piece of information needed in EML

```
#read the meta data table
df <- read.csv("Sparkling2014_NEP.csv", header=TRUE, sep=",", quote("\""))
#look at the table structure
str(df)
```

```
## 'data.frame':    92 obs. of  3 variables:
##  $ datetime    : Factor w/ 92 levels "2014-06-01","2014-06-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ NEP.mle.cole: num  0.154 0.01 0.319 0.037 0.084 -0.037 0.285 0.19 0.177 0.175 ...
##  $ NEP.mle.read: num  0.33 0.093 0.398 0.099 0.126 -0.008 0.352 0.245 0.2 0.215 ...
```

```
#set up the template for attribute metadata
rows <- ncol(df)
attributes <- data.frame(attributeName = character(rows),
                         formatString = character(rows),
                         unit = character(rows),
                         numberType = character(rows),
                         definition = character(rows),
                         attributeDefinition = character(rows),
                         columnClasses = character(rows),
                         minimum = character(rows),
                         maximum = character(rows),
                         stringsAsFactors = FALSE)

#get some metadata from data frame
#add the column names to the template file
attributes$attributeName <- names(df)

#get the data types for each column
attributes$columnClasses <- sapply(df, class)

#set what R thinks is integer to numeric
attributes$columnClasses[attributes$columnClasses == "integer"] <- "numeric"

#write the prepared template to a csv file
write.csv(attributes, file = "Sparkling2014Metadata_NEP.csv", row.names = FALSE)
```

Open the metadata csv file in excel and enter information

`attributeName` column should be filled out already

`formatString` is needed for dates only. Preferrably the date is formatted as yyyy-mm-dd hh:mm:ss. But any format can be described. If the data file is opened in Excel the date format usually gets altered, only open in a simple text editor.

`unit` is needed for every numeric column. Look at the standard units already defined in EML and use one of them if possible. Otherwise define your own custom unit (see below).

```
standardUnits <- get_unitList()
View(standardUnits$units)
```

`numberType` can be whole, integer or real. `definition` is only needed for text type columns and can be the same as the `attributeDescription` `attributeDescription`: a human readable explanation of the what is in this column `columnClass` describes the data type of the column (text, Date, numeric) may need to be changed if it is Date instead of text/character/factor this depends on how R reads the given data table. `minimum` and `maximum` are prescriptive, i.e., consider this a range check for your data rather than using the actual minimum and maximum in this particular dataset

```
#read metadata back from csv
attributes <- read.csv("Sparkling2014Metadata_NEPCopy.csv", header = TRUE, sep = ",", quote = "\"", as.i
```

```r
# get the col_classes back out of the data frame
#this is needed to use the function in the EML package
col_classes <- attributes[,"columnClasses"]
attributes$columnClasses <- NULL

#define the custom unit
unitType <- data.frame(id = c("massDensityFlux", "massDensityFlux"), dimension = c("massDensity", "time
custom_units <- data.frame(id = "milligramsPerLiterPerDay", unitType = "massDensityFlux", parentSI = "g
unitsList <- set_unitList(custom_units, unitType)

#turn the attribute descriptions into XML
attributeList <- set_attributes(attributes, col_classes = col_classes)
```

```
## Warning in set_attribute(attributes[i, ], factors = factors): unit 'milligramsPerLiterPerDay' is not
##                         Please define a custom unit or replace with a
##                         recognized standard unit (see set_unitList() for details)

## Warning in set_attribute(attributes[i, ], factors = factors): unit 'milligramsPerLiterPerDay' is not
##                         Please define a custom unit or replace with a
##                         recognized standard unit (see set_unitList() for details)
```

`physical` describes the basics of the file, e.g., column separator, row end code etc. and is guessed for a csv file with windows hard returns for row delimiters.

```r
physical <- set_physical("Sparkling2014_NEP.csv")

#put the data table description together
dataTable <- new("dataTable",
                 entityName = "Sparkling2014_NEP.csv",
                 entityDescription = "Daily oxygen flux calculated in two different methdos",
                 physical = physical,
                 attributeList = attributeList)


#put the dataset together
dataset <- new("dataset",
              title = title,
              creator = creator,
              pubDate = pubDate,
              intellectualRights = intellectualRights,
              abstract = abstract,
              #associatedParty = associatedParty,
              keywordSet = keywordSet,
              coverage = coverage,
              contact = contact,
              methods = methods,
              dataTable = dataTable)

#wrap it all into the eml tag and add additional metadata
eml <- new("eml",
           packageId = "knb-lter-ntl.10101.1",
           system = "knb",
           dataset = dataset,
           additionalMetadata = as(unitsList, "additionalMetadata"))

eml_validate(eml)
```

```
## Warning in is.na(encoding): is.na() applied to non-(list or vector) of type
## 'NULL'
```

```
## [1] TRUE
```