Package: BioSIM (via r-universe)

October 11, 2024

Type Package

Title 'BioSIM' Client for 'BioSIM' Web API

Version 1.0.5

Description Provide a client that retrieves the climate variables from the original 'BioSIM' application hosted on a server. The 'BioSIM' application is being developed and maintained by the Canadian Forest Service.

URL https://github.com/RNCan/BioSimClient_R/wiki

Depends R (>= 3.5.0)

Imports J4R (>= 1.1.9)

Remotes CWFC-CCFB/J4R

License LGPL-3

BugReports https://github.com/RNCan/BioSimClient_R/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

SystemRequirements Java 8

Suggests testthat

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

RemoteUrl https://github.com/RNCan/BioSimClient_R

RemoteRef HEAD

RemoteSha 1e14e1c828337a70752ef560e2913ac627ccbf5f

Contents

`allMonths` *The list of all months*

Description

The list of all months

Usage

```
allMonths
```

Format

An object of class `character` of length 12.

`biosimclient.config` *Configure the client*

Description

The forceClimateGenerationEnabled argument forces BioSIM to generate climate for past dates instead of using the observations from the climate stations. By default this option is set to false. The nbNearestNeighbours argument sets the number of stations for the imputation of climate variables.

Usage

```
biosimclient.config(
  forceClimateGenerationEnabled = NULL,
  nbNearestNeighbours = NULL,
  isLocalConnectionEnabled = NULL,
  isTestModeEnabled = NULL
)
```

Arguments

`forceClimateGenerationEnabled`	a logical
`nbNearestNeighbours`	an integer
`isLocalConnectionEnabled`	a logical (only for test purposes)
`isTestModeEnabled`	a logical (only for test purpose)

Details

If an argument is set to null, there is no effect at all. If all the arguments are set to null, then the configuration is reset to its default value: the climate variables of past dates relies on observations and the number of climate stations is set to 4.

Examples

```
## Not run:
### enables the climate generation for past dates and uses 20 climate stations
biosimclient.config(T, 20)

### reset the configuration
biosimclient.config()
## End(Not run)
```

`biosimclient.getConfiguration`
Report of the climate generation settings

Description

The isForceClimateGenerationEnabled setting forces BioSIM to generate climate for past dates instead of using the observations from the climate stations. By default this option is set to false. The nbNearestNeighbours setting is the number of stations used to impute climate variables to a particular location.

Usage

```
biosimclient.getConfiguration()
```

Details

All the settings can be changed through the biosimclient.config function.

Value

a data.frame object

clearCache *Clear the cache of the client (DEPRECATED).*

Description

When using the weather generator, some objects are stored in memory on the server and a reference is stored in the client, so that subsequent calls on models for the same location and time interval does not have to generate the climate over and over again. After a while it may happen that a large number of objects are kept in memory. This method clears this cache on both the server and the client ends.

Usage

```
clearCache()
```

Examples

```
## Not run:
clearCache()
## End(Not run)
```

generateModelOutput *Generate climate and apply a model (DEPRECATED).*

Description

This function generated the basic climate variables for some locations and applies a particular model on this generated climate.

Usage

```
generateModelOutput(
  modelName,
  fromYr,
  toYr,
  id,
  latDeg,
  longDeg,
  elevM = rep(NA, length(longDeg)),
  isEphemeral = T,
  rep = 1,
  repModel = 1,
  rcp = "RCP45",
  climModel = "RCM4",
  additionalParms = NULL
)
```

Arguments

`modelName`	a character. Should be one of the models listed in the available models (see the getModelList() method)
`fromYr`	the starting date (yr) of the period (inclusive)
`toYr`	the ending date (yr) of the period (inclusive)
`id`	a vector with the ids of the plots
`latDeg`	the latitudes of the plots
`longDeg`	the longitudes of the plots
`elevM`	the elevations of the plots (can contain some NA or can be NULL, in which cases BioSim relies on a digital elevation model)
`isEphemeral`	a logical. If set to true, the generated climate is not stored on the server, which implies a greater computational burden and inconsistencies if different models are applied on the same locations. By default, it is set to true.
`rep`	number of replicates of generated climate (is set to 1 by default)
`repModel`	number of replicates on the model end (is set to 1 by default)
`rcp`	an representative concentration pathway (either "RCP45" or "RCP85")
`climModel`	a climatic model (either "RCM4", "GCM4" or "Hadley")
`additionalParms`	a named vector with the additional parameters if needed

Value

a data.frame object

Examples

```
locations <- BioSIM::twoLocationsInSouthernQuebec
addParms <- c("LowerThreshold"=5)
## Not run:
degreeDays <- generateModelOutput("DegreeDay_Annual", 2017, 2021, locations$Name, locations$Latitude,
                               locations$Longitude, locations$Elevation,
                             rcp = "RCP85", climModel = "GCM4", additionalParms = addParms)
## End(Not run)
```

`generateWeather` *Generate a meteorological time series and apply one or many models.*

Description

This function generated a meteorological time series for some locations and applies one or many models on this series.

Usage

```
generateWeather(
  modelNames,
  fromYr,
  toYr,
  id,
  latDeg,
  longDeg,
  elevM = rep(NA, length(longDeg)),
  rep = 1,
  repModel = 1,
  rcp = "RCP45",
  climModel = "RCM4",
  additionalParms = NULL
)
```

Arguments

`modelNames`	a character or a vector of character. Should be one or some models listed in the available models (see the getModelList() method)
`fromYr`	the starting date (yr) of the period (inclusive)
`toYr`	the ending date (yr) of the period (inclusive)
`id`	a vector with the ids of the plots
`latDeg`	the latitudes of the plots
`longDeg`	the longitudes of the plots
`elevM`	the elevations of the plots (can contain some NA or can be NULL, in which cases BioSim relies on a digital elevation model)
`rep`	number of replicates of generated climate (is set to 1 by default)
`repModel`	number of replicates on the model end (is set to 1 by default)
`rcp`	an representative concentration pathway (either "RCP45" or "RCP85")
`climModel`	a climatic model (either "RCM4", "GCM4" or "Hadley")
`additionalParms`	a list of named vectors with the additional parameters if needed

Value

a list of data.frame objects

Examples

```
locations <- BioSIM::twoLocationsInSouthernQuebec
addParms <- c("LowerThreshold"=5)
## Not run:
degreeDays <- generateWeather("DegreeDay_Annual", 2017, 2021, locations$Name, locations$Latitude,
                   locations$Longitude, locations$Elevation,
                   rcp = "RCP85", climModel = "GCM4",
                   additionalParms = list(addParms))
## End(Not run)
```

`getAnnualNormals` *Return the annual normals for a period*

Description

Return the annual normals for a period

Usage

```
getAnnualNormals(
  period,
  id,
  latDeg,
  longDeg,
  elevM = rep(NA, length(longDeg)),
  rcp = "RCP45",
  climModel = "RCM4"
)
```

Arguments

`period`	a string representing the period (either "1951_1980", "1961_1990", "1971_2000", "1981_2010" up to "2071_2100")
`id`	a vector with the ids of the plots
`latDeg`	the latitudes of the plots
`longDeg`	the longitudes of the plots
`elevM`	the elevations of the plots (can contain some NA, in which case BioSim relies on a digital elevation model)
`rcp`	an representative concentration pathway (either "RCP45" or "RCP85")
`climModel`	a climatic model (either "RCM4", "GCM4" or "Hadley")

Value

a data.frame object

Examples

```
locations <- BioSIM::twoLocationsInSouthernQuebec
## Not run:
annualNormals <- getAnnualNormals("1981_2010", locations$Name, locations$Latitude,
                                  locations$Longitude, locations$Elevation)
## End(Not run)
```

getModelDefaultParameters *Provide help for a particular model*

Description

Provide help for a particular model

Usage

```
getModelDefaultParameters(modelName)
```

Arguments

modelName	should be one of the character string returned by the getModelList function

Examples

```
## Not run:
getModelHelp("Spruce_Budworm_Biology")
## End(Not run)
```

getModelHelp *Provide help for a particular model*

Description

Provide help for a particular model

Usage

```
getModelHelp(modelName)
```

Arguments

modelName	should be one of the character string returned by the getModelList function

Examples

```
## Not run:
getModelHelp("Spruce_Budworm_Biology")
## End(Not run)
```

`getModelList` *Return the list of models available in BioSim*

Description

Provide the list of model that can be used in BioSIM after generating the climate for some locations.

Usage

```
getModelList()
```

Examples

```
## Not run:
getModelList()
## End(Not run)
```

`getModelOutput` *Generate climate and apply a model (DEPRECATED).*

Description

This function generated the basic climate variables for some locations and applies a particular model on this generated climate.

Usage

```
getModelOutput(
  fromYr,
  toYr,
  id,
  latDeg,
  longDeg,
  elevM = rep(NA, length(longDeg)),
  modelName,
  isEphemeral = T,
  rep = 1,
  repModel = 1,
  rcp = "RCP45",
  climModel = "RCM4",
  additionalParms = NULL
)
```

Arguments

`fromYr`	the starting date (yr) of the period (inclusive)
`toYr`	the ending date (yr) of the period (inclusive)
`id`	a vector with the ids of the plots
`latDeg`	the latitudes of the plots
`longDeg`	the longitudes of the plots
`elevM`	the elevations of the plots (can contain some NA, in which case BioSim relies on a digital elevation model)
`modelName`	a character. Should be one of the models listed in the available models (see the getModelList() method)
`isEphemeral`	a logical. If set to true, the generated climate is not stored on the server, which implies a greater computational burden and inconsistencies if different models are applied on the same locations. By default, it is set to true.
`rep`	number of replicates of generated climate (is set to 1 by default)
`repModel`	number of replicates on the model end (is set to 1 by default)
`rcp`	an representative concentration pathway (either "RCP45" or "RCP85")
`climModel`	a climatic model (either "RCM4", "GCM4" or "Hadley")
`additionalParms`	a named vector with the additional parameters if needed

Value

a data.frame object

Examples

```
locations <- BioSIM::twoLocationsInSouthernQuebec
addParms <- c("LowerThreshold"=5)
## Not run:
degreeDays <- getModelOutput(2017, 2021, locations$Name, locations$Latitude,
                             locations$Longitude, locations$Elevation, "DegreeDay_Annual",
                            rcp = "RCP85", climModel = "GCM4", additionalParms = addParms)
## End(Not run)
```

`getMonthlyNormals` *Return the monthly normals for a period*

Description

Return the monthly normals for a period

Usage

```
getMonthlyNormals(
  period,
  id,
  latDeg,
  longDeg,
  elevM = rep(NA, length(longDeg)),
  rcp = "RCP45",
  climModel = "RCM4"
)
```

Arguments

`period`	a string representing the period (either "1951_1980", "1961_1990", "1971_2000", "1981_2010" up to "2071_2100")
`id`	a vector with the ids of the plots
`latDeg`	the latitudes of the plots
`longDeg`	the longitudes of the plots
`elevM`	the elevations of the plots (can contain some NA, in which case BioSim relies on a digital elevation model)
`rcp`	an representative concentration pathway (either "RCP45" or "RCP85")
`climModel`	a climatic model (either "RCM4", "GCM4" or "Hadley")

Value

a data.frame object

Examples

```
locations <- BioSIM::twoLocationsInSouthernQuebec
## Not run:
monthlyMeans <- getMonthlyNormals("1981_2010", locations$Name, locations$Latitude,
                                  locations$Longitude, locations$Elevation)
## End(Not run)
```

`getNormals` *Return the normals for a period*

Description

If the argument averageOverTheseMonths is left NULL or empty, the monthly normals are provided. If this argument is filled with some months, then the normal are aggregated over these months.

Usage

```
getNormals(
  period,
  id,
  latDeg,
  longDeg,
  elevM = rep(NA, length(longDeg)),
  averageOverTheseMonths,
  rcp = "RCP45",
  climModel = "RCM4"
)
```

Arguments

`period`	a string representing the period (either "1951_1980", "1961_1990", "1971_2000", "1981_2010" up to "2071_2100")
`id`	a vector with the ids of the plots
`latDeg`	the latitudes of the plots
`longDeg`	the longitudes of the plots
`elevM`	the elevations of the plots (can contain some NA, in which case BioSim relies on a digital elevation model)
`averageOverTheseMonths`	a vector with some months if there is a need for aggregating the climate varibles
`rcp`	an representative concentration pathway (either "RCP45" or "RCP85")
`climModel`	a climatic model (either "RCM4", "GCM4" or "Hadley")

Value

a data.frame object

Examples

```
locations <- BioSIM::twoLocationsInSouthernQuebec
## Not run:
summerMean <- getNormals("1981_2010", locations$Name, locations$Latitude,
                         locations$Longitude, locations$Elevation,
                         c("June", "July", "August"))
## End(Not run)
```

`settingEnv` *The settings environment for this package*

Description

This environment contains the general settings of the package.

Usage

```
settingEnv
```

Format

An object of class `environment` of length 0.

`shutdownClient` *Shut down the Java server*

Description

This method overrides the original function of the J4R package. It only adds a call to the clearCache function before calling the original function of the J4R package.

Usage

```
shutdownClient()
```

Examples

```
## Not run:
shutdownClient()
## End(Not run)
```

`shutdownJava` *Shut down the Java server*

Description

This method overrides the original function of the J4R package. It only adds a call to the clearCache function before calling the original function of the J4R package.

Usage

```
shutdownJava()
```

Examples

```
## Not run:
shutdownJava()
## End(Not run)
```

`twoLocationsInSouthernQuebec` *A list of two plots located in southern Quebec*

Description

A list of two plots located in southern Quebec

Usage

```
data(twoLocationsInSouthernQuebec)
```

Format

An object of class `data.frame` with 2 rows and 5 columns.

Examples

```
data(twoLocationsInSouthernQuebec)
```

Index