Generation of test landscape rasters for LANDIS-II

Alec Kretchun, modified by Vincent Bisquay Gracia

2016, modified 2021

I. Set up

I.a) Packages

```
library(raster)
library(rgdal)
library(magrittr)
```

I.b) Directories settings

The input folder (*input_dir*) and the output one (*save_dir*) are identified. The list *ListRaster*, containing the names of the original rasters, is created.

```
input_dir = "E:/FIF/Stage 3A/Extract/Documents/Klamath_(CA_only)_2021 - CC only/..1/"

save_dir = "E:/FIF/Stage 3A/Results single cell/CC_only_single_cell/"

wd = "D:/Internship 3A/GitHub/Project-Klamath-2021/Model Parameterization"

setwd(wd)
ListRaster = list.files(paste(input_dir,sep=""),pattern = ".tif$")

Int = c("ecoregions_v2.tif", "full_suppression.tif","klamath_IC.tif",
"klamath_slope_CAonly_150m_v2.tif", "klamath_upslope_CAonly_150m_v2.tif")
Fact = c("ecoregions_v2.tif", "klamath_IC.tif")
```

I.c) Dimentions of output rasters

Here the objective is to make sigle cell rasters, but this number can be increased changing these variables.

```
side1 = 1
side2 = 1
```

II. Creating single cells rasters

I.a) Reference layer

Somes of the layers doesn't have a proper coordonate sistems (there are more images than real rasters). A raster of reference with a defined and correct projection system is selected (here *RasterRef*, corresponding to the first raster of the list). All the values of interest are then stored.

```
RasterRef = paste(input_dir,ListRaster[1],sep="") %>% raster()
# the first raster of the list has a proper definition.
#It will serve as reference for the rasters that doesn't have one.

CRSRef = crs(RasterRef)
NrowRef = RasterRef@nrows
NcolRef = RasterRef@ncols
ResRef = res(RasterRef)[1]
XminRef = RasterRef@extent@xmin
XmaxRef = RasterRef@extent@xmax
YminRef = RasterRef@extent@ymin
YmaxRef = RasterRef@extent@ymax
```

I.a) Signle cell rasters

I.a.1) General method

for all the files in *ListRaster*, the caracteristic of the raster of reference are used to make sure all have the same coordinates and projection system. Then, the input raster is changed into a matrix and all the "NA" and "0" values are removed.

The mean of all the values of the matrix, *r_value* is used as the value of the single cell raster. This later is then exported in the output folder.

```
ValuesList = c("File Name", "Created Value", "Exported Value", "Type")

for (file in ListRaster) {

  Name = file
  RasterTot_NoDef = paste(input_dir,file,sep="") %>% raster()
  RasterTot = raster(vals=getValues(RasterTot_NoDef),
                     nrow=NrowRef,ncol=NcolRef,resolution=ResRef,
                     xmn=XminRef, xmx=XmaxRef,ymn =YminRef,ymx=YmaxRef,crs=CRSRef)
  # The raster is changed to have the same definition than the reference

  RasterM = as.matrix(RasterTot)
  Raster_noNa_no0 = RasterTot[!is.na(RasterTot) & RasterTot!=0] #without NA and 0
  if (file %in% Fact){
    r_value = sample(Raster_noNa_no0, size = 1) %>% as.double
    # The IC is a factor, thus calculating a mean makes no sense
  } else {
    r_value = mean(Raster_noNa_no0) %>% as.double
  }

  NewRaster = matrix(r_value, nrow=side1, ncol= side2) %>% raster
```

```
  if (file %in% Int) {
    Type = "Integer"
    writeRaster(NewRaster, file=paste(save_dir,Name, sep=""),
                format = "GTiff", datatype='INT4S',overwrite=T)
  # The field capacity need to be exported differently
  }else {
    Type = "Float"
    writeRaster(NewRaster, file=paste(save_dir,Name, sep=""),
                format = "GTiff", datatype='FLT4S',overwrite=T)
  }
  VerifRaster = paste(save_dir,file,sep="") %>% raster()
  VerifValue = VerifRaster@data@min

  ValuesList = rbind(ValuesList, c(Name, r_value, VerifValue, Type))
}

ValuesList = as.data.frame(ValuesList)
colnames(ValuesList)=ValuesList[1,] %>% unlist %>% as.character
ValuesList=ValuesList[-1,]
```

I.a.2) Verification and particular cases

A dataframe *Values list* is created with the values for all the created rasters, allowing to better understand the errors that LANDIS-II can show.

```
head(ValuesList, n=4)
```

```
##               File Name       Created Value     Exported Value    Type
## X          basefrac.tif 0.00999999977648258 0.0099999997764826   Float
## X.1 deadcoarseroots.tif    1160.37616945045    1160.3761694505   Float
## X.2          depth2.tif    46.7042372229382    46.704237222938   Float
## X.3   ecoregions_v2.tif                 133                133 Integer
```

Runing with the general method lead to 2 errors fixed here :

i) The value of *fieldcap2.tif* is not between 0.0 and 1.0 This is due to the datatype *INT4S* of the *writeRaster* fonction that round the mean value to 0. This datatype is compulsory for a majority of other rasters (LANDIS-II don't read them otherwise). The only choice is then to export specifically this layer with another datatype (*FLT4S*). To avoid this issue, alist of all the rasters that need an integer value (*Int*) is created. This is done reading the user guide manuals as the type of data needed is detailed.

ii) The Initial Community is not known This is due to the calculation of a mean for a value that is initially a factor (a map code). One solution is to take this raster apart and pic one value of initial community randomly (this is the case here). Another solution can be to calculate a median value instead of a mean. Similarly to the precedent case, a list of all the rasters needing a map code is created (*Fact*).