

Package: ScaleInMultilayerNetworks (via r-universe)

October 21, 2024

Title Package Accompanying: The Problem And Promise Of Scale In Multilayer Animal Social Networks.

Version 0.1.1

Description Scale remains a foundational concept in ecology. Spatial scale, for instance, has become a central consideration in the way we understand landscape ecology and animal space use. Meanwhile, scale-dependent social processes can range from fine-scale interactions to co-occurrence and overlapping home ranges. Furthermore, sociality can vary within and across seasons. Multilayer networks promise the explicit integration of the social, spatial and, temporal contexts. Given the complex interplay of sociality and animal space use in heterogeneous landscapes, there remains an important gap in our understanding of the influence of scale on animal social networks. Using an empirical case study, we discuss ways of considering social, spatial and, temporal scale in the context of multilayer caribou social networks. Effective integration of social and spatial processes, including biologically meaningful scales, within the context of animal social networks is an emerging area of research. We incorporate perspectives that link the social environment to spatial processes across scales in a multilayer context.

License GPL-3 | file LICENSE

Encoding UTF-8

LazyData true

Imports data.table, spatso, igraph, asnipe

RoxxygenNote 7.1.1

Repository <https://robital.ec.r-universe.dev>

RemoteUrl <https://github.com/robital.ec/ScaleInMultilayerNetworks>

RemoteRef HEAD

RemoteSha 3f0bd569874f0fb4aa92018a014640ea6f9b4f87

Contents

edge_overlap	2
edge_overlap_mat	2
layer_neighbors	3
layer_relevance	4
layer_strength	4
list_edges	5
list_gbi	5
list_graphs	6
list_nets	6
multi_degree	7
property_matrix	7
shear_xy	8

Index	9
--------------	----------

edge_overlap	<i>Edge overlap</i>
--------------	---------------------

Description

Edge overlap

Usage

edge_overlap(edges)

Arguments

graphLs

edge_overlap_mat	<i>Edge overlap matrix</i>
------------------	----------------------------

Description

Layer A vs Layer B, count overlap

Usage

edge_overlap_mat(edges)

Arguments

edges

layer_neighbors	<i>Neighbourhood</i>
-----------------	----------------------

Description

Number of neighbors adjacent to each actor. Calculated excluding self from set of neighbors.

Usage

```
layer_neighbors(DT, id, splitBy = NULL)
```

Arguments

DT	a data.table with column "group" generated by spatsoc::group_pts
id	
splitBy	the column which defines the layers of the network

Value

The input DT with additional column "neigh" and optionally "splitNeigh" if a column was provided for the 'splitBy' argument.

Examples

```
# Load data.table and spatsoc
library(data.table)
library(spatsoc)

# Read example data
DT <- fread(system.file("extdata", "DT.csv", package = "spatsoc"))

# Cast the character column to POSIXct
DT[, datetime := as.POSIXct(datetime, tz = 'UTC')]

# Temporal grouping
group_times(DT, datetime = 'datetime', threshold = '20 minutes')

# Spatial grouping with timegroup
group_pts(DT, threshold = 5, id = 'ID',
          coords = c('X', 'Y'), timegroup = 'timegroup')

# Pseudo-season
DT[, season := sample(c(1, 2), .N, replace = TRUE)]

layer_neighbors(DT, 'ID', splitBy = 'season')
```

layer_relevance	<i>Relevance</i>
-----------------	------------------

Description

Proportion of neighbours present on each layer.

Usage

```
layer_relevance(DT, id, splitBy)
```

Arguments

DT
id
splitBy
var

References

Berlingerio, Michele, et al. "Foundations of multidimensional network analysis." 2011 international conference on advances in social networks analysis and mining. IEEE, 2011.

layer_strength	<i>Calculate graph strength for each graph in a list</i>
----------------	--

Description

Calculate graph strength for each graph in a list

Usage

```
layer_strength(graphLs)
```

Arguments

graphLs

<code>list_edges</code>	<i>Edge lists</i>
-------------------------	-------------------

Description

Edge lists

Usage

```
list_edges(graphLs)
```

Arguments

edgeLs

<code>list_gbi</code>	<i>GBI</i>
-----------------------	------------

Description

GBI

Usage

```
list_gbi(DT, id, splitBy, group = "group")
```

Arguments

DT

id

splitBy

group

splitList

`list_graphs`*Graphs*

Description

Graphs

Usage

```
list_graphs(netLs, mode = "undirected", diag = FALSE, weighted = TRUE)
```

Arguments`netLs``mode``diag``weighted`

`list_nets`*Networks*

Description

Networks

Usage

```
list_nets(gbiLs, format = "GBI", ai = "SRI")
```

Arguments`gbiLs``format``ai`

multi_degree	<i>Multidegree</i>
--------------	--------------------

Description

Multidegree

Usage

```
multi_degree(DT, degree, id, splitBy)
```

Arguments

DT

degree

id

Value

Column added named multideg

property_matrix	<i>Property Matrix</i>
-----------------	------------------------

Description

Property Matrix

Usage

```
property_matrix(DT, id, metric, by, layer = "layer")
```

Arguments

DT

id

metric

by

layer

References

Bródka P, Chmiel A, Magnani M, Ragozini G. 2018 Quantifying layer similarity in multiplex networks: a systematic study. R.Soc.opensci. 5:171747. <http://dx.doi.org/10.1098/rsos.171747>

shear_xy *Shear XY for stacked plotting Thanks to
[@rafapereirabr](https://github.com/rafapereirabr) for this gist
(https://gist.github.com/rafapereirabr/97a7c92d40f91cd20a10e8e0165a0aef)
and Barry Rowlingson for the original SO answer
(http://gis.stackexchange.com/questions/189490/plot-tilted-map-
in-r)*

Description

Shear XY for stacked plotting Thanks to [@rafapereirabr](https://github.com/rafapereirabr) for this gist (https://gist.github.com/rafapereirabr/97a7c92d40f91cd20a10e8e0165a0aef) and Barry Rowlingson for the original SO answer (http://gis.stackexchange.com/questions/189490/plot-tilted-map-in-r)

Usage

```
shear_xy(DT, coordcols, shearmatrix = matrix(c(2, 1.2, 0, 1), 2, 2))
```

Arguments

DT
coordcols length 2
shearmatrix

Index

edge_overlap, [2](#)
edge_overlap_mat, [2](#)

layer_neighbors, [3](#)
layer_relevance, [4](#)
layer_strength, [4](#)
list_edges, [5](#)
list_gbi, [5](#)
list_graphs, [6](#)
list_nets, [6](#)

multi_degree, [7](#)

property_matrix, [7](#)

shear_xy, [8](#)