

fitcorridor vignette v1.0.0

Tristan Nuñez

2022-07-12

This vignette accompanies the article “A statistical framework for predicting migration corridors” by Nuñez, T., Hurley, M., Graves, T. Ortega, A. Sawyer, H., Fattebert, J., Merkle, J., and Kauffman, M., accepted for publication in *Methods in Ecology and Evolution*, and the fitcorridor code to go with it.

The fitcorridor code fits cost-distance models to animal movement tracks.

The fitcorridor model fitting code will be updated in the future (including conversion to R package format). Check the github page for the latest version at github.com/tristannunez/fitcorridor.

Getting Started

Review the manuscript, which outlines the calculations and algorithms involved in fitting cost-distance models to movement tracks; Figure 1 and Sections 2.2 through 2.8 of the Methods section are particularly relevant.

Prerequisites

The code requires the following R packages be installed: gdistance, parallel, rgdal, ggplot2, ggspatial, and rgeos. Make sure they are installed before proceeding.

Let's fit a cost-distance model!

We'll begin by fitting a cost distance model to a set of simulated movement tracks. The model we will fit is $conductance = e^{\beta \cdot HumanFootprint}$. For each movement track, the code will find the value of β that maximizes the likelihood of the data given the model. It will also find the optimal value of λ , the rate parameter of the exponential distribution used to model the probability of deviations from the least cost path (see Methods Section 2.4 in the manuscript). $1/\lambda$ is the mean residual cost distance of the fitted distribution, allowing us to directly assess how far on average an animal deviates from the modeled least cost path in cost distance units.

Setting up

First, make sure your working directory is the fitcorridor folder:

```
getwd()
#> [1] "C:/Users/Tristan/Desktop/github/fitcorridor"
```

Source the R file containing the functions.

```
source("./fitcorridor_functions.R")
```

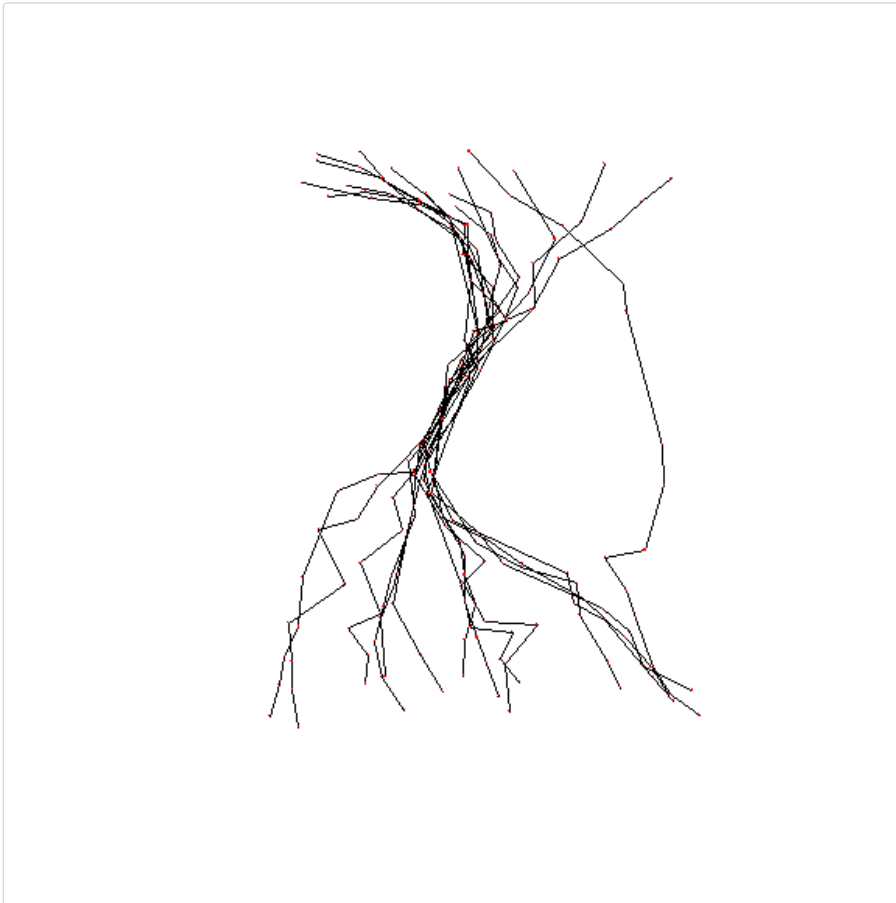
Read in the animal movement tracks. In this case, we will be using a set of 15 tracks simulated from a human footprint surface.

```
tracks<-readOGR("./data/tracks/HF_sim_tracks.shp", stringsAsFactors = F)
#> OGR data source with driver: ESRI Shapefile
#> Source: "C:/Users/Tristan/Desktop/github/fitcorridor/data/tracks/HF_sim_tracks.shp", Layer:
      "HF_sim_tracks"
#> with 208 features
#> It has 4 fields
# Set the coordinate reference system
mycrs <- "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs +towgs84=0,0,0"
crs(tracks) <- mycrs
# turn the tracks into SpatialLinesDataFrame just for plotting
tracks$id <- tracks$trackid # TrackSPDFtoTrackSPLLines requires an $id column
```

```

tracks.as.lines <- TrackSPDFtoTrackSPLFLines(tracks)
plot(tracks.as.lines)
plot(tracks, add=T, col="red", pch=".", cex=1.2)

```

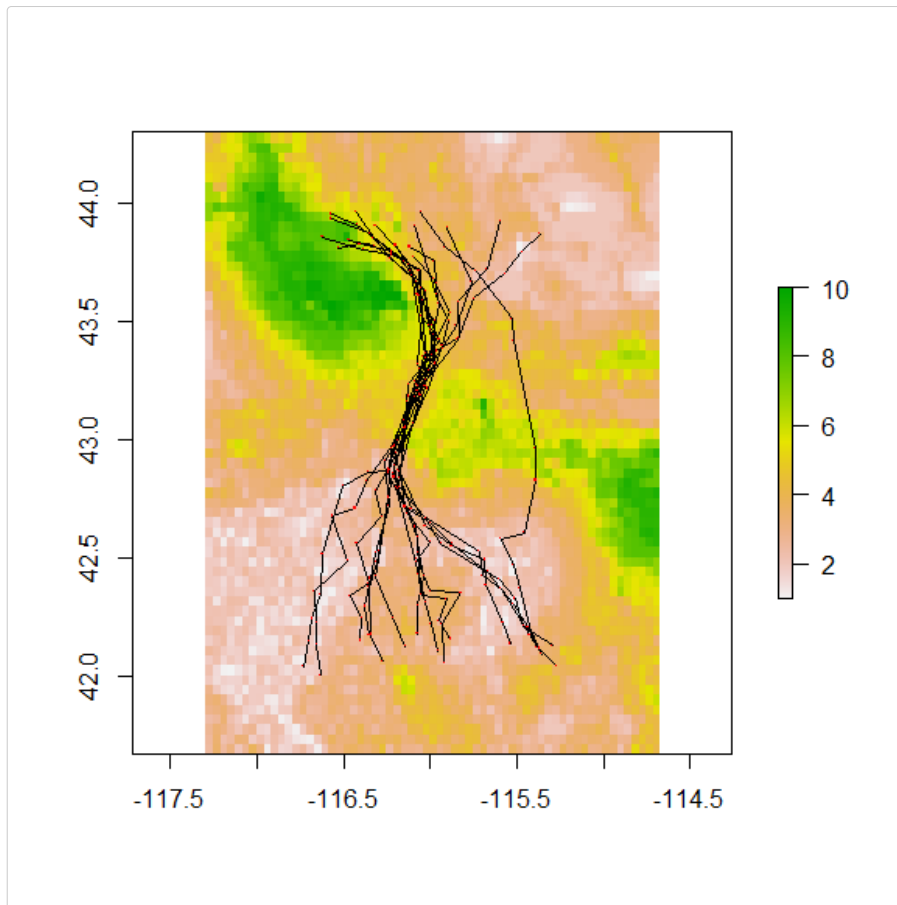


Read in the environmental covariates. Here, we will read in the human footprint surface previously used to simulate the movement tracks. This data is from Leu et al. 2008 (<https://doi.org/10.1890/07-0480.1>, <http://sagemap.wr.usgs.gov/>), cropped to the Boise, ID area.

```

# Higher values represent more human influence
covars <- brick("./data/covariates/HF.tif")
# specify the Coordinate Reference System
crs(covars) <- mycrs
# assign names to layers
names(covars) <- c("HF") # if more than one layer, use this format: c("HF", "DEM")
# plot to check
plot(covars)
plot(tracks.as.lines, add=T, cex=.75)
plot(tracks, add=T, pch=".", col="red", cex=1.2)

```



The simulated tracks, which move between randomly selected start and end points in the upper and lower quarter of the human footprint raster, avoid areas of high human footprint. This is because they were simulated using a β value of -1.

Running the model fitting code

Now that we have read in the inputs, we can fit the cost distance model $conductance = e^{\beta \cdot HumanFootprint}$ to the movement tracks. We do this using the `FitSingleModel` function:

```
FitSingleModel(
  tracks = tracks,
  covars = covars,
  model.to.fit = "iso(HF)",
  neighbors = 8,
  covar.aggregate = 1,
  covar.standardize = c(1),
  param.factor = c(1),
  startval.string = c(-5, 0),
  dataset = "SimHerd1",
  out.folder.path = "./fittedmods/",
  n.inside.cores = 4,
  Lt.tiff = T
)
#> Warning in .local(x, ...): all fact(s) were 1, nothing to aggregate
```

This takes around ten minutes on a PC laptop (Intel(R) i7 CPU @ 2.90GHz with 4 cores). Disregard the “In `.local(x, ...)`: all fact(s) were 1, nothing to aggregate” warning. Let’s explore each of the arguments for the `FitSingleModel` function:

tracks

A `SpatialPointsDataFrame`. The dataframe needs to have a ‘trackid’ column with a unique id for each movement