

R CODE: GRTS Identifier for NABat Mobile Transects

This script is designed to identify the NABat GRTS Cell in which a call was recorded using the associated latitude and longitude (decimal degrees). First, copy the code on the following page, beginning with 'install.packages ('rgdal')' and ending with 'missing_files.' Next, open R Studio, click the green + button under 'File,' and paste the code into the new script. The code requires users to load a tab delimited .txt file that contains three columns: the first column should contain the file names, the second column should contain the associated latitude, and the third column should contain the associated longitude. This file must be stored in the same folder that contains your call files. If your lat/long data is contained in a .csv file, replace the code on line 29 (gps_df = read.csv (file_name, sep = '\t') with the following code: gps_df <- read.csv(file_name). If your lat/long data is formatted with N/W rather than +/-, the code will automatically reformat those values.

File Name	Latitude	Longitude
3773_NWRD_20180817_202223.wav	40.3456	-103.1234
3773_NWRD_20180817_202315.wav	40.3468	-103.1212
3773_NWRD_20180817_202355.wav	40.3482	-103.1278

Lines 1-11 will install and load packages required for the script to run. Once packages are successfully installed, lines 1-5 can be deleted or ignored for future use. Lines 7-11 (beginning with 'library') must be run once each time R is opened. Once packages are loaded, lines 7-11 can be ignored as long as the R session remains open.

The script will create a new folder with copies of your acoustic files sorted into folders by GRTS Cell. In line 14, copy the file path where you would like the new folder to be created. **Windows automatically replaces forward slashes (/) with backslashes (\). This will create errors in R. After you copy and paste the file path, press Ctrl + F to open the find/replace function in R and substitute forward slashes for backslashes in the file path.** At the end of the file path, insert a / and what you wish the new folder to be named.

e.g. out_dir_name = "C:/Users/danesmith/documents/new_folder"

Next, copy and paste the file path to the folder that contains the acoustic files and text file with lat/long values into line 22. **This file path MUST end in a forward slash (/) or the code will not run properly.**

e.g. file_name_dir = 'C:/Users/danesmith/documents/Calls/Transect_1/'

Finally, copy and paste the file path from line 22 into line 25 and add the name of the file containing your lat/long values after the backslash. Be sure to include the file extension (.txt/.csv).

e.g. file_name = 'C:/Users/danesmith/documents/Calls/Transect_1/Transect_1_GPS.txt'

The script also references a spatial data file to identify the GRTS cell that corresponds to each lat/long. Before running the script, users must download the 'conus_mastersample_10km_GRTS.zip' file at <https://www.sciencebase.gov/catalog/item/5b7b563ae4b0f5d57884615b>, extract all files, and store them locally. Users should then copy and paste the file path into line 50 of the script (CONUS = rgdal::readOGR('INSERT FILE PATH HERE'))

Lastly, this code will create a new Excel CSV file containing a column of file names, latitude, longitude, and the GRTS cell in which the file was recorded. Insert your desired name for this file in line 114. Be sure the desired file name ends in .csv.

e.g. write.csv(final_wav_df, paste0(out_dir_name, '/Insert_Desired_File_Name_Here.csv'))

Code (copy and paste into new R script):

```
install.packages('rgdal')
install.packages('leaflet')
install.packages('sp')
install.packages('raster')
install.packages('rgeos')

library(plyr)
library(sp)
library(raster)
library(leaflet)
library(rgdal)

# Location for data created from this script
out_dir_name = "E:/R_FILES/TEST"

# Create the out_dir_name above
dir.create(out_dir_name)

# Location of the .txt file containing file names and lat/long values
# The file_name_dir should lead to the folder containing the file
# Be sure the file path ends with forward slash (/)
file_name_dir = 'E:/WICA#/DATA/Road/2018/WICA033842RD/WICA033842RD_20180807/'
# The file_name directory should be the same as the file_name_dir but
# end with the name of the .txt file containing the file names and lat/long values
file_name = 'E:/WICA#/DATA/Road/2018/WICA033842RD/WICA033842RD_20180807/WICA033842RD_gps.txt'

# Read in the .txt file as a csv using the \t as the delimiter
gps_df = read.csv(file_name, sep = '\t')
gps_df
file_name

#Format the lat/long columns
no_space_lat = gsub(" ", "", gps_df$LATITUDE, fixed = TRUE)
clean_lat = as.double(gsub("N", "", no_space_lat, fixed = TRUE))
gps_df$LATITUDE = clean_lat

no_space_lon = gsub(" ", "", gps_df$LONGITUDE, fixed = TRUE)
clean_lon = as.double(gsub("W", "", no_space_lon, fixed = TRUE))

if (clean_lon[1] < 0){
  gps_df$LONGITUDE = clean_lon
}else {
  gps_df$LONGITUDE = clean_lon * -1
}
```

```

# Make sure that the order is lng/lat
lng_lat = gps_df[,c('LONGITUDE','LATITUDE')]

# Building a spatial points dataframe to intersect with the CONUS GRTS Shapefile
these_points = sp::SpatialPointsDataFrame(coords = lng_lat, data = gps_df,
                                           proj4string = CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0"))

# The following lines require the NABat conus mastersample file, available at
# www.sciencebase.gov/catalog/item/5b7753bde4b0f5d578820455. Store the file anywhere on your machine
# and paste the file path in the CONUS line below
CONUS = rgdal::readOGR('C:/Users/danesmith/Desktop/conus_mastersample_10km_GRTS') # Takes a min
CONUS_1 = spTransform(CONUS, CRS("+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"))

# Intersect the GRTS shapefile by the mobile transect points
grts_cells = raster::intersect(CONUS_1, these_points)
mobile_points = raster::intersect(these_points, CONUS_1)
# Display for accuracy
leaflet() %>% addTiles() %>% addCircleMarkers(data = mobile_points) %>% addPolygons(data = grts_cells)
# Create final dataframe
final_combined_df = as.data.frame(mobile_points)

# Clean final combined dataframe
#final_df = select(final_combined_df, c('NAME', 'LATITUDE', 'LONGITUDE', 'GRTS_ID'))
final_df<-data.frame(NAME=final_combined_df$NAME,LATITUDE=final_combined_df$LATITUDE,
LONGITUDE=final_combined_df$LONGITUDE, GRTS=final_combined_df$GRTS_ID)

# Get unique grts ids to loop through
grts_ids = unique(final_df$GRTS)
# Loop through grts ids and their wav files to find
final_wav_df = data.frame()
for (id in grts_ids){
  wav_exists_df = data.frame()
  grts_df = subset(final_df, final_df$GRTS == id)
  grts_dir = paste0(out_dir_name, '/',id)
  dir.create(grts_dir)
  trues = 0
  falses = 0
  for (row in 1:dim(grts_df)[1]){
    this_row = grts_df[row, ]
    this_file_name = as.character(this_row$NAME)
    # If the file exists copy the wav file to the new output dir
    if (file.exists(paste0(file_name_dir, this_file_name))){
      trues = trues + 1
    }

    # Build new dataframe iteratively if file exists
    if (dim(wav_exists_df)[1]==0){
      wav_exists_df = this_row
    }
  }
}

```

```

} else{
  wav_exists_df = rbind(wav_exists_df, this_row)
}

# If the file doesn't exist
}else {
  falses = falses + 1
}
}
# Save wave files to directory
file.copy(paste0(file_name_dir, wav_exists_df$NAME), grts_dir)
# Sanity check (aka QC)
print (paste0('-----', id, '-----'))
print (paste0('number of real wav files: ', trues))
print (paste0('length of new dir: ', length(list.files(grts_dir))))
print (paste0('number of non existent wav files: ', falses))
print (paste0('number of wav files found in this grts: ', dim(grts_df)[1]))
if (dim(final_wav_df)[1]==0){
  final_wav_df = wav_exists_df
} else{
  final_wav_df = rbind(final_wav_df, wav_exists_df)
}
}
final_wav_df
# This line creates a CSV file containing columns for file name, lat, long, and GRTS Cell
# Name the file as you wish (replace GPS_File.csv with name of your choice)
write.csv(final_wav_df, paste0(out_dir_name, '/20180724_gps.csv'))

# Find any files that didn't get moved into new GRTS folders
initial_wav_files = as.character(strsplit(list.files(file_name_dir), file_name_dir))
final_wav_files = as.character(final_wav_df$NAME)
missing_files = setdiff(initial_wav_files, final_wav_files)
missing_files

```