A great part of my job as a bat ecologist is to classify bat species from their echolocation calls. I regularly use automatic recording devices that generate thousands of recordings per location. Dealing with this huge amount of information is not an easy task as you can imagine. In the old days each recording was manually analysed and the bat species classified using some sort of identification key. All this changed with the advent of automatic classification systems. Nevertheless, due to the great intra-specific variability of echolocation calls, the error rate of these systems is still quite high and a manual check is still needed. What I normally do is to randomly select 10% to 20% of the recordings and manually classify them to check the overall error rate of the automatic classification. This presents the challenge of: 1) selecting the files to check and 2) to copy them to a new folder for easy handling. I’ve prepared a function and a working script for this end that I’ll present and explain in this post.

random\_files() is a function that receives as parameters the path of the folder containing the files, the % or number of files to select and the extension of the files to be selected. This function creates a new folder called “selected” inside the original folder containing all the files, randomly selects and moves the desired % or number of files to that folder and creates a file called “selected\_files.csv” with the names of the selected files.

random\_files <- function(path, percent\_number, pattern = "wav$|WAV$"){

####################################################################

# path = path to folder with files to select

#

# percent\_number = percentage or number of recordings to select. If value is

# between 0 and 1 percentage of files is assumed, if value greater than 1,

# number of files is assumed

#

# pattern = file extension to select. By default it selects wav files. For

# other type of files replace wav and WAV by the desired extension

####################################################################

# Get file list with full path and file names

files <- list.files(path, full.names = TRUE, pattern = pattern)

file\_names <- list.files(path, pattern = pattern)

# Select the desired % or number of file by simple random sampling

randomize <- sample(seq(files))

files2analyse <- files[randomize]

names2analyse <- file\_names[randomize]

if(percent\_number <= 1){

size <- floor(percent\_number \* length(files))

}else{

size <- percent\_number

}

files2analyse <- files2analyse[(1:size)]

names2analyse <- names2analyse[(1:size)]

# Create folder to output

results\_folder <- paste0(path, '/selected')

dir.create(results\_folder, recursive=TRUE)

# Write csv with file names

write.table(names2analyse, file = paste0(results\_folder, "/selected\_files.csv"),

col.names = "Files", row.names = FALSE)

# Move files

for(i in seq(files2analyse)){

file.rename(from = files2analyse[i], to = paste0(results\_folder, "/", names2analyse[i]) )

}

}

I normally use this function inside a little script for some extra functionalities:

1. first I set up the environment by sourcing the required functions and loading the packages,
2. as I always do when using functions that use randomness, I set a seed to be able to reproduce my results in a later time,
3. as the function uses a folder path, I’ve included a little search window with tcltk to select the folder instead of having to write the full path by hand.

# Load packages and functions

require(tcltk)

source("random\_files.R")

# Set seed to reproduce results

set.seed(1001)

# Select folder with recordings

path <- tcltk::tk\_choose.dir()

# Percentage or number of recordings

percent\_number <- 0.2 # using percentage

# Random sampling of files

random\_files(path, percent\_number, pattern = "wav$|WAV$")

This function was written for a specific purpose but with some tweaks you can probably adapt it for other purposes other than the one I use it for.