

Package ‘SpNetPrep’

August 29, 2018

Title Linear Network Preprocessing for Spatial Statistics

Version 1.1

Author Alvaro Briz-Redon

Maintainer Alvaro Briz-Redon <alvaro.briz@uv.es>

Description Launches a Shiny application that allows users to carry out some of the steps that are required to curate both a linear network object based on a road structure and a point pattern that lies on such a network, becoming two previous steps to the performance of a spatial statistics analysis.

Depends R (>= 3.5.0)

License GPL-2

Imports spatstat, leaflet, shiny, sp, raster, maptools, rgdal,
shinythemes, prodlim

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

NeedsCompilation no

R topics documented:

runAppSpNetPrep	1
SampleDirectedNetwork	2
SampleNetwork	3
SamplePointPattern	3
SimplifyLinearNetwork	3
Index	5

runAppSpNetPrep	<i>Runs the SpNetPrep Shiny web application.</i>
-----------------	--

Description

Spatial statistics studies have been commonly based on geographic structures made of polygons representing an administrative or political division of different order, depending on the size of the region being analyzed and on the specific interest of the researchers. More specifically, the basic spatial units in these studies have ranged from larger (countries or counties) to smaller (cities, boroughs, census tracks, etc.), allowing the employment of the usually available information regarding these kind of population units. However, last years are bringing a higher number of spatial analysis that are defined over network structures, which allow a better understanding of some spatial point patterns of great interest. Basically, the use of spatial networks has become quite frequent when the events of study actually take place in roads, streets, highways, etc., which oblige to discard most of the areal region of the zone of analysis if an accurate investigation is intended. The SpNetPrep package does not deal with statistics, but with the previous steps that can be required in order to perform a spatial statistical analysis of a point pattern that lies on a linear network representing a road structure. In this regard, the name chosen for the package summarizes its main goal of "Spatial Network Preprocessing" (SpNetPrep). The main feature provided by the SpNetPrep package is an interactive application that allows to carry out the complete preprocessing of a linear network that comes from a road structure. This application launches when the user types the instruction `runAppSpNetPrep()` in the R console (after the installation of the package), and it is also available online in <https://albriz.shinyapps.io/spnetprep/>. If the application is run from the console, it is necessary to click the option "Open in browser" when it shows, or define "Run external" for the opening of Shiny applications in order to be able to download the modifications performed on the objects uploaded to it.

Usage

```
runAppSpNetPrep()
```

SampleDirectedNetwork *Linear network representing a directed road structure*

Description

An `SpatialLinesDataFrame` representing a directed road structure. The `data.frame` attached has three columns, V1, V2 and Dir that determine the direction of the network. The column Dir indicates the type of flow existing between V1 and V2: none (0), from V1 to V2 (1), from V2 to V1 (-1) or bidirectional (2).

Usage

```
SampleDirectedNetwork
```

Format

An object of class `SpatialLinesDataFrame` with 2513 rows and 3 columns.

SampleNetwork	<i>Linear network representing a road structure</i>
---------------	---

Description

An SpatialLines representing a road structure

Usage

SampleNetwork

Format

An object of class SpatialLines of length 2513.

SamplePointPattern	<i>Point pattern on a road network simulating traffic accidents</i>
--------------------	---

Description

An lpp object generated with a Poisson process of intensity 0.01 over the SampleNetwork road structure. The pattern is marked with three categorical variables: accident.date, collision and vehicles, which were also randomly constructed

Usage

SamplePointPattern

Format

An object of class lpp (inherits from ppx) of length 3.

SimplifyLinearNetwork	<i>Simplifies a linear network object by merging the edges that meet at a second-degree vertex if some conditions on the angle they form and on their lengths are satisfied</i>
-----------------------	---

Description

This algorithm attempts to automatically reduce a linear network's complexity without altering its basic geometric configuration. The main objective of the algorithm is to merge the pairs of edges of the network that are connected by a second-degree vertex (with only two incident edges) into only one edge. Equivalently, this action means to join two vertex of the network whose path of connection only passes through another vertex of the network.

Usage

SimplifyLinearNetwork(network, Angle = NULL, Length = NULL, M = NULL)

Arguments

network	- A linnet object representing a linear network structure
Angle	- An angle (in degrees, from 0 to 90) to adjust the level of simplification performed by the algorithm
Length	- A length (in meters) to adjust the level of simplification performed by the algorithm
M	- A $k \times 2$ matrix that represents a combined effect for the Angle and Length parameters. First column represents a partition of the length space, whereas the second contains the angles considered in the simplification procedure when the lengths of both edges meeting in a second-degree vertex are below the threshold indicated in the first column

Value

Returns a simplification of the linear network passed as an input (linnet class)

Examples

```
library(SpNetPrep)
library(spatstat)
library(sp)
library(raster)
library(maptools)
network <- chicago$domain
Angle <- 10
Length <- 20
simplified_network_1 <- SimplifyLinearNetwork(network,Angle=Angle,Length=Length)
## Not run:
M <- matrix(c(10,60,40,25),nrow=2)
simplified_network_2 <- SimplifyLinearNetwork(network,M=M)

## End(Not run)
```

Index

*Topic **datasets**

SampleDirectedNetwork, [2](#)

SampleNetwork, [3](#)

SamplePointPattern, [3](#)

runAppSpNetPrep, [1](#)

SampleDirectedNetwork, [2](#)

SampleNetwork, [3](#)

SamplePointPattern, [3](#)

SimplifyLinearNetwork, [3](#)