Package 'RFmerge'

May 22, 2020

Type Package

Title Merging of Satellite Datasets with Ground Observations using Random Forests

Version 0.1-10

Author Mauricio Zambrano-Bigiarini [aut, cre, cph] (<https://orcid.org/0000-0002-9536-643X>), Oscar M. Baez-Villanueva [aut, cph], Juan Giraldo-Osorio [ctb]

Maintainer Mauricio Zambrano-Bigiarini <mzb.devel@gmail.com>

Description S3 implementation of the Random Forest MErging Procedure (RF-MEP), which combines two or more satellite-based datasets (e.g., precipitation products, topography) with ground observations to produce a new dataset with improved spatio-temporal distribution of the target field. In particular, this package was developed to merge different Satellite-based Rainfall Estimates (SREs) with measurements from rain gauges, in order to obtain a new precipitation dataset where the time series in the rain gauges are used to correct different types of errors present in the SREs. However, this package might be used to merge other hydrological/environmental satellite fields with point observations. For details, see Baez-Villanueva et al. (2020) <doi:10.1016/j.rse.2019.111606>. Bugs / comments / questions / collaboration of any kind are very welcomed.

License GPL (>= 3)

Depends R (>= 3.5.0)

Imports raster, sp, sf, randomForest, zoo, parallel, methods, stats, utils, pbapply

Suggests knitr, rmarkdown, rgdal

VignetteBuilder knitr

URL https://github.com/hzambran/RFmerge

MailingList https://stat.ethz.ch/mailman/listinfo/r-sig-ecology

BugReports https://github.com/hzambran/RFmerge/issues

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-22 08:00:02 UTC

R topics documented:

`RFmerge-package` *Merging of Satellite Datasets with Ground Observations using Random Forests*

Description

S3 implementation of the Random Forest MErging Procedure (RF-MEP), which combines two or more satellite-based datasets (e.g., precipitation products, topography) with ground observations to produce a new dataset with improved spatio-temporal distribution of the target field. In particular, this package was developed to merge different Satellite-based Rainfall Estimates (SREs) with measurements from rain gauges, in order to obtain a new precipitation dataset where the time series in the rain gauges are used to correct different types of errors present in the SREs. However, this package might be used to merge other hydrological/environmental satellite fields with point observations. For details, see Baez-Villanueva et al. (2020) <doi:10.1016/j.rse.2019.111606>. Bugs / comments / questions / collaboration of any kind are very welcomed.

Details

Package:	RFmerge
Type:	Package
Version:	0.1-10
Date:	2020-05-21
License:	GPL >= 3
LazyLoad:	yes
Packaged:	Thu May 21 12:28:07 -04 2020; MZB
BuiltUnder:	R version 4.0.0 (2020-04-24) – "Arbor Day" ; x86_64-pc-linux-gnu (64-bit)

Author(s)

Mauricio Zambrano-Bigiarini, Oscar M. Baez-Villanueva

Maintainer: Mauricio Zambrano-Bigiarini <mzb.devel@gmail>

References

Baez-Villanueva, O. M.; Zambrano-Bigiarini, M.; Beck, H.; McNamara, I.; Ribbe, L.; Nauditt, A.; Birkel, C.; Verbist, K.; Giraldo-Osorio, J.D.; Thinh, N.X. (2020). RF-MEP: a novel Random Forest method for merging gridded precipitation products and ground-based measurements, Remote Sensing of Environment, 239, 111610. doi: 10.1016/j.rse.2019.111606. <https://authors.elsevier.com/c/1aKrd7qzSnJWL>.

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., & Gr\"aler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. PeerJ, 6, e5518.

See Also

https://cran.r-project.org/package=raster.
https://cran.r-project.org/package=hydroGOF.

Examples

```
library(rgdal)
library(raster)

data(ValparaisoPPts)
data(ValparaisoPPgis)
data(ValparaisoSHP)

chirps.fname   <- system.file("extdata/CHIRPS5km.tif",package="RFmerge")
prsnncdr.fname <- system.file("extdata/PERSIANNcdr5km.tif",package="RFmerge")
dem.fname      <- system.file("extdata/ValparaisoDEM5km.tif",package="RFmerge")

CHIRPS5km        <- brick(chirps.fname)
PERSIANNcdr5km   <- brick(prsnncdr.fname)
ValparaisoDEM5km <- raster(dem.fname)

covariates <- list(chirps=CHIRPS5km, persianncdr=PERSIANNcdr5km,
                   dem=ValparaisoDEM5km)



# The following code assumes that the region is small enough for neglecting
# the impact of computing Euclidean distances in geographical coordinates.
# If this is not the case, please read the vignette 'Tutorial for merging
# satellite-based precipitation datasets with ground observations using RFmerge'

# without using parallelisation
rfmep <- RFmerge(x=ValparaisoPPts, metadata=ValparaisoPPgis, cov=covariates,
                 id="Code", lat="lat", lon="lon", mask=ValparaisoSHP, training=1)

# Detecting if your OS is Windows or GNU/Linux,
# and setting the 'parallel' argument accordingly:
onWin <- ( (R.version$os=="mingw32") | (R.version$os=="mingw64") )
ifelse(onWin, parallel <- "parallelWin", parallel <- "parallel")
```

```
#Using parallelisation, with a maximum number of nodes/cores to be used equal to 2:
par.nnodes <- min(parallel::detectCores()-1, 2)
rfmep <- RFmerge(x=ValparaisoPPts, metadata=ValparaisoPPgis, cov=covariates,
                 id="Code", lat="lat", lon="lon", mask=ValparaisoSHP,
                 training=0.8, parallel=parallel, par.nnodes=par.nnodes)
```

RFmerge *Merging of satellite datasets with ground observations using Random Forests (RF)*

Description

Merging of satellite datasets with ground observations using Random Forests (RF)

Usage

```
RFmerge(x, ...)

## Default S3 method:
RFmerge(x, metadata, cov, mask, training,
        id="id", lat = "lat", lon = "lon", ED = TRUE,
        seed = NULL, ntree = 2000, na.action = stats::na.omit,
        parallel=c("none", "parallel", "parallelWin"),
        par.nnodes=parallel::detectCores()-1,
        par.pkgs= c("raster", "randomForest", "zoo"), write2disk=FALSE,
        drty.out, use.pb=TRUE, verbose=TRUE,...)

## S3 method for class 'zoo'
RFmerge(x, metadata, cov, mask, training,
        id="id", lat = "lat", lon = "lon", ED = TRUE,
        seed = NULL, ntree = 2000, na.action = stats::na.omit,
        parallel=c("none", "parallel", "parallelWin"),
        par.nnodes=parallel::detectCores()-1,
        par.pkgs= c("raster", "randomForest", "zoo"), write2disk=FALSE,
        drty.out, use.pb=TRUE, verbose=TRUE, ...)
```

Arguments

x	data.frame witht the ground-based values that will be used as the dependent variable to train the RF model. Every column must represent one ground-based station and the codes of the stations must be provided as colnames. class(data) must be zoo.

`metadata` data.frame with the metadata of the ground-based stations. At least, it MUST have the following 3 columns:
-) `id`: This column stores the unique identifier (ID) of each ground-based observation. Default value is `"id"`.
-) `lat`: This column stores the latitude of ech ground observation. Default value is `"lat"`.
-) `lon`: This column stores the longitude of ech ground observation. Default value is `"lon"`.

`cov` List with all the covariates used as independent variables in the Random Forest model. The individual covariates can be a `RasterStack` or `RasterBrick` object when they vary in time, or they can be a single `RasterLayer` object when they do not change in time (e.g., a digital elevation model).
All time-varying covariates in `cov` MUST have the same number of layers (bands). Covariates that do not change in time (e.g., a DEM) are internally trasnformed into `RasterStack` or `RasterBrick` objects with the same number of layers as the other time-varying elements in `cov`

`mask` OPTIONAL. If provided, the final merged product maks out all values in `cov` outside `mask`.
Spatial object (vectorial) with the spatial borders of the study area (e.g., catchment, administrative borders). `class(mask)` must be a `sf` object with "POLYGON" or "MULTIPOLYGON" geometry.

`training` Numeric indicating the percentage of stations that will be used in the training set.
The valid range is from zero to one. If `training = 1`, all the stations will be used for training purposes.

`id` Character, with the name of the column in `metadata` where the identification code (ID) of each station is stored.

`lat` Character, with the name of the column in `metadata` where the latitude of the stations is stored.

`lon` Character, with the name of the column in `metadata` where the longitude of the stations is stored.

`ED` logical, should the Euclidean distances be computed an used as covariates in the random fores model?. The default value is `TRUE`.

`seed` Numeric, indicating a single value, interpreted as an integer, or null.

`parallel` character, indicates how to parallelise 'RFmerge' (to be precise, only the evaluation of the objective function `fn` is parallelised). Valid values are:
-)`none`: no parallelisation is made (this is the default value)
-)`parallel`: parallel computations for network clusters or machines with multiple cores or CPUs. A 'FORK' cluster is created with the `makeForkCluster` function. When `fn.name="hydromod"` the evaluation of the objective function `fn` is done with the `clusterApply` function of the **parallel** package. When `fn.name!="hydromod"` the evaluation of the objective function `fn` is done with the `parRapply` function of the **parallel** package.
-)`parallelWin`: parallel computations for network clusters or machines with multiple cores or CPUs (this is the only parallel implementation that works on Windows machines). A 'PSOCK' cluster is created with the `makeCluster`

function. When `fn.name="hydromod"` the evaluation of the objective function `fn` is done with the `clusterApply` function of the **parallel** package. When `fn.name!="hydromod"` the evaluation of the objective function `fn` is done with the `parRapply` function of the **parallel** package.

`par.nnodes` OPTIONAL. Used only when `parallel!='none'`
numeric, indicates the number of cores/CPUs to be used in the local multi-core machine, or the number of nodes to be used in the network cluster.
By default `par.nnodes` is set to the amount of cores detected by the function `detectCores()` (**parallel** package)

`par.pkgs` OPTIONAL. Used only when `parallel='parallelWin'`
list of package names (as characters) that need to be loaded on each node for allowing the objective function `fn` to be evaluated. By default `c("raster","randomForest","zoo")`.

`ntree` Numeric indicating the maximum number trees to grow in the Random Forest algorithm. The default value is set to 2000. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times. If this value is too low, the prediction may be biased.

`na.action` A function to specify the action to be taken if NAs are found. (NOTE: If given, this argument must be named.)

`write2disk` logical, indicates if the output merged raster layers and the training and avaluation datasets (two files each, one with time series and other with metadata) will be written to the disk. By default `write2disk=FALSE`

`drty.out` Character with the full path to the directory where the final merged product will be exported as well as the training and evaluation datasets. Only used when `write2disk=TRUE`

`use.pb` logical, indicates if a progress bar should be used to show the progress of the random forest computations (it might reduce a bit the performance of the computations, but it is useful to track if everything is working well). By default `use.pb=TRUE`

`verbose` logical, indicates if progress messages are to be printed. By default `verbose=TRUE`

`...` further arguments to be passed to the low level function `randomForest.default`.

Value

It returns a `RasterStack` object with as many layers as time steps are present in `x`. Each one of the layers in the output object has the same spatial resolution and spatial extent as the `cov` argument.

Author(s)

Oscar M. Baez-Villanueva, <obaezvil@th-koeln.de>
Mauricio Zambrano-Bigiarini, <mzb.devel@gmail>
Juan D. Giraldo-Osorio, <j.giraldoo@javeriana.edu.co>

References

Baez-Villanueva, O. M.; Zambrano-Bigiarini, M.; Beck, H.; McNamara, I.; Ribbe, L.; Nauditt, A.; Birkel, C.; Verbist, K.; Giraldo-Osorio, J.D.; Thinh, N.X. (2020). RF-MEP: a novel Random Forest

method for merging gridded precipitation products and ground-based measurements, Remote Sensing of Environment, 239, 111610. doi: 10.1016/j.rse.2019.111606. <https://authors.elsevier.com/c/1aKrd7qzSnJWL>.

Hengl, T., Nussbaum, M., Wright, M. N., Heuvelink, G. B., & Gr\"aler, B. (2018). Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. PeerJ, 6, e5518.

See Also

raster, stack, brick, resample, rotate, crop.

Examples

```
library(rgdal)
library(raster)

data(ValparaisoPPts)
data(ValparaisoPPgis)
data(ValparaisoSHP)

chirps.fname   <- system.file("extdata/CHIRPS5km.tif",package="RFmerge")
prsnncdr.fname <- system.file("extdata/PERSIANNcdr5km.tif",package="RFmerge")
dem.fname      <- system.file("extdata/ValparaisoDEM5km.tif",package="RFmerge")

CHIRPS5km        <- brick(chirps.fname)
PERSIANNcdr5km   <- brick(prsnncdr.fname)
ValparaisoDEM5km <- raster(dem.fname)

covariates <- list(chirps=CHIRPS5km, persianncdr=PERSIANNcdr5km,
                   dem=ValparaisoDEM5km)



# The following code assumes that the region is small enough for neglecting
# the impact of computing Euclidean distances in geographical coordinates.
# If this is not the case, please read the vignette 'Tutorial for merging
# satellite-based precipitation datasets with ground observations using RFmerge'


# without using parallelisation
rfmep <- RFmerge(x=ValparaisoPPts, metadata=ValparaisoPPgis, cov=covariates,
                id="Code", lat="lat", lon="lon", mask=ValparaisoSHP, training=1)


# Detecting if your OS is Windows or GNU/Linux,
# and setting the 'parallel' argument accordingly:
onWin <- ( (R.version$os=="mingw32") | (R.version$os=="mingw64") )
ifelse(onWin, parallel <- "parallelWin", parallel <- "parallel")

#Using parallelisation, with a maximum number of nodes/cores to be used equal to 2:
par.nnodes <- min(parallel::detectCores()-1, 2)
rfmep <- RFmerge(x=ValparaisoPPts, metadata=ValparaisoPPgis, cov=covariates,
```

```
id="Code", lat="lat", lon="lon", mask=ValparaisoSHP,
training=0.8, parallel=parallel, par.nnodes=par.nnodes)
```

`ValparaisoPPgis` *Spatial location of rain gauges in the Valparaiso region (Chile)*

Description

Spatial location of the 34 rain gauges with daily precipitation for the Valparaiso region (dataset *'ValparaisoPPts'*), Chile, with more than 70% of days with information (without missing values)

Usage

```
data(ValparaisoPPgis)
```

Format

A data.frame with seven fields:
*) 'ID : identifier of each gauging station.
*) 'STATION_NAME' : name of the gauging station.
*) 'lon' : easting coordinate of the gauging station, EPSG:4326.
*) 'lat' : northing coordinate of the gauging station, EPSG:4326.
*) 'ELEVATION' : elevation of the gauging station, [m a.s.l.].
*) 'BASIN_ID' : identifier of the subbasin in which the gauging station s located.
*) 'BASIN_NAME' : name of the subbasin in which the gauging station s located.

Details

Projection: EPSG:4326

Source

Downloaded ('Red de Control Meteorologico') from the web site of the Confederacion Hidrografica del Ebro (CHE) http://www.chebro.es/ (original link http://oph.chebro.es/ContenidoCartoClimatologia.htm, last accessed [March 2008]), and then the name of 7 selected fields were translated into English language.

These data are intended to be used for research purposes only, being distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.

ValparaisoPPts *Daily Precipitation Time Series for Valparaiso Region (Chile)*

Description

Daily time series for the year 1983 on 34 rain gauges of the Valparaiso region (Chile), with more than 90% of days with information (without missing values)

Usage

```
data(ValparaisoPPts)
```

Format

A zoo object with 34 columns (one for each rain gauge) and 365 rows (one for each day in 1983). `colnames(ValparaisoPPts)` must coincide with the *ID* column in *ValparaisoPPgis*.

Details

Daily time series of ground-based daily precipitation for 1900-2018 were downloaded from a dataset of 816 rain gauges from the Center of Climate and Resilience Research (CR2; http://www.cr2.cl/datos-de-precipitacion/).
The 34 stations in this dataset wer selected because they had less than 10% of missing values in year 1983.

Source

The CR2 dataset unifies individual datasets provided by Direcci\'on General de Aguas (DGA) and Direcci\'on Meteorol\'ogica de Chile (DMC), the Chilean water and meteorological agencies, respectively.
These data are intended to be used for research purposes only, being distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.

ValparaisoSHP *Administrative borders of the Valparaiso Region (Chile)*

Description

Simple feature polygon collection with 1 feature and 10 fields

Usage

```
data(ValparaisoSHP)
```

Format

SpatialPolygonsDataFrame-class.

The fields stored in the `@data` slot of this object are:

-) *NOM_REG*: Name of the administrative region
-) *NOM_PROV*: Name of the administrative province
-) *COD_COM*: ID of the administrative comune
-) *NOM_COM*: Name of the administrative comune
-) *COD_REGI*: Numeric code of the administrative region
-) *SUPERFICIE*: Spatial area within the administrative borders of the region, [km2]
-) *POBLAC02*: Probably, it corresponds to the number of inhabitants of the region according to the 2002 census.
-) *POBL2010*: Probably, it corresponds to the number of inhabitants of the region according to the 2010 census.
-) *SHAPE_Leng*: Total length of the administrative border of the region, [m]
-) *SHAPE_Area*: Spatial area within the administrative borders of the region, [m2]

Details

Projection: EPSG:4326

Note

The orginal file is not longer available at he Biblioteca del Congreso Nacional de Chile

Source

Originally downloaded from the Biblioteca del Congreso Nacional de Chile and then reprojected into geographic coordiantes (EPSG:4326). Last accessed [March 2016]).
These data are intended to be used for research purposes only, being distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.

Index