Introduction to spatsoc

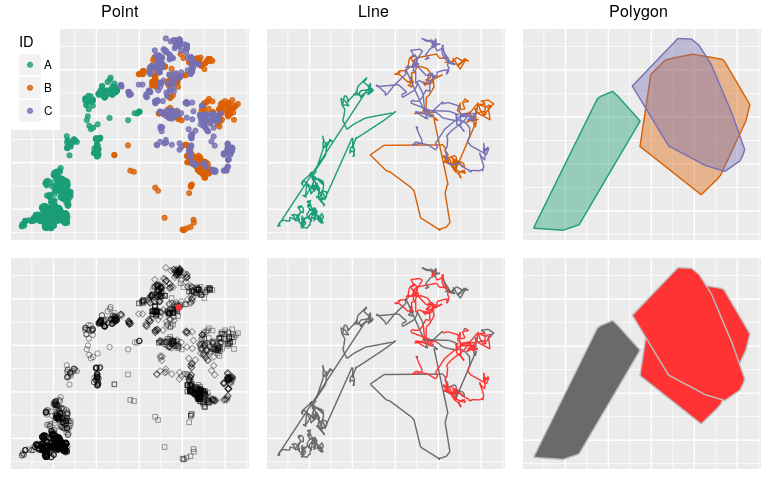
Alec Robitaille

2017-10-24

The spatsoc package provides functionality for analyzing animal relocation data in time and space to identify potential interactions among individuals or social groups.

The package contains grouping functions that are used for identifying spatially and temporally explicit groups from input data. In addition, the package contains functions for determining the nearest neighbor and mean pairwise distance between individuals, optionally across time steps. Finally, we provide a function for randomizing individual identifiers within groups, designed to test whether social networks generated from animal relocation data were based on non-random social proximity among individuals.

The functions were developped for application across animal relocation data, for example, proximity based social network analyses and spatial and temporal clustering of points.



## Grouping

Let's read in spatsoc's example data.

library(spatsoc); library(data.table)  
data(locs)  
locs

|  |  |  |  |
| --- | --- | --- | --- |
| ID | EASTING | NORTHING | datetime |
| C | 683147 | 5309527 | 2011-02-10 20:00:49 |
| A | 651526 | 5298946 | 2012-02-04 14:00:40 |
| B | 685874 | 5323236 | 2012-02-18 12:00:41 |
| B | 679010 | 5326294 | 2010-02-14 00:00:48 |
| A | 649727 | 5298351 | 2012-01-31 08:00:12 |

The data consist of relocations of 3 individuals over 31 days. Relocations were recorded using GPS collars with fixes obtained every two hours. Using these data, we can compare the various grouping methods available in spatsoc.

### GroupPts

The GroupPts function compares the relocations of all individuals at each time step and groups individuals based on a buffer distance provided by the user.

Since GPS fixes taken at regular intervals have some level of variability, we will provide a time threshold (timeThreshold), to consider all fixes within this threshold taken at the same time. For example, we will use a 5 minute threshold.

GroupPts(locs, 50, 'datetime', '5 minutes',   
 '+proj=utm +zone=21 ellps=WGS84',   
 c('EASTING', 'NORTHING'), 'ID')

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| timeGroup | EASTING | NORTHING | ID | group |
| 130 | 652022 | 5306566 | A | 130\_2 |
| 130 | 679214 | 5328150 | B | 130\_1 |
| 130 | 679203 | 5328159 | C | 130\_1 |
| 131 | 651843 | 5306425 | A | 131\_3 |
| 131 | 679470 | 5328345 | B | 131\_1 |
| 131 | 679280 | 5328337 | C | 131\_2 |
| 132 | 651391 | 5306861 | A | 132\_3 |
| 132 | 678768 | 5328675 | B | 132\_1 |

### GroupLines

The GroupLines function groups individuals whose trajectories intersect in a specified time interval (eg: daily). This represents a coarser grouping method than GroupPts which can help understand shared space at daily, weekly or other temporal resolutions.

GroupLines(locs, bufferWidth = 50, timeField = 'datetime', timeThreshold = '1 day', projection = '+proj=utm +zone=21 ellps=WGS84', coordFields = c('EASTING', 'NORTHING'),   
 idField = 'ID')

|  |  |  |
| --- | --- | --- |
| timeGroup | ID | group |
| 37 | B | 1\_20 |
| 37 | C | 1\_20 |
| 38 | A | 2\_21 |
| 38 | B | 1\_21 |
| 38 | C | 1\_21 |
| 39 | A | 2\_22 |
| 39 | B | 1\_22 |

### GroupPolys

The GroupPolys function groups individuals whose home ranges intersect. This represents the coarsest grouping method, to provide a measure of overlap across seasons, years or all available relocations. It can either return the proportion of home range area overlapping between individuals or simple groups. Home ranges are calculated using adehabitatHR::kernelUD or adehabitatHR::mcp. Alternatively, a SpatialPolygonsDataFrame can be input to the spPolys argument.

GroupPolys(area = TRUE, hrType = 'mcp',   
hrParams = list('percent' = 95),   
locs, projection = '+proj=utm +zone=21 ellps=WGS84',  
coordFields = c('EASTING', 'NORTHING'), idField = 'ID')

|  |  |  |
| --- | --- | --- |
| area | ID1 | ID2 |
| 3.06e+08 | A | A |
| 5.24e+08 | B | B |
| 360422056 | B | C |
| 360422056 | C | B |
| 425295792 | C | C |

## Notes

Package dependencies for spatsoc are sp, rgeos, and data.table. data.table provides efficient methods for manipulating large (or small) datasets. As a result, input DT for all spatsoc functions must be a data.table and if it isn't, you can simply use setDT(df) to convert it by reference.

In addition, since the rgeos package is used in most functions (notably GroupPts and GroupLines) the input DT's coordinate system is important. rgeos expects planar coordinates and this requirement is carried forward for spatsoc.