

1 PISCR.fn - Fits spatially-explicit capture recapture partial ID model

Function `PISCR.fn` requires user to specify the variables for the MCMC iterations. MCMC samples from the posterior distributions of the parameters are generated and saved as files, together with summary statistics.

1.1 R codes

1. ***utility.functions.R***: It has user defined functions such as log-likelihood function (viz. `logLfn.da`), Euclidean distance between two points (viz. `e2dist`), function which provides a good choice for initial value of 'L' vector (viz. `Lvec`) etc. The required R packages are - 'mvtnorm', 'MCMCpack', 'abind', 'ggplot2', 'raster', 'maptools', 'coda', 'spatstat', 'gtools'.
2. ***partialID.simdata.R***: This R code contains a function - `sim.gender.partial.data`. The function simulates partial ID capture-recapture data coming from two detectors.
3. ***partialID.sim.da.R***: This R code runs MCMC to fit the partial ID model to a simulated capture-recapture data (obtained by `sim.gender.partial.data`) of Dey et al (2017) and to get posterior estimates of different parameters. This code needs *partialID.simdata.R*, *utility.functions.R* to run as a preamble.
4. ***partialID.da.R***: This R code contains a function `PISCR.fn`. It is the main function that runs MCMC to fit the partial ID model of Dey et al (2017) to a given data set. It obtains posterior estimates of different parameters and saves as files. This code needs *utility.functions.R* to run as a preamble.

1.2 STEP 1: SETTING UP THE INPUT FILES FOR ANALYSIS

`PISCR.fn` requires five types of data input:

1. Animal Capture Data coming from detector 1 (Session, Animal ID, trap ID, capture occasion)

2. Animal Capture Data coming from detector 2 (Session, Animal ID, trap ID, capture occasion)
3. Data on genders of the captured individuals (Animal ID, Gender)
4. Trap Deployment Data (Trap ID, location and deployment record)
5. State-space Data (the Potential Animal Home Range Center locations and habitat suitability indicator for these home range centers)

The first four data inputs are matrices. The fifth data input is shape file of the state space (should be in UTM units).

INPUT DATA 1 and 2: Animal Capture Details from detector 1 and 2 (viz. capture_1, capture_2)

Both of these capture files are matrix with 4 columns: Session, Location Number, the Animal Identity Number and the Sampling Occasion number, in that order. Each of the two matrices take numeric integer entries. Each captured individual captured should be given a unique identification number, ranging from 1 to n , where n is the total number of individuals included in this file. n can vary from one detector to another. Note that, first IDfixed number of row entries in each of the two matrices should be same where ‘IDfixed’ is the number of captured individuals who are completely identified using the two detectors. In the example below, the records have been sorted by ANIMAL_ID to facilitate checking that no numbers are missing. Location Numbers must correspond to the numbers in the Trap Deployment file. If only 7 and 6 animals were photo-captured and identified by detector 1 and 2, of which only the first 5 are fully identified (across the detectors), the INPUT DATA 1 and 2 might look like this: Note

Session	ANIMAL_ID	SO	LOC_ID	Session	ANIMAL_ID	SO	LOC_ID
1	1	14	11	1	1	14	11
1	2	12	15	1	2	12	15
1	3	20	9	1	3	20	9
1	4	17	1	1	4	17	1
1	5	13	5	1	5	13	5
1	6L	17	7	1	6R	10	14
1	7L	9	6				

the naming of the sixth individual captured by both the detectors viz. 6L, 6R; this means that these two animals might not be the same and are partially identified. For example, the first row of data tells us that Animal ID no 1 was captured at Location ID 11 on the 14th sampling occasion. The first row of the file must have the exact column headings as shown above, or it will not be recognized by PISCR.fn.

INPUT DATA 3: Gender information details (viz. gender)

This file has four columns: ANIMAL_ID_1, Genders_1, ANIMAL_ID_2, Genders_2. These animal ids should be the same as given in the capture matrices. The Genders should be specified as Male or Female. An example of such a matrix is given below :

ANIMAL_ID_1	Genders_1	ANIMAL_ID_2	Genders_2
1	Male	1	Male
2	Female	2	Female
3	Female	3	Female
4	Male	4	Male
5	Female	5	Female
6L	Male	6R	Female
7L	Female		

INPUT DATA 4: Trap Deployment Details (viz. traploc)

The first column of this file has the Location Number. The next two columns have the X and Y coordinates of the trap location: see the example below. Use a projected coordinate system (ie, in meters, not degrees), such as UTM (Universal Transverse Mercator) or your national grid system. After these first three columns, there is one column for each trapping occasion. Here, "1" means that a trap was operating at that location on that occasion, "0" means that no trap was operating. You do not have to have a trap at every location for the whole period. In particular, trap malfunction, theft, vandalism, etc. can be accounted for. An example trap deployment data is shown below. This file has 12 locations, but there were only 4 traps, which were moved between locations. The table shows that the first four locations were operational during sampling occasions 1-5, except that the trap at location 3 was not working on occasion 4. No traps were operational at these sites on occasions 6 to 15. Again, the first row must have column headings as shown in the example above.

LOC_ID	X_Coord	Y_Coord	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	619303	1325966	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	624151	1325013	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
3	624722	1323864	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
4	621806	1322453	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
5	622451	1320137	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
6	622599	1317937	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
7	623179	1315941	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	625156	1315587	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
9	626022	1314224	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
10	627568	1315494	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
11	619604	1324739	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
12	621478	1324515	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

INPUT DATA 5: State-space Data (viz. activity)

These is a shape file indicating a state space in UTM coordinates. It comprises of the potential animal activity center locations. There should be a total of 5 files with extensions ‘.dbf’, ‘.sbn’, ‘.sbx’, ‘.shp’, ‘.shx’. But all the files should have the same name. An example of such a list of files are as follows : ‘shape.dbf’, ‘shape.sbn’, ‘shape.sbx’, ‘shape.shp’, ‘shape.shx’. An R command to load and store such a data file to a variable in an R session can be :

```
activity <- readShapeSpatial("shape",
proj4string=CRS("+proj=utm +zone=43 +datum=WGS84")).
```

1.3 STEP 2: SETTING THE MARKOV-CHAIN MONTE CARLO (MCMC) PARAMETERS

Argument	Definition
ndraws	Number of MCMC iterations
burnin	Burn-in period of MCMC iterations
M	Maximum number of individuals within the state space.
scale	Scaling factor for the UTM coordinates.
nloopL	Number of iterations in optimizing the initial value of L vector.
n.update	Number of MCMC updates for L vector.
batchsize	The size of a batch to update proposal variance.
mindelta	Reducing factor of proposal variance.
sigmaOfProposal.logitphi	Proposal variance of $\text{logit}(\phi)$.
sigmaOfProposal.logitp0	Proposal variance of $\text{logit}(p_0)$.
sigmaOfProposal.logsigmam	Proposal variance of $\log(\sigma_m)$.
sigmaOfProposal.logsigmaf	Proposal variance of $\log(\sigma_m)$.
sigmaOfProposal.L	Proposal variance of required to update L vector.
sigmaOfProposal.s	Proposal variance of \mathbf{S} . It is a vector of length M .
R	Maximum permissible value of movement range for each individual during the study.

Example

An example R code to run the function `PISCR.fn` is given below :

```
# Loading the required libraries

library(raster)
library(maptools)
library(spatstat)
library(gtools)

# Reading the data files

capture_1 = read.csv(file="crdata_1.csv", header=TRUE) # Capture data from detector 1
```

```

capture_2 = read.csv(file="crdata_2.csv", header=TRUE) # Capture data from detector 2
traploc = read.csv(file="trap.csv", header=TRUE) # Trap deployment data
gender = read.csv(file="NH15_gender.csv", header = T) # Data on gender
activity = readShapeSpatial("shape",
proj4string=CRS("+proj=utm +zone=43 +datum=WGS84")) # Shape file of the state space

# Calling the R code to source the function PISCR.fn
source("partialID.da.R")

PISCR.fn ( capture_1, capture_2, gender, activity, traploc,
ndraws = 10000, burnin = 5000, M = 400, scale = 10^4,
nloopL = 50, n.update = 20, batchsize = 1000, mindelta = 0.01,
sigmaOfProposal.logitphi = 0.05, sigmaOfProposal.logitp0 = 0.08,
sigmaOfProposal.logsigmam = 0.02, sigmaOfProposal.logsigmaf = 0.02,
sigmaOfProposal.L = 4, sigmaOfProposal.s = rep(3, 400), R = 5)

```