

# Package ‘moveNT’

February 28, 2020

**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

**Author** Guillaume Bastille-Rousseau [aut, cre]

**Maintainer** Guillaume Bastille-Rousseau <gbr@colostate.edu>

**RemoteType** github

**RemoteHost** https://api.github.com

**RemoteRepo** moveNT

**RemoteUsername** BastilleRousseau

**RemoteRef** master

**RemoteSha** a3fe40032d9a0db589b31a34090820fd75d0d870

**GithubRepo** moveNT

**GithubUsername** BastilleRousseau

**GithubRef** master

**GithubSHA1** a3fe40032d9a0db589b31a34090820fd75d0d870

## R topics documented:

adj2stack . . . . .	2
clustnet . . . . .	3
clust_stack . . . . .	3
dot . . . . .	4
graphmet . . . . .	4
ind_clust . . . . .	5
interpolation . . . . .	5

loop . . . . .	6
mosaic_network . . . . .	7
pop_clust . . . . .	8
pop_overl . . . . .	8
pop_stack . . . . .	9
quant . . . . .	10
sim_mov . . . . .	10
table_cluster . . . . .	11
traj2adj . . . . .	12
val . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

adj2stack	<i>Calculation of network metrics</i>
-----------	---------------------------------------

---

## Description

Transform an adjacency matrix to a series of network metrics at the node-level (weight, degree, betweenness, transitivity, eccentricity) 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	<i>Normal mixture model for clustering of single node level metric</i>
----------	--

---

**Description**

Apply a normal mixture model to a single node-level metric

**Usage**

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

**Arguments**

stack	An object produce by the function adj2stack (not compatible with loop or interpolation)
id	Metric to be used (2=Weight, 3=Degree, 4=Betweenness, 5=Transitivity, 6=Ec-centricity)
grph	Whether resulting classification should be plotted

**Value**

A list of 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]])
```

---

clust_stack	<i>Back-association of population-level clustering to individual clusters</i>
-------------	---

---

**Description**

Generate individual-level rasters of population-level clustering. For each individual, the fuction generate a raster stack containing a raster of the most likely cluster, and several rasters giving the probability of observing each cluster.

**Usage**

```
clust_stack(grid, pop_clust, ind_clust, table)
```

**Arguments**

grid	The output of the loop function
pop_clust	The output of the pop_clust function
ind_clust	The output of the ind_clust function
table	The output of table_cluster

**Value**

A list of raster stack object.

**Examples**

```
data(albatross)
grid<-loop(albatross, 35000)
table_grid<-table_cluster(albatross, grid)
ls_ind<-ind_clust(table_grid, max.n.clust=8)
pop<-pop_clust(albatross, ls_ind)
clust_stack<-clust_stack(grid, pop, ls_ind, table_grid)
plot(clust_stack[[1]])
```

---

dot	<i>dot product</i>
-----	--------------------

---

**Description**

dot product

**Usage**

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

---

graphmet	<i>Summarize graph-level metrics</i>
----------	--------------------------------------

---

**Description**

Summarize graph-level metrics from an object generated by adj2stack

**Usage**

```
graphmet(grid)
```

**Arguments**

grid	An object generated by the function adj2stack
------	---

**Value**

A vector

**Examples**

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

---

ind_clust	<i>Individual-level clustering of movement metrics</i>
-----------	--

---

### Description

Perform individual-level clustering (first step) of movement metrics. This function uses the output of table\_cluster and perform a mixture-model. Users can select which variables will be used and the maximum number of clusters. See also mclust

### Usage

```
ind_clust(table, max.n.clust = 8, modelname = "EEV",
  vars = c("Weight", "Degree", "Betweenness", "Speed", "DotP"))
```

### Arguments

table	An output from the table_cluster function
max.n.clust	The maximum number of clusters to test, see the documentation for mclust for more information. Default = 8.
modelname	The model structure of the clustering, see the documentation for mclust for more information. Default is equal mean and variance for each clusters (EEV).
vars	The variable to be included. Default = c("Weight", "Degree", "Betweenness", "Speed", "DotP")

### Value

A list object with each element representing an individual.

### Examples

```
data(albatross)
grid<-loop(albatross, 35000)
table_grid<-table_cluster(albatross, grid)
ls_ind<-ind_clust(table_grid, max.n.clust=8)
table(unlist(lapply(ls_ind, function(x) x$G)))
```

---

interpolation	<i>Interpolation based on movement steps for all individuals</i>
---------------	--

---

### Description

Use movement steps to linearly interpolate raster produced by loop. User can select if the mean or max is taken when multiple steps overlap in a single pixel. Function need to be applied following 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 betweenness (default = max)
spe	Whether mean or max should be used for speed (default = mean)
dt	Whether mean, max, or dot product should be used for turning angle (default = dot)

**Value**

A list of 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	<i>Looping over all individuals</i>
------	-------------------------------------

---

**Description**

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

**Usage**

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

**Arguments**

traj	An object produce by the function adehabitatLT with multiple individuals
res	Grid size, will be apply to all individuals

**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 (combine) individual raster 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 raster layer object.

**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)

```

---

pop_clust	<i>Population-level clustering of movement metrics</i>
-----------	--

---

### Description

Combine individual-level clustering of movement metrics into a population-level clustering (second step). Users can the maximum number of clusters. See also mclust

### Usage

```
pop_clust(traj, ls, max.n.clust = 8)
```

### Arguments

traj	The trajectory object
max.n.clust	The maximum number of clusters to test, see the documentation for mclust for more information. Default = 8.
ls_ind	Individual-level clustering object, the output of ind_clust.

### Value

A list object with each element representing an individual.

### Examples

```
data(albatross)
grid<-loop(albatross, 35000)
table_grid<-table_cluster(albatross, grid)
ls_ind<-ind_clust(table_grid, max.n.clust=8)
pop<-pop_clust(albatross, ls_ind)
pop[[1]]$parameters$mean
pop[[1]]$parameters$pro
```

---

pop_overl	<i>Population-level multi-use map</i>
-----------	---------------------------------------

---

### Description

Produce a single map (raster) indicating all types of use found in a cluster.

### Usage

```
pop_overl(clust_stack)
```

### Arguments

clust_stack	The output of clust_stack
-------------	---------------------------

### Value

A stack object with each raster showing use of each cluster



**Examples**

```

data(albatross)
grid<-loop(albatross, 35000)
table_grid<-table_cluster(albatross, grid)
ls_ind<-ind_clust(table_grid, max.n.clust=8)
pop<-pop_clust(albatross, ls_ind)
clust_stack<-clust_stack(grid, pop, ls_ind, table_grid)
pop_overl<-pop_overl(clust_stack)
table(values(pop_overl))
plot(pop_overl)

```

---

pop_stack	<i>Population-level single-use maps of each cluster</i>
-----------	---

---

**Description**

Produce maps (raster) indicating if at least one individual is using a pixel as a given cluster

**Usage**

```
pop_stack(clust_stack)
```

**Arguments**

clust\_stack      The output of clust\_stack

**Value**

A stack object with each raster showing use of each cluster

**Examples**

```

data(albatross)
grid<-loop(albatross, 35000)
table_grid<-table_cluster(albatross, grid)
ls_ind<-ind_clust(table_grid, max.n.clust=8)
pop<-pop_clust(albatross, ls_ind)
clust_stack<-clust_stack(grid, pop, ls_ind, table_grid)
pop_stack<-pop_stack(clust_stack)
plot(pop_stack)

```

---

quant	<i>Sample quantile of distance for ltraj object</i>
-------	---

---

### Description

Wrapper function that extract the sample quantile of distance of a trajectory object

### 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	<i>Simulation of patch-based movement trajectory</i>
---------	--

---

### Description

Simulate a movement trajectory with a 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)
```

---

table_cluster	<i>Convert a list of adj2stack object to a data.frame for clustering</i>
---------------	--

---

**Description**

Convert output of loop function to a data.frame.

**Usage**

```
table_cluster(traj, grid)
```

**Arguments**

traj	The trajectory used in loop (a traj object)
grid	The output of the loop function

**Value**

A data.frame object.

**Examples**

```
data(albatross)
grid<-loop(albatross, 35000)
table_grid<-table_cluster(albatross, grid)
head(table_grid)
```

---

traj2adj	<i>Generation of adjacency matrix from movement data</i>
----------	--

---

### 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 (based on coordinate system of movement trajectory)
grid	User specified grid (a raster), needs to have a larger extent than the movement trajectory

### Value

A list of objects 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	<i>Extract occupied cells in a raster object</i>
-----	--

---

### 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=Ec-centricity)

### 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

- \*Topic **Mclust**
  - clustnet, 3
- \*Topic **adj2stack**
  - clust\_stack, 3
  - clustnet, 3
  - graphmet, 4
  - ind\_clust, 5
  - interpolation, 5
  - loop, 6
  - mosaic\_network, 7
  - pop\_clust, 8
  - pop\_overl, 8
  - pop\_stack, 9
  - sim\_mov, 10
  - table\_cluster, 11
  - traj2adj, 12
  - val, 12
- \*Topic **ind\_clust**
  - clust\_stack, 3
  - pop\_clust, 8
  - pop\_overl, 8
  - pop\_stack, 9
- \*Topic **interpolation**
  - mosaic\_network, 7
- \*Topic **loop**
  - clust\_stack, 3
  - ind\_clust, 5
  - interpolation, 5
  - mosaic\_network, 7
  - pop\_clust, 8
  - pop\_overl, 8
  - pop\_stack, 9
  - table\_cluster, 11
- \*Topic **ltraj**
  - quant, 10
- \*Topic **pop\_clust**
  - clust\_stack, 3
  - ind\_clust, 5
- \*Topic **table\_cluster**
  - clust\_stack, 3
  - ind\_clust, 5
  - pop\_clust, 8
  - pop\_overl, 8
  - pop\_stack, 9
- \*Topic **traj2adj**
  - adj2stack, 2
  - clust\_stack, 3
  - clustnet, 3
  - ind\_clust, 5
  - interpolation, 5
  - loop, 6
  - mosaic\_network, 7
  - pop\_clust, 8
  - pop\_overl, 8
  - pop\_stack, 9
  - sim\_mov, 10
  - table\_cluster, 11
- adj2stack, 2
- clust\_stack, 3
- clustnet, 3
- dot, 4
- graphmet, 4
- ind\_clust, 5
- interpolation, 5
- loop, 6
- mosaic\_network, 7
- pop\_clust, 8
- pop\_overl, 8
- pop\_stack, 9
- quant, 10
- sim\_mov, 10
- table\_cluster, 11
- traj2adj, 12
- val, 12