Pipeline for soil and climate variable

Bastien BENNETOT

14 avril 2017

How and why to use it?

Purpose of this pipeline

If you have some localization data like longitude and latitude for several observations? You may like to study its repartition. To do this you can get different parameters linked to the location of your observations. That's what this algorithm do. For each coordinate (longitude and latitude), it gives a bunch of soil and climate parameters.

How to format your localization data

First, this algorithm take a table with localization and return a table with different variables for each location added. The entry table must have a column with longitude and another one with latitude. Your table should be in .csv format with "," as separator and "." as decimal. It must contain the two column named "longitude" and "latitude". Variable used come from Harmonized World Soil Database v 1.2 and WorldClim - Global Climate Data.

What to do?

You just have to compute the markdown called "Pipeline to add variable from database.Rmd" in a R software. Work on Rstudio. It will create a dataframe called data_processed.csv in the folder.

Locate your folder

Please change the path. You can find it easily by going to menu : session -> set working directory -> to source file location. You will get in the Console a command like setwd("~/path/to/file/bla/bla") Copy and replace the command below by the given command.

```
setwd("~/Bureau/STAGE M1 IMPORTANT/algo/pipeline")
```

Gathering of data file and import

Please check that all these package are already installed on your R software. You may have to install some software on your computer depending on your distribution.

```
data <-read.table( file = "data.csv", sep = ",",dec = ".", header=TRUE)
library(sp)
library(SDMTools)
library(raster)
```

```
##
## Attaching package: 'raster'
```

```
## The following object is masked from 'package:SDMTools':
##
##     distance
```

```
library(foreach)
library(Hmisc)
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:raster':
##
##     mask, zoom
## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(rgdal)
```

```
## rgdal: version: 1.2-6, (SVN revision 651)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01
##  Path to GDAL shared files: /usr/share/gdal/2.1
##  Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
##  Path to PROJ.4 shared files: (autodetected)
##  Linking to sp version: 1.2-4
```

```
library(RSQLite)
```

Algorithm functionning

Regularly, you don't have to understand next code and explanation to make it work.

Gathering of climate data

Import bioclim variable. The data used are WorldClim 1.4: Current conditions (~1960-1990) with a resolution of 2.5 minutes. Different variables and resolution are available from the website (30 arc-seconds (~1 km)).

```
  #For a new import of worldclim data. Res=resolution.
  #BClim = getData("worldclim", var="bio", res=2.5, path="data/")
  #writeRaster(BClim, filename="YbrevBC_2.5.grd", overwrite=T)

#Import map of bioclim variable from the file
BClim = brick("database/YbrevBC_2.5.grd")
#Indicate position of different observation
  points <- cbind(data$longitude,data$latitude)
#Export variable for position of observation
 clim<- extract(BClim, points)
```

```
#Bind new variable to data
  data<-cbind(data,clim)
```

Gathering of Soil data (harmonized global database)

Harmonized World Soil Database v 1.2 was used. It contains several soil parameters (organic Carbon, pH, water storage capacity, soil depth, cation exchange capacity of the soil and the clay fraction, total exchangeable nutrients, lime and gypsum contents, sodium exchange percentage, salinity, textural class and granulometry)

```
#Import data
hwsd <- raster("database/hwsd.bil")
#Project variable into longitude latitude axis
(proj4string(hwsd) <- "+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0")
```

```
## [1] "+proj=longlat +datum=WGS84 +ellps=WGS84 +towgs84=0,0,0"
```

```
#Export variable for position of observation
soil1<- extract(hwsd, points)
  pointcoord<-cbind(points,soil1)
  colnames(pointcoord)<-c("lon","lat","hwsd")
```

Use of SQLite to repair soil data

For this database, i had to rebuild some SQLite library because it was no more found on internet. It's needed to link location to variable content.

```
  m <- dbDriver("SQLite")
  con <- dbConnect(m, dbname = "database/hwsd.db")
  dbWriteTable(con, name="WINDOW_BHUTAN", value=data.frame(smu_id=pointcoord[1,3]), overwrite=TRUE)
```

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

```
  records.bhutan <- dbGetQuery(con, "select T.* from HWSD_DATA as T join WINDOW_BHUTAN as U on T.MU_GLO
  tablesoil<-records.bhutan
  tablesoil<-tablesoil[0,]
    for (i in 1:nrow(pointcoord)){
    dbWriteTable(con, name="WINDOW_BHUTAN", value=data.frame(smu_id=pointcoord[i,3]), overwrite=TRUE)
    records.bhutan <- dbGetQuery(con, "select T.* from HWSD_DATA as T join WINDOW_BHUTAN as U on T.MU_G
 tablesoil<-rbind(tablesoil,records.bhutan[1,])}
```

Supress useless soil variable for reference (OPTIONNAL)

Some column are used as reference, see Harmonized World Soil Database v1.1.pdf for more information.

```
  tablesoil$id<-NULL
  tablesoil$mu_global<-NULL
  tablesoil$mu_source1<-NULL
  tablesoil$mu_source2<-NULL
  tablesoil$coverage<-NULL
  tablesoil$issoil<-NULL
  tablesoil$seq<-NULL
  tablesoil$share<-NULL
  tablesoil$su_code74<-NULL
  tablesoil$su_code85<-NULL
```

```
tablesoil$su_code90<-NULL
tablesoil$su_sym74<-NULL
tablesoil$su_sym85<-NULL
tablesoil$su_sym90<-NULL
```

Add column of soil variable to data

```
data<-cbind(data,tablesoil)
```

Some extra variable from harmonized global database sup

Some soil parameter are not in the standard soil database but they are available below "A selection of the data is downloadable and available here :" in the website

```
varsup = brick("database/varsup.grd")
varsupex<- extract(varsup, points)
data<-cbind(data,varsupex)
```

Save data with variable in data_processed.csv

Everything is done. A new data file will be written. It's called data_processed.csv (separator=";")

```
write.table(data, file = "data_processed.csv", sep = ";")
```

Bibliography

1. Hijmans RJ, Cameron SE, Parra JL, Jones PG, Jarvis A: Very high resolution interpolated climate surfaces for global land areas. International Journal of Climatology 2005, 25:1965–1978. [doi: 10.1002/joc.1276]
2. Nachtergaele FO, van Velthuizen H, Verelst L, Batjes NH, Dijkshoorn JA, van Engelen VWP, Fischer G, Jones A, Montanarella L, Petri M, et al.: Harmonized world soil database (Version 1.0) [Internet]. Food and Agric Organization of the UN (FAO); International Inst. for Applied Systems Analysis (IIASA); ISRIC-World Soil Information; Inst of Soil Science-Chinese Acad of Sciences (ISS-CAS); EC-Joint Research Centre (JRC); 2008.