```
library(raster)
library(ggplot2)
library(reshape)
library(dplyr)

# This parses the model outputs and creates collated raster(s).

# Clear the environment to avoid errors from old files
rm(list=ls())

# Browse to the DEM
DEM <- raster(file.choose(), sep=",")

# Get the dimensions of the raster to collate the route data
dem.x <- dim(DEM)[2]
dem.y <- dim(DEM)[1]

# Browse to and read the main output file created by BehaviorSpace
master <- read.csv(file.choose(), sep=",", skip = 6, stringsAsFactors = F)

# Remove the double quote imported, which is necessary to filter runs
master$optimization <- gsub("^\"|\"$", "", master$optimization)

# To iterate over different datasets and create raster for each optimization (if applicable)
optimization <- unique(master$optimization)

# Assign the folder in which all the individual simulation output files are located
my_dir = "[write path to folder here]"

# Create a list of all the file names in the identified folder
all_files = list.files(path = my_dir, all.files = TRUE, full.names = TRUE, pattern = "\\.csv$")

# Give the number of files to analyze
list_size = length(all_files)

for (op in optimization){

  # Create a temporary dataframe that will take on the collated coordinate and popularity values
  dat.final <- data.frame(x=double(0),
                          y=double(0),
                          value=double(0))

  # Create a temporary dataframe that will help create raster
  routes <- data.frame(x=double(0),
                       y=double(0),
                       value=double(0))

  ## Create the progress bar
  pb <- txtProgressBar(min = 0, max = list_size, style = 3)

  # Iterate over all the files located in the given folder
  for(l in 1:list_size[1]){
```

```
  # Reformat the name of the files so that it can be used later
  file.name = strsplit(all_files[l],"/")
  file.name = unlist(file.name)
  name.size = length(file.name)
  new.file = file.name[name.size]

  # Separate out the components of the file name
  filename.split = strsplit(new.file,"_")
  filename.split = unlist(filename.split)
  opt = filename.split [3]
  origin = gsub("\\)|\\(", "", filename.split[6]) # Origin of the run
  goal = gsub("\\)|\\(", "", filename.split[7]) # Goal of the run

  # If that specific run is of the correct optimization, read it and collate its data. If it is not, skip it.
  if (op == opt){

    # Import the data without headers
    ds <- read.table(paste(my_dir,new.file,sep=""), fill = TRUE, skip = 19, stringsAsFactors = FALSE, sep = ",")

    # Keep only the coordinates of the paths
    ds <- ds[,c(2,6)]

    # Change the names and reduce the floats coordinates to integers
    colnames(ds) <- c("x","y")
    ds$x <- as.integer(ds$x)
    ds$y <- as.integer(ds$y)

    # Remove duplicates cells created because of knight movements that often stops on cell edges before reaching
its destination
    ds <- ds [!duplicated(ds[c(1,2)]),]

    # Extract the coordinates of the path's start and end points
    start.x.raw <- unlist(strsplit(gsub("[[:punct:]]", "", origin), " "))
    start.x <- start.x.raw[2]
    start.y <- start.x.raw[3]
    end.x.raw <- unlist(strsplit(gsub("[[:punct:]]", "", goal), " "))
    end.x <- end.x.raw[2]
    end.y <- end.x.raw[3]

    # Remove the origin and goal's patches (if present) to avoid skewing the data
    ds<-ds[!(ds$x==start.x & ds$y==start.y),]
    ds<-ds[!(ds$x==end.x & ds$y==end.y),]

    # For each path, each cell is walked on only once
    ds$value <- 1

    # If this is not an empty dataset
    if(nrow(ds) > 1){

      # Add this new path to the big dat dataset
      routes <- rbind(routes,ds)

      # Then group by coordinates and sum up the number of times each cell is walked on
```

```
      route.group <- group_by(routes, x, y)
      b <- dplyr::summarize(route.group, value = sum(value))
      routes <- as.data.frame(b)

      # Assign the dataset to the global environment so it can be used outside the loop.
      assign('routes',routes, envir = .GlobalEnv)
    }
   }

   # Update progress bar
   setTxtProgressBar(pb, l)
 }

 ################################################
 ## USING ROUTES TO IDENTIFY MOST POPULAR PATH ##
 ################################################
 print("creating route") # Show progress
 dat <- routes

 # Change the name of the dat file because we will add new columns
 colnames(dat) <- c("long","lat","value")

 # Ensure that the x and y columns are numeric
 dat$x <- as.numeric(as.character(dat[,1]))
 dat$y <- as.numeric(as.character(dat[,2]))

 # Change the order of the dat file to have x,y,value.
 dat <- dat[,c(4,5,3)]

 # Transform the times walked on into a 0-1 value (divide by the max times walked)
 dat$value <- dat$value / max(dat$value)

 # Create the final dataset and remove the dat dataset to avoid errors in subsequent loop iterations
 dat.final <- dat
 rm(dat)

 # Transform into a raster with the same coordinates as the imported DEM
 dat.final$x <- (dat.final$x * xres(DEM) ) + xmin(DEM) + (xres(DEM) / 2) # xmin extent of the original map
 dat.final$y <- (dat.final$y * yres(DEM) ) + ymin(DEM) + (yres(DEM) / 2) # ymin extent of the original map

 # Create the raster
 r.sub <- rasterFromXYZ(dat.final)

 # Output it
 writeRaster(r.sub, paste("[write path to folder here]/FETE_all_",op,".asc", sep = ""), overwrite = T)
}
```