# Package 'moveNT'

July 3, 2018

**Title** An R package for the analysis of movement data using network theory

**Version** 0.0.0.9000

**Description**
This package provides a series of functions to analyse movement data using network theory.

**Depends** R (>= 3.3.2), raster, sp, adehabitatLT

**License** GPL (>=3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1.9000

**Imports** igraph, mclust, moveHMM, uuid

## R topics documented:



---

`adj2stack` *Calculation of network metrics*

---

### Description

Transform an adjancency matrix to a series of network metrics at the node-level (weight, degree, betweenness, transitivity, eccenctricity) and graph level (diameter, transitivity, density, and modularity)

### Usage

```
adj2stack(adjmov, grph = T, mode = "directed", weighted = T, ...)
```

### Arguments

| | |
|---|---|
| adjmov | Adjacency matrix, need to be an object produced by function traj2adj |
| grph | Whether node level metrics are to be plotted |
| mode | Whether the graph should be "directed" or "undirected. Default="directed". See "graph_from_adjacency_matrix" from package "igraph" |
| weighted | Whether the graph should be weighted (=TRUE) or unweighted (= NULL). Default is weighted. See "graph_from_adjacency_matrix" from package "igraph" |

### Value

A raster stack object

### Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
stck<-adj2stack(traj2adj(traj1, res=100), grph=T)
```

---

## clustnet *Normal mixture model for clustering of node level metrics*

---

### Description

Apply a normal mixture model to a node-level metric

### Usage

```
clustnet(stack, id = 2, nclust = 2, grph = T)
```

### Arguments

| | |
|---|---|
| stack | An object produce by the function adj2stack |
| id | Metric to be used (2=Weight, 3=Degree, 4=Betweenness, 5=Transitivity, 6=Eccentricity) |
| grph | Whether resulting classification should be plotted |

### Value

A list object containing a Mclust object and a raster object

### Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
stck<-adj2stack(traj2adj(traj1, res=100), grph=T)
cl<-clustnet(stck, id=2, nclust=2, grph=T)
summary(cl[[1]])
```

## dot *dot product*

### Description

dot product

### Usage

```
dot(x, ...)
```

## graphmet *Summarize graph-level metrics*

### Description

Summarize graph-level metrics from an object generated by adj2stack

### Usage

```
graphmet(grid)
```

### Arguments

| | |
|---|---|
| grid | An object generated by the function adj2stack |
| id | Metric to be used (2=Weight, 3=Degree, 4=Betweenness, 5=Transitivity, 6=Eccentricity) |

### Value

A vector

### Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
stck<-adj2stack(traj2adj(traj1, res=quant(traj1)), grph=T)
graphmet(stck)
```

`interpolation` *Interpolation based on movement steps for all individuals*

## Description

Use movement steps to interpolate raster. User can extract the mean or max when multiple steps overlap in a single pixel. Function need to be applied followign the loop function. This process is very slow

## Usage

```
interpolation(traj, ls, wei = mean, deg = mean, bet = max,
  spe = mean, dt = dot)
```

## Arguments

| | |
|---|---|
| `traj` | An object produce by the function adj2stack |
| `ls` | An object produced by the loop |
| `wei` | Whether mean or max should be used for weight (default = mean) |
| `deg` | Whether mean or max should be used for degree (default = mean) |
| `bet` | Whether mean or max should be used for betweeness (default = max) |
| `spe` | Whether mean or max should be used for speed (default = mean) |
| `dt` | Whether mean, max, or dot produc should be used for turning angle (default = dot) |

## Value

A list object containing a raster stack object for each individual

## Examples

```
data(puechabonsp)
locs <- puechabonsp$relocs
xy <- coordinates(locs)
df <- as.data.frame(locs)
da <- as.character(df$Date)
da <- as.POSIXct(strptime(as.character(df$Date),"%y%m%d", tz="Europe/Paris"))
litr <- as.ltraj(xy, da, id = id)
out1<-loop(litr)
out2<-interpolation(litr, out1)
```

## loop *Looping over all individuals*

### Description

Extract adjancency matrix and calculate network metrics for all individuals in a trajectory object. Also calculate mean speed, mean direction, and dot product.

### Usage

```
loop(traj, res = 100)
```

### Arguments

| | |
|---|---|
| traj | An object produce by the function adj2stack |
| res | Grid size |

### Value

A list object containing a raster stack object for each individual

### Examples

```
data(puechabonsp)
locs <- puechabonsp$relocs
xy <- coordinates(locs)
df <- as.data.frame(locs)
da <- as.character(df$Date)
da <- as.POSIXct(strptime(as.character(df$Date),"%y%m%d", tz="Europe/Paris"))
litr <- as.ltraj(xy, da, id = id)
out1<-loop(litr)
```

## mosaic_network *Mosaic individuals together for a given variable*

### Description

Use output of loop or interpolation and combine all individuals (mosaic) together using the mean or max values.

### Usage

```
mosaic_network(ls, index = 2, sc = T, fun = mean)
```

### Arguments

| | |
|---|---|
| ls | An object produced by the loop or interpolate functions |
| index | Index indicating which layer to take in the stack |
| sc | Whether to scale all individual rasters (default = TRUE) |
| fun | Whether mean or max should be used as the mosaic function (default = mean) |

## Value

A list object containing a raster stack object for each individual

## Examples

```
data(puechabonsp)
locs <- puechabonsp$relocs
xy <- coordinates(locs)
df <- as.data.frame(locs)
da <- as.character(df$Date)
da <- as.POSIXct(strptime(as.character(df$Date),"%y%m%d", tz="Europe/Paris"))
litr <- as.ltraj(xy, da, id = id)
out1<-loop(litr)
mean_weight<-mosaic_network(out1, index=2, sc=T, fun=mean) #Perform mean weight (not-interpolated)
plot(mean_weight)
```

---

# quant *Sample quantile of distance for ltraj object*

---

## Description

Wrapper function that extract the sample quantile of distance

## Usage

```
quant(x, p = 0.5)
```

## Arguments

| | |
|---|---|
| x | A ltraj object |
| p | Probability, default=0.5 (median) |

## Value

A vector of length p

## Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
stck<-adj2stack(traj2adj(traj1, res=quant(traj1)), grph=T)
```

sim_mov *Simulation of patch-based movement trajectory*

## Description

Simulate a movement trajectory with user defined number of patches and interpatch movement

## Usage

```
sim_mov(type = c("2states", "OU"), npatches = 5, ratio = 5,
  nswitch = 150, ncore = 200, spacecore = 200,
  seq_visit = sample(1:npatches, nswitch, replace = T),
  stepDist = "gamma", angleDist = "vm", stepPar = c(0.5, 3, 1, 5),
  anglePar = c(pi, 0, 0.5, 2), s = diag(40, 2), grph = F)
```

## Arguments

| | |
|---|---|
| type | whether movement within patches should be based on a 2states process (from package moveHMM) or a Bivariate Ornstein-Uhlenbeck process (OU) (from package adehabitatLT) |
| npatches | Number of patches, default=5 |
| ratio | Ratio (in percent) of locations associated to interpatch movement, default=5 |
| nswitch | Number of switch/depart from patches, default=150 |
| ncore | Number of locations within a patch per visit, default=200 |
| spacecore | Minimum distance between center of patches, default=200 |
| seq_visit | Specify the sequence of visit among patches, default is random sequence |
| stepDist | Distribution for step length if 2states specified in type, see simData of moveHMM package |
| angleDist | Distribution for turn angle if 2states specified in type, see simData of moveHMM package |
| stepPar | Parameters for step length distribution if 2states specified in type, see simData of moveHMM package |
| anglePar | Parameters for turn angle distribution if 2states specified in type, see simData of moveHMM package |
| s | Parameters for the OU process, see simm.mou of adehabitatLT package |
| grph | Whether a graph of the trajectory should be produced, default=F |

## Value

A ltraj (adehabitatLT) object

## Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
traj2<-sim_mov(type="2states", npatches=2, grph=T)
```

## traj2adj *Generation of adjacency matrix from movement data*

### Description

Transform an ltraj object to an adjacency matrix using a user-specified grid size

### Usage

```
traj2adj(mov, res = 100, grid = NULL)
```

### Arguments

| | |
|---|---|
| mov | Movement trajectory, need to be a ltraj object |
| res | Grid size |
| grid | User specified grid (a raster), needs to have a larger extent than the movement trajectory |

### Value

A list of object containing the adjacency matrix, the grid use, and patch/corridor identification (only useful if sim_mov was used)

### Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
adj<-traj2adj(traj1, res=100)
```

## val *Extract occupied cells in a raster object*

### Description

Extract only occupied cells in a raster object,

### Usage

```
val(grid, id)
```

### Arguments

| | |
|---|---|
| grid | An object generated by the function adj2stack |
| id | Metric to be used (2=Weight, 3=Degree, 4=Betweenness, 5=Transitivity, 6=Eccentricity) |

### Value

A vector

## Examples

```
traj1<-sim_mov(type="OU", npatches=3, grph=T)
stck<-adj2stack(traj2adj(traj1, res=quant(traj1)), grph=T)
mean(val(stck, 2))
```

# Index