Describing spatiotemporal memory patterns1using animal movement modelling: Supplementary Material

Peter R. Thompson, Andrew E. Derocher, Mark A. Edwards, Mark A. Lewis

May 6, 2021

This repository contains all the code necessary to replicate the analyses conducted in our manuscript, "Describing spatiotemporal memory patterns using animal movement modelling". This readme serves as a guide on how and when each file should be used. Below is a list of files included in the directory. The R code used here also depends on a number of packages, which are listed below:

- NLMR (generating of spatially autocorrelated random landscapes for simulation)
- raster (reading in real-life environmental data formatted as rasters)
- circular (simulating from the von Mises distribution)
- fdrtool (simulating from the half-Gaussian distribution)
- VGAM (optional - simulating from the Rayleigh distribution)
- fields (producing heatmaps of environmental layers as well as likelihood surfaces)
- TMB (fitting models)
- doParallel (optional - parallel programming)
- ggplot2 (optional - producing violinplots of parameter estimates)
- qPCR (optional - calculating Akaike weights for model selection)
- gplots (optional - 2-D Bhattacharyya coefficient)

In order for this code to work you must also have Rtools (https://cran.r-project.org/bin/windows/Rtools/) installed on your computer. Make sure to add Rtools to your PATH!

Some of these scripts take some time (in total, more than a day) to run, although parallel programming or remote programming can speed this up.

Data

- `NLMs` (directory): Contains three simulated landscapes generated from the R NLMR package. All three of these landscapes were used in the analysis for Section 3.3 (along with many others). Additional NLMs can be generated using the `write.NLM` function.
- `inputs` (directory): Contains the six true environmental rasters used for all the grizzly bear model fits.
- `GIFs` (directory): Contains simulated movement tracks (in the form of animated GIFs) from each model (null, resource-only, memory-only, resource-memory) from `grizzly_simHMM_20210506.R`. These can be recreated by toggling GIF = TRUE, although make sure to set the seed properly to get the same exact graphs.
- `outputdata_given` (directory): Contains the model fitting outputs for the four simulated tracks. Rerunning this script will produce the same files in the main folder.
- `bears_csv` (directory): Contains the raw bear data to be passed into the `grizzly_preparedata.R` script. We include data for the eight bears included in this study.

- `grizzly_pars_forsim.csv`: Continas the parameter estimates (some are converted to their original units for modelling purposes) for each of the bears included in this analysis. This CSV file is intended to be used for `grizzly_simbasedonrealpars_20210506.R`.

Scripts

- `sim_functions_20200326.R`: This file serves as the main engine for the analysis, containing all the functions necessary for data manipulation, simulation of animal movement data, and model fitting.
- `grizzly_simHMM_20210506.R`: This file simulates movement tracks according to the four different models (null, resource, memory, resource-memory) and fits all the models to each track. This script is used to generate the analysis from Section 3.3.
- `grizzly_preparedata_20200423.R`: This file takes in the raw bear data along with the raw environmental data. Using `sim_functions_20200326.R`, this script prepares the data for analysis with the `fit.SSF` function. This script also generates available points for each used step, which will of course vary with each random seed.
- `grizzly_fitactualbears_20200424.R`: This is the file used to fit the model to the grizzly bear data. **You must run `grizzly_preparedata_20200423.R` first for this to work properly.** The output is stored as a CSV file (similar to `grizzly_simHMM_20210506.R`).
- C++ files (`grizzly_*modelname*_*date*.cpp`): These are compiled in the generateFunction function (from `sim_functions_20200326.R`) and used to efficiently calculate the negative log-likelihood of each model.
- `distributions_R_new.hpp`: A helper function for Template Model Builder. Includes the manually coded von Mises distribution, which is used in our model. This script is included manually in each of the C++ files so they compile properly.
- `grizzly_subsetbeardata_20201013.R`: A small script that subsets each of the bear movement tracks as is done in Section 3.3.2. **You must run `grizzly_preparedata_20200423.R` first for this to work properly.**
- `grizzly_fitsubsetbears_20201013.R`: The analog of `grizzly_fitactualbears_20200424.R` for the subsetted bear data. Produces a file output of the model fits. **You must run `grizzly_preparedata_20200423.R` AND`grizzly_subsetbeardata_20201013.R`‘ first for this to work properly.**
- `grizzly_simbasedonrealpars_20210506.R`: Carries out the simulation portion of the analysis done in Section 3.3.2. Uses the real parameters (`grizzly_pars_forsim.csv`) from the model fit and simulates many random tracks. **You must run `grizzly_preparedata_20200423.R` first for this to work properly.**
- `grizzly_profileCI_20200823.R`: Computs confidence intervals for each parameter in each bear (only for the best model) using profile likelihood. This takes a **very** long time (especially without parallel programming). **You must run `grizzly_preparedata_20200423.R` first for this to work properly.**

If anything is not working please contact Peter Thompson (pt1@ualberta.ca).