Package 'rangeModelMetadata'

June 11, 2021

Type Package

Title Provides Templates for Metadata Files Associated with Species Range Models

Version 0.1.4

Maintainer Cory Merow <cory.merow@gmail.com>

Description Range Modeling Metadata Standards (RMMS) address three challenges: they (i) are designed for convenience to encourage use, (ii) accommodate a wide variety of applications, and (iii) are extensible to allow the community of range modelers to steer it as needed. RMMS are based on a data dictionary that specifies a hierarchical structure to catalog different aspects of the range modeling process. The dictionary balances a constrained, minimalist vocabulary to improve standardization with flexibility for users to provide their own values. Merow et al. (2019) <DOI:10.1111/geb.12993> describe the standards in more detail. Note that users who prefer to use the R package 'ecospat' can obtain it from <https://github.com/ecospat/ecospat>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.1

Suggests BIEN, biomod2, dismo, ecospat, googlesheets, knitr, rmarkdown

Imports dplyr, jsonlite, MASS, raster, rgbif, rgdal, rgeos, shiny, sp, spatstat, spocc, spThin, utils

VignetteBuilder knitr

NeedsCompilation no

Author Cory Merow [aut, cre], Brian Maitner [aut], Hannah Owens [aut], Jamie Kass [aut], Brian Enquist [aut], Rob Guralnik [aut], Damaris Zurrell [aut], Christian Koenig [aut]

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2021-06-11 08:40:02 UTC

R topics documented:

`cleanForCSV` *Helper function for non-string metadata in rmmToCSV*

Description

Cleans up metadata instances that get messy if one tries to write them directly to csv tables (i.e. extent objects, bibtex objects.)

Usage

```
cleanForCSV(x = NULL)
```

Arguments

`x`	An `rmm` entry that returned to the `rmmToCSV` function.

Details

This is a utility function for use by `rmmToCSV`.

Value

Reformatted element for use in rmmToCSV function.

Author(s)

Hannah Owens <hannah.owens@gmail.com>, Cory Merow <cory.merow@gmail.com>

See Also

Other csvConversion: csvToRMM(), rmmToCSV()

---

csvToRMM *Create rangeModelMetaData ('rmm') object from a .csv File*

Description

Takes user-input .csv file and converts it to a rangeModelMetaData ('rmm') object.

Usage

```
csvToRMM(csv, family = NULL)
```

Arguments

csv	A character file path to the csv file.
family	character string; specifies an application profile (use case) by specifiying the families of entitiies that should be included. Specifying NULL includes all entities. Use rmmFamilies() to see supported values.

Details

See Examples.

Value

An rmm object that was read from the supplied .csv text file.

Author(s)

Hannah Owens <hannah.owens@gmail.com>

See Also

Other csvConversion: cleanForCSV(), rmmToCSV()

Examples

```
csv <- "somePathOnYourMachine/rmm_example.csv";
## Not run: temp <- csvToRMM(csv);
```

`rmmAutofillBIEN` *Add occurrence metadata from a BIEN query to an rmm object*

Description

This function populates occurrence field in an rmm object with output from a BIEN_occurrence_... query

Usage

```
rmmAutofillBIEN(rmm, occurrences)
```

Arguments

`rmm`	an rmm list
`occurrences`	an occurrence data.frame obtained from a BIEN occurrence query

Details

See Examples.

Value

a range model metadata list

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

`BIEN_occurrence_species`

Other autofill: `rmmAutofillEnvironment()`, `rmmAutofillPackageCitation()`, `rmmAutofillspocc()`

Examples

```
## Not run:
rmm <- rmmTemplate()
xs <- BIEN::BIEN_occurrence_species(species="Xanthium strumarium")
rmmAutofillBIEN(rmm = rmm, occurrences = xs)

## End(Not run)
```

`rmmAutofillEnvironment` *Add relevant environmental data information to an rmm object*

Description

This can be used with environmental layers used for fitting or transferring

Usage

```
rmmAutofillEnvironment(rmm, env, transfer)
```

Arguments

`rmm`	an rmm list
`env`	a raster stack
`transfer`	0 if not transfer, 1:n for n environments that you're transferring to

Details

See Examples.

Value

a range model metadata list

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other autofill: `rmmAutofillBIEN()`, `rmmAutofillPackageCitation()`, `rmmAutofillspocc()`

Examples

```
## Not run:
rmm=rmmTemplate()
rasterFiles=list.files(path=paste(system.file(package='dismo'), '/ex', sep=''),
                       pattern='grd', full.names=TRUE)
#make a stack of the rasters
env=raster::stack(rasterFiles)
# for fitting environment
rmm=rmmAutofillEnvironment(rmm,env,transfer=0)
# for the first environment that you're transfering to
rmm=rmmAutofillEnvironment(rmm,env,transfer=1)
# for the second environment that you're transfering to, etc.
rmm=rmmAutofillEnvironment(rmm,env,transfer=2)
```

```
## End(Not run)
```

`rmmAutofillPackageCitation` *Add all package citations to an rmm object*

Description

Using bibtex citations

Usage

```
rmmAutofillPackageCitation(rmm, packages)
```

Arguments

`rmm`	an rmm list
`packages`	a vector of quoted package names

Details

See Examples.

Value

a range model metadata list

Author(s)

Brian Maitner <bmaitner@gmail.com>, Cory Merow <cory.merow@gmail.com>

See Also

Other autofill: `rmmAutofillBIEN()`, `rmmAutofillEnvironment()`, `rmmAutofillspocc()`

Examples

```
rmm=rmmTemplate()
rmm=rmmAutofillPackageCitation(rmm,c('raster','sp'))
```

`rmmAutofillspocc` *Add occurrence metadata from a spocc query to an rmm object*

Description

This function populates occurrence field in an rmm object with output from a spocc query

Usage

```
rmmAutofillspocc(rmm, occ)
```

Arguments

`rmm`	an rmm list
`occ`	Output from `occ`

Details

See Examples.

Value

a range model metadata list

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

`occ`

Other autofill: `rmmAutofillBIEN()`, `rmmAutofillEnvironment()`, `rmmAutofillPackageCitation()`

Examples

```
## Not run:
rmm=rmmTemplate()
xs <- spocc::occ("Xanthium strumarium")
rmmAutofillspocc(rmm = rmm, occ = xs)

## End(Not run)
```

`rmmCheckEmpty` *Check an rmm object for empty fields*

Description

Identify empty fields in an rmm object and classify these into obligate and optional fields.

Usage

```
rmmCheckEmpty(rmm, family = c("base"))
```

Arguments

`rmm`	a range model metadata list
`family`	an rmm family, "base" by default

Details

See Examples.

Value

A dataframe containing empty fields labelled as obligate, optional, or suggested.

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other check: `rmmCheckFinalize()`, `rmmCheckMissingNames()`, `rmmCheckName()`, `rmmCheckValue()`, `rmmCleanNULLs()`

Examples

```
#First, make an empty rmm object:
rmm<-rmmTemplate()
#Next, we check for emtpy fields:
empties1<-rmmCheckEmpty(rmm = rmm)
#If looks like there are quite a few empty obligate fields.  Let's populate a few:
rmm$data$occurrence$taxon<-"Acer rubrum"
rmm$data$environment$variableNames<-"Bio1"
#Now, if we run rmmCheckEmpty again, we see there are 2 fewer empty, obligate fields
empties2<-rmmCheckEmpty(rmm = rmm)
```

`rmmCheckFinalize` *Run a final check of an rmm object*

Description

Check an rmm object for non-standard and missing values and fields

Usage

```
rmmCheckFinalize(rmm, family = c("base"))
```

Arguments

`rmm`	a range model metadata list
`family`	The rmm family to check the rmm against

Details

See Examples.

Value

Prints feedback to point out possible errors.

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other check: `rmmCheckEmpty()`, `rmmCheckMissingNames()`, `rmmCheckName()`, `rmmCheckValue()`, `rmmCleanNULLs()`

Examples

```
rmm<-rmmTemplate() # Make an empty template
rmmCheckFinalize(rmm)
```

`rmmCheckMissingNames` *Check for missing fields*

Description

Identify obligate fields that are missing

Usage

```
rmmCheckMissingNames(rmm, family = c("base"))
```

Arguments

`rmm`	a range model metadata list
`family`	The rmm family to check the rmm against

Details

See Examples.

Value

A vector of names that are missing from the rmm object.

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other check: `rmmCheckEmpty()`, `rmmCheckFinalize()`, `rmmCheckName()`, `rmmCheckValue()`, `rmmCleanNULLs()`

Examples

```
rmm<-rmmTemplate() # Make an empty template
```

rmmCheckName *Check field names of a range model metadata list against conventions*

Description

Identify nonstandard fields

Usage

```
rmmCheckName(
  rmm,
  cutoff_distance = 3,
  returnData = F,
  interactiveCorrections = FALSE
)
```

Arguments

rmm	a range model metadata list
cutoff_distance	number of allowed different characters to match standardized names
returnData	logical. If FALSE, the functon will return the (possibly) corrected rmm object. If TRUE, the function will return a data.frame containing information on incorrect names.
interactiveCorrections	logical. If TRUE, the user will be prompted to indicate whether the proposed correction should be accepted, thereby modifying the 'rmm' object. If FALSE, suggestions will just be printed to the screen and users can edit them manually.

Details

See Examples.

Value

Either an rmm list object ('returnData=FALSE') or a data.frame containing information on possible name errors ('returnData=TRUE').

Note

Names returned by this check may be either incorrectly named or correctly named but missing from the data dictionary.

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other check: `rmmCheckEmpty()`, `rmmCheckFinalize()`, `rmmCheckMissingNames()`, `rmmCheckValue()`, `rmmCleanNULLs()`

Examples

```
rmm<-rmmTemplate() # Make an empty template
rmm$dataPrep$biological$taxonomicHarmonization$taxonomy_source<-"The Plant List"
# Add a new, non-standard field
rmm.1=rmmCheckName(rmm)
# Checking the names should identify the new, non-standard field we've added ("taxonomy_source")
```

`rmmCheckShiny` *RangeModelMetadata Check in Shiny*

Description

Run shiny app to visualize rmm check functions

Usage

```
rmmCheckShiny()
```

Details

See Examples.

Value

None

Note

This function launches a shiny app in the default web browser

Author(s)

Jamie Kass <jamie.m.kass@gmail.com>

Examples

```
## Not run:
rmm1=rmmTemplate()
rmm1=rmmAutofillPackageCitation(rmm1,c('raster','sp'))
rasterFiles=list.files(path=paste(system.file(package='dismo'), '/ex', sep=''),
                       pattern='grd', full.names=TRUE)
make a stack of the rasters
env=raster::stack(rasterFiles)
# for fitting environment
rmm1=rmmAutofillEnvironment(rmm1,env,transfer=0)
# for transfer environment 1 (assuming different than for fitting)
rmm1=rmmAutofillEnvironment(rmm1,env,transfer=1)
# for transfer environment 2 (assuming different than 1)
rmm1=rmmAutofillEnvironment(rmm1,env,transfer=2)

## End(Not run)
## Not run:  rmmCheckShiny(rmm1)
```

`rmmCheckValue` *Check values of a range model metadata list against commonly used values*

Description

Identify nonstandard values

Usage

```
rmmCheckValue(rmm, cutoff_distance = 3, returnData = F)
```

Arguments

`rmm`	a range model metadata list
`cutoff_distance`	The maximum allowable similarity (Levenshtein (edit) distance) for use in fuzzy matching.
`returnData`	Should a dataframe containing information on matched and unmatched values be returned? Default is FALSE

Details

See Examples.

Value

Text describing identical, similar and non-matched values for rmm entities with suggested values. If returnData = T, a dataframe is returned containing 5 columns: field (the rmm entity), exact_match (values that appear correct), partial_match (values that are partial_match to common values), not_matched(values that are dissimilar from accepted values), partial_match_suggestions (suggested values for partial_match values).

Note

Names returned by this check may be either incorrectly named or correctly named but missing from the data dictionary.

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other check: `rmmCheckEmpty()`, `rmmCheckFinalize()`, `rmmCheckMissingNames()`, `rmmCheckName()`, `rmmCleanNULLs()`

Examples

```
rmm<-rmmTemplate() #First, we create an empty rmm template
rmm$data$environment$variableNames<- c("bio1", "bio 2", "bio3", "cromulent")
#We add 3 of the bioclim layers, including a spelling error (an extra space) in bio2,
# and a word that is clearly not a climate layer, 'cromulent'.
rmmCheckValue(rmm = rmm)
#Now, when we check the values, we see that bio1 and bio2 are reported as exact matches,
#while 'bio 2' is flagged as a partial match with a suggested value of 'bio2',
# and cromulent is flagged as not matched at all.
#If we'd like to return a dataframe containing this information in a perhaps more useful format:
rmmCheckValue_output<-rmmCheckValue(rmm = rmm,returnData = TRUE)
```

---

rmmCleanNULLs *Remove NULL entries range model metadata list*

Description

Check if fields are NULL in a range model metadata list and toss

Usage

```
rmmCleanNULLs(rmm)
```

Arguments

rmm	a range model metadata list

Details

See Examples.

Value

printout to the console

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

See Also

Other check: `rmmCheckEmpty()`, `rmmCheckFinalize()`, `rmmCheckMissingNames()`, `rmmCheckName()`, `rmmCheckValue()`

Examples

```
# see vignette('rmm_vignette')
```

---

rmmDataDictionary *Open range model metadata dictionary.*

Description

For viewing only

Usage

```
rmmDataDictionary(excel = FALSE)
```

Arguments

excel	logical; open in excel?

Value

If 'excel==FALSE', returns a data.frame, if 'excel==TRUE' it returns nothing but attempts to open the metadata dictionary in excel.

Examples

```
dd=rmmDataDictionary()
```

rmmFamilies *Print supported family names for rmm objects*

Description

Used to see options to for specifying an rmm object template

Usage

```
rmmFamilies()
```

Value

a vector of characters, indicating which RMMS families are supported

Examples

```
rmmFamilies()
```

rmmSuggest *Suggest inputs for a range model metadata list*

Description

Supply fields to receive suggested inputs

Usage

```
rmmSuggest(charString, fullFieldDepth = FALSE)
```

Arguments

charString	string referencing fields of the form 'field1$field2' or 'field1$field2$field3', etc.
fullFieldDepth	print all fields below the current field depth
	rmm1=rmmTemplate() rmmSuggest('dataPrep',fullFieldDepth=FALSE) rmmSuggest('dataPrep',fullFieldDepth=TRUE) rmmSuggest('dataPrep$errors$duplicateRemoval') rmmSuggest('dataPrep$errors$duplicateRemoval$rule') rmmSuggest('model') rmmSuggest('modelFit$algorithmSettings$') rmmSuggest('modelFit$algorithmSettings$maxent$') rmmSuggest('$modelFit$algorithmSettings$maxent$featureSet')

Details

See Examples.

Value

list of suggestions

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>,

---

`rmmTemplate` *Range modeling metadata*

Description

Make an empty metadata list

Usage

```
rmmTemplate(family = NULL)
```

Arguments

`family`	character string; specifies an application profile (use case) by specifiying the families of entitiies that should be included. Specifying NULL includes all entities. Use 'rmmFamilies' to see supported values.

Details

See Examples.

Value

a range model metadata list

Author(s)

Cory Merow <cory.merow@gmail.com>, Brian Maitner <bmaitner@gmail.com>

Examples

```
rmm1=rmmTemplate()
rmm2=rmmTemplate(family=c('base'))
str(rmm2)
```

rmmToCSV *Create .csv File From rangeModelMetaData Object*

Description

Takes user-input rangeModelMetaData object and from it generates a .csv file that can be used to document range model metadata for a variety of applications.

Usage

```
rmmToCSV(x = rmmTemplate(family = NULL), filename = NULL)
```

Arguments

x	An object of class rmm that the user wishes transposed into a .csv file.
filename	The name of the transcription .csv file.

Details

See Examples.

Value

An data frame containing all the information from an rmm object.

Author(s)

Hannah Owens <hannah.owens@gmail.com>, Cory Merow <cory.merow@gmail.com>

See Also

Other csvConversion: cleanForCSV(), csvToRMM()

Examples

```
rmm=rmmTemplate()
rasterFiles=list.files(path=paste(system.file(package='dismo'), '/ex', sep=''),
                       pattern='grd', full.names=TRUE)
#make a stack of the rasters
env=raster::stack(rasterFiles)
# for fitting environment
rmm=rmmAutofillEnvironment(rmm,env,transfer=0)
# for the first environment that you're transfering to
rmm=rmmAutofillEnvironment(rmm,env,transfer=1)
# for the second environment that you're transfering to, etc.
rmm=rmmAutofillEnvironment(rmm,env,transfer=2)
## Not run:
tmp=rmmToCSV(rmm,file='somePathOnYourMachine/rmm_example.csv')

## End(Not run)
```

Index