Package ‘EpiContactTrace’

February 19, 2015

Title  Epidemiological tool for contact tracing
Version  0.8.8
Date  2013-11-04
Description  Routines for epidemiological contact tracing
            and visualisation of network of contacts.
License  EUPL
URL  https://github.com/stewid/EpiContactTrace
Type  Package
LazyLoad  yes
Imports  plyr, R2HTML
Depends  R(>= 3.0.0), methods, Rcpp (>= 0.9.13)
Suggests  animation, ggmap, testthat
Collate  ‘Contacts.r’ ‘ContactTrace.r’ ‘EpiContactTrace-package.r’
         ‘trace.R’ ‘in-degree.r’ ‘out-degree.r’
         ‘ingoing-contact-chain.r’ ‘outgoing-contact-chain.r’
         ‘network-summary.r’ ‘network-structure.r’ ‘report.r’
         ‘shortest-paths.r’ ‘show.r’ ‘animate.r’
LinkingTo  Rcpp
Author  Maria Noremark [aut],
        Stefan Widgren [aut, cre]
Maintainer  Stefan Widgren <stefan.widgren@sva.se>
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2013-11-05 07:48:35

R topics documented:

  EpiContactTrace-package ...........................................  2
  Animate ..............................................................  3
  Contacts-class ......................................................  5
EpiContactTrace-package

Epidemiological tool for contact tracing.

Description

Routines for epidemiological contact tracing and visualisation of network of contacts.

Details

In many countries, livestock movement data are collected with the major objective to enable contact tracing during disease outbreaks. Livestock movement data can also be of relevance for risk based surveillance - both during outbreak or when investigating if a disease is present in the population. However, the livestock movement databases are not always structured in such a way that relevant information for contact tracing or surveillance design is easily retrieved. EpiContactTrace uses the network parameters in-degree, out-degree, ingoing contact-chain and outgoing contact-chain, which are relevant for forward- and backward contact-tracing respectively. The measures can also be used for identifying herds with many contacts, which can be used in risk based disease surveillance. Different time periods for ingoing and outgoing contacts can be of interest in the contact tracing, based on possible window of introduction, and this can be adjusted in the tool. The output from the analysis is available as a dataset, but moreover, the tool automatically generates a report on farm level. The report both contains an overview of the situation on the farm, including a graph, as well as detailed information including dates of movements on group or individual level on all contacts.

Maintainer

Stefan Widgren <stefan.widgren@sva.se>
Note

Plots are not supported in version 0.8.6 since igraph0 has been archived. We intend to resolve the issue in a future version. Install version 0.8.5 and igraph0 manually from the archive if plots are required. See section 6.3 in 'R Installation and Administration' on how to install packages from source.

Author(s)

Stefan Widgren Maria Noremark

References


Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contacttrace <- trace(movements=transfers,
    root='R6TUL',
    tEnd='2005-10-31',
    days=90)

show(contacttrace)

## Not run:
## Generate an html report showing details of the contact tracing for
## root 2645.
## Note: Creates the files 2645.html and 2645.png in the working
## directory.
Report(contacttrace)

## End(Not run)
```

Description

Visualize and animate movements on a map.
Animate

Usage

Animate(movements, coords, map,
    interval = c("all", "day", "week", "month", "quarter", "year"),
    outdir = getwd(), title = "Animation of contacts")

Arguments

movements a data.frame data.frame with movements, see details.
coords a data.frame data.frame with coordinates, see details.
map a ggmap object to use as background map, see `get_map`.
interval the time interval to aggregate movements in the animation. Can be any of 'all', 'day', 'week', 'month', 'quarter' or 'year'. Defaults to 'all', which aggregates all movements on one map.
outdir the output directory for the animation, see `ani.options`. Defaults to `getwd()`.
title the title of the animation in the HTML, see `ani.options`. Defaults to 'Animation of contacts'.

Details

The argument movements in Animate is a data.frame with the following columns:

- **source** an integer or character identifier of the source holding.
- **destination** an integer or character identifier of the destination holding.
- **t** the Date of the transfer

The argument coords in Animate is a data.frame with the following columns:

- **id** an integer or character identifier of the holding.
- **lat** the latitude of holding.
- **lon** the longitude of the holding.

Value

invisible(NULL)

Note

The packages animation and ggmap must be installed for this functionality.

References

## Examples

```r
## Not run:
require(ggmap)

data(transfers)

## First extract all source and destination from the dataset
root <- unique(c(transfers$source, transfers$destination))

## For this example, generate a random coordinate for each
## holding. Note that some coordinates might end up in water,
## but the coordinates are only for demonstrating the animation.
ngen <- length(root)
set.seed(123)
lon_min <- 13
lon_max <- 17
lat_min <- 56
lat_max <- 63

lon <- lon_min + runif(ngen) * (lon_max - lon_min)
lat <- lat_min + runif(ngen) * (lat_max - lat_min)
coords <- data.frame(id=root, lon, lat)

## Fetch a map over Sweden
sweden <- get_map('Sweden', zoom=5)

## Select a subset of all movements to visualize
i <- sample(seq_len(nrow(transfers)), 100, replace=FALSE)

## Perform the animation and view the movements aggregated by week
## in a web-browser.
Animate(transfers[i,], coords, sweden, "week")

## End(Not run)
```

---

### Description

Class to handle contacts.

### Details

The `Contacts` class keeps track of all ingoing or outgoing livestock transfers in the contact chain for a specific root within the time window used for contact tracing. The slots; source, destination, t, id, n and category contains contact information extracted from the movement dataset during contact tracing. The index slot is an index to the extracted contacts within the class that together with the distance slot can be used to rebuild the exact contacts that were extracted from each search step during the contact tracing.
ContactTrace-class

Slots

root A character vector of length one with the identifier of the root.
tBegin A Date vector of length one with the start date of the time window used for contact tracing.
tEnd A Date vector of length one with the end date of the time window used for contact tracing.
source A character vector with the identifiers of the source holdings of the livestock transfer.
destination A character vector with the identifier of the destination holdings of the livestock transfer.
t A Date vector of the livestock transfer.
id A character vector with the identifiers of the animals.
n A numeric vector with the number of animals transferred.
category A character vector with the category of animals e.g. cattle.
index A integer index vector.
distance A integer vector with the distance from root for the contact[index]
direction A character vector of length one equal to the direction ‘in’ or ‘out’ of the contacts.

Objects from the Class

Objects can be created by calls of the form new("Contacts", root, startDate, days, source, destination, t, id, n, category, index, distance, direction).

Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contactTrace <- Trace(movements = transfers,
  root = 2645,
  tEnd = "2005-10-31",
  days = 90)

## Show structure of ingoing contacts
str(contactTrace@ingoingContacts)

## Show structure of ougoing contacts
str(contactTrace@outgoingContacts)
```

ContactTrace-class Class "ContactTrace"

Description

Class to handle contact tracing.
Details

The ContactTrace class holds information for the ingoing and outgoing contact chain for a specific root within the time window used for contact tracing.

Slots

- **root**: A character vector of length one with the identifier of the root.
- **ingoingContacts**: A Contacts object with the contacts for the ingoing contact chain.
- **outgoingContacts**: A Contacts object with the contacts for the outgoing contact chain.

Objects from the Class

Objects can be created by calls of the form `new("ContactTrace", root, ingoingContacts, outgoingContacts,...)

Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contactTrace <- Trace(movements = transfers,
                       root = 2645,
                       tEnd = '2005-10-31',
                       days = 90)

## Show structure
str(contactTrace)
```

### InDegree-methods

**InDegree**

**Description**

The number of herds with direct movements of animals to the root herd during the defined time window used for tracing.

**Arguments**

- **x**: a ContactTrace object, or a list of ContactTrace objects or a data.frame with movements of animals between holdings, see `Trace` for details.
- **root**: vector of roots to calculate indegree for.
- **tEnd**: the last date to include ingoing movements. Defaults to NULL.
- **days**: the number of previous days before tEnd to include ingoing movements. Defaults to NULL.
- **inBegin**: the first date to include ingoing movements. Defaults to NULL.
- **inEnd**: the last date to include ingoing movements. Defaults to NULL.
Details

The time period used for `InDegree` can either be specified using `tEnd` and `days` or `inBegin` and `inEnd`.

If using `tEnd` and `days`, the time period for ingoing contacts ends at `tEnd` and starts at `days` prior to `tEnd`. The indegree will be calculated for each combination of `root`, `tEnd` and `days`.

An alternative way is to use `inBegin` and `inEnd`. The time period for ingoing contacts starts at `inBegin` and ends at `inEndDate`. The vectors `root inBegin`, `inEnd` must have the same lengths and the indegree will be calculated for each index of them.

The movements in `InDegree` is a `data.frame` with the following columns:

- **source**: an integer or character identifier of the source holding.
- **destination**: an integer or character identifier of the destination holding.
- **t**: the Date of the transfer
- **id**: an optional character vector with the identity of the animal.
- **n**: an optional numeric vector with the number of animals moved.
- **category**: an optional character or factor with category of the animal e.g. Cattle.

Value

A `data.frame` with the following columns:

- **root**: The root of the contact tracing
- **inBegin**: The first date to include ingoing movements
- **inEnd**: The last date to include ingoing movements
- **inDays**: The number of days in the interval `inBegin` to `inEnd`
- **inDegree**: The `InDegree` of the root within the time-interval

Methods

- `signature(x = "ContactTrace")`: Get the `InDegree` of a `ContactTrace` object.
- `signature(x = "list")`: Get the `InDegree` for a list of `ContactTrace` objects. Each item in the list must be a `ContactTrace` object.
- `signature(x = "data.frame")`: Get the `InDegree` for a `data.frame` with movements, see details and examples.

References

**Description**

The ingoing contact chain is the number of holdings in the network of direct and indirect contacts to the root holding, with regard to temporal and order of the contacts during the defined time window used for contact tracing.

---

### Examples

```
## Load data
data(transfers)

## Perform contact tracing using tEnd and days
callTrace <- Trace(movements=transfers,
                  root=2645,
                  tEnd='2005-10-31',
                  days=91)

## Calculate indegree from a ContactTrace object
id.1 <- indegree(contactTrace)

## Calculate indegree using tEnd and days
id.2 <- indegree(transfers,
                  root=2645,
                  tEnd='2005-10-31',
                  days=91)

## Check that the result is identical
identical(id.1, id.2)

## Not run:
## Calculate indegree for all included herds
## First extract all source and destination from the dataset
root <- sort(unique(c(transfers$source,
                      transfers$destination)))

## Calculate indegree
result <- indegree(transfers,
                    root=root,
                    tEnd='2005-10-31',
                    days=91)

## End(Not run)
```
IngoingContactChain-methods

Arguments

x          a ContactTrace object, or a list of ContactTrace objects or a data.frame with movements of animals between holdings, see Trace for details.
root       vector of roots to calculate ingoing contact chain for.
tEnd       the last date to include ingoing movements. Defaults to NULL
days       the number of previous days before tEnd to include ingoing movements. Defaults to NULL
inBegin    the first date to include ingoing movements. Defaults to NULL
inEnd      the last date to include ingoing movements. Defaults to NULL

Details

The time period used for IngoingContactChain can either be specified using tEnd and days or inBegin and inEnd.

If using tEnd and days, the time period for ingoing contacts ends at tEnd and starts at days prior to tEnd. The indegree will be calculated for each combination of root, tEnd and days.

An alternative way is to use inBegin and inEnd. The time period for ingoing contacts starts at inBegin and ends at inEndDate. The vectors root inBegin, inEnd must have the same lengths and the indegree will be calculated for each index of them.

The movements in IngoingContactChain is a data.frame with the following columns:

source an integer or character identifier of the source holding.
destination an integer or character identifier of the destination holding.
t the Date of the transfer
id   an optional character vector with the identity of the animal.
n an optional numeric vector with the number of animals moved.
category an optional character or factor with category of the animal e.g. Cattle.

Value

A data.frame with the following columns:

root The root of the contact tracing
inBegin The first date to include ingoing movements
inEnd The last date to include ingoing movements
inDays The number of days in the interval inBegin to inEnd
ingoingContactChain The IngoingContactChain of the root within the time-interval

Methods

signature(x = "ContactTrace") Get the IngoingContactChain of a ContactTrace object.
signature(x = "list") Get the IngoingContactChain for a list of ContactTrace objects. Each item in the list must be a ContactTrace object.
signature(x = "data.frame") Get the IngoingContactChain for a data.frame with movements, see details and examples.
IngoingContactChain-methods

References


See Also

NetworkSummary

Examples

```r
## Load data
data(transfers)

## Perform contact tracing using tEnd and days
contactTrace <- Trace(movements=transfers,
  root=2645,
  tEnd='2005-10-31',
  days=91)

## Calculate ingoing contact chain from a ContactTrace object
ic.1 <- IngoingContactChain(contactTrace)

## Calculate ingoing contact chain using tEnd and days
ic.2 <- IngoingContactChain(transfers,
  root=2645,
  tEnd='2005-10-31',
  days=91)

## Check that the result is identical
identical(ic.1, ic.2)

## Not run:
## Calculate ingoing contact chain for all included herds
## First extract all source and destination from the dataset
root <- sort(unique(c(transfers$source,
  transfers$destination))

## Calculate ingoing contact chain
result <- IngoingContactChain(transfers,
  root=root,
  tEnd='2005-10-31',
  days=91)

## End(Not run)
```
Description

Methods for function `NetworkStructure` in package `EpiContactTrace` to get the network tree structure from the contact tracing.

Details

The contact tracing performs a depth first search starting at the root. The `NetworkStructure` gives the distance from root at each node. The network tree structure given by the depth first search is shown by `show`.

Value

A `data.frame` with the following columns:

- `root` The root of the contact tracing
- `inBegin` If the direction is ingoing, then `inBegin` equals `inBegin` in `Trace` else NA.
- `inEnd` If the direction is ingoing, then `inEnd` equals `inEnd` in `Trace` else NA.
- `outBegin` If the direction is outgoing, then `outBegin` equals `outBegin` in `Trace` else NA.
- `outEnd` If the direction is outgoing, then `outEnd` equals `outEnd` in `Trace` else NA.
- `direction` If the direction is ingoing, then direction equals 'in' else 'out'
- `source` The source of the contacts in the depth first search
- `destination` The destination of the contacts in the depth first search
- `distance` The distance from the destination to root in the depth first search

Methods

- `signature(object = "Contacts")` Get the network structure for the Contacts object.
- `signature(object = "ContactTrace")` Get the network structure for the ingoing and outgoing Contacts of a ContactTrace object.
- `signature(object = "list")` Get the network structure for a list of ContactTrace objects. Each item in the list must be a ContactTrace object.

See Also

`show`
Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contactTrace <- Trace(movements=transfers, root=2645, tEnd='2005-10-31', days=90)

NetworkStructure(contactTrace)
```

Description

NetworkSummary gives a summary of the contact tracing including the time-window, InDegree, OutDegree, IngoingContactChain and OutgoingContactChain.

Details

The time period used for NetworkSummary can either be specified using tEnd and days or inBegin, inEnd, outBegin and outEnd.

If using tEnd and days, the time period for ingoing and outgoing contacts ends at tEnd and starts at days prior to tEnd. The network summary will be calculated for each combination of root, tEnd and days.

An alternative way is to use inBegin, inEnd, outBegin and outEnd. The time period for ingoing contacts starts at inBegin and ends at inEndDate. For outgoing contacts the time period starts at outBegin and ends at outEnd. The vectors root inBegin, inEnd, outBegin and outEnd must have the same lengths and the network summary will be calculated for each index of them.

The movements in NetworkSummary is a data.frame with the following columns:

- **source** an integer or character identifier of the source holding.
- **destination** an integer or character identifier of the destination holding.
- **t** the Date of the transfer
- **id** an optional character vector with the identity of the animal.
- **n** an optional numeric vector with the number of animals moved.
- **category** an optional character or factor with category of the animal e.g. Cattle.
Value
A `data.frame` with the following columns:

- **root**  The root of the contact tracing
- **inBegin**  Equals `inBegin` in `Trace`
- **inEnd**  Equals `inEnd` in `Trace`
- **outBegin**  Equals `outBegin` in `Trace`
- **outEnd**  Equals `outEnd` in `Trace`
- **inDegree**  The `InDegree` of the contact tracing
- **outDegree**  The `OutDegree` of the contact tracing
- **ingoingContactChain**  The `IngoingContactChain` of the contact tracing
- **outgoingContactChain**  The `OutgoingContactChain` of the contact tracing

Methods

```r
signature(x = "ContactTrace")  Get the network summary for the ingoing and outgoing Contacts of a ContactTrace object.
signature(x = "list")  Get the network summary for a list of ContactTrace objects. Each item in the list must be a ContactTrace object.
signature(x = "data.frame")  Get the network summary for a data.frame with movements, see details and examples.
```

References


Examples

```r
## Load data
data(transfers)

## Perform contact tracing using tEnd and days
contactTrace <- Trace(movements=transfers,
                      root=2645,
                      tEnd='2005-10-31',
                      days=91)

## Calculate network summary from a ContactTrace object
ns.1 <- NetworkSummary(contactTrace)

## Calculate network summary using tEnd and days
ns.2 <- NetworkSummary(transfers,
```
root=2645,
tEnd='2005-10-31',
days=91)

## Check that the result is identical
identical(ns.1, ns.2)

## Calculate network summary using inBegin, inEnd
## outBegin and outEnd
ns.3 <- NetworkSummary(transfers, 
  root=2645,
  inBegin='2005-08-01',
  inEnd='2005-10-31',
  outBegin='2005-08-01',
  outEnd='2005-10-31')

## Check that the result is identical
identical(ns.2, ns.3)

## Not run:

## When calculating the network summary for a data.frame of movements
## a data.frame for each combination of root, tEnd and days are returned.
root <- c(1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L, 1L)
tEnd <- c("2005-09-01", "2005-10-01")
days <- c(30, 45)

## The network summary are calculated at the following
## 12 combinations.
## 12 combinations.
## root = 1, tEnd = "2005-09-01", days = 30
## root = 1, tEnd = "2005-09-01", days = 45
## root = 1, tEnd = "2005-10-01", days = 30
## root = 1, tEnd = "2005-10-01", days = 45
## root = 2, tEnd = "2005-09-01", days = 30
## root = 2, tEnd = "2005-09-01", days = 45
## root = 2, tEnd = "2005-10-01", days = 30
## root = 2, tEnd = "2005-10-01", days = 45
## root = 3, tEnd = "2005-09-01", days = 30
## root = 3, tEnd = "2005-09-01", days = 45
## root = 3, tEnd = "2005-10-01", days = 30
## root = 3, tEnd = "2005-10-01", days = 45
NetworkSummary(transfers, root, tEnd, days)

## Create a network summary for all included herds
## First extract all source and destination from the dataset
root <- sort(unique(c(transfers$source,
  transfers$destination)))

## Perform contact tracing using tEnd and days
result.1 <- NetworkSummary(transfers, 
  root=root,
  tEnd='2005-10-31',
  days=90)
OutDegree-methods

## Perform contact tracing using inBegin, inEnd, outBegin and outEnd.
result.2 <- NetworkSummary(transfers,
  root=root,
  inBegin=rep('2005-08-02', length(root)),
  inEnd=rep('2005-10-31', length(root)),
  outBegin=rep('2005-08-02', length(root)),
  outEnd=rep('2005-10-31', length(root)))

## End(Not run)

---

OutDegree-methods	OutDegree

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>The number of herds with direct movements of animals from the root herd during the defined time window used for tracing.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>x</code></td>
</tr>
<tr>
<td>a ContactTrace object, or a list of ContactTrace objects or a data.frame with movements of animals between holdings, see <code>trace</code> for details.</td>
</tr>
<tr>
<td><code>root</code></td>
</tr>
<tr>
<td>vector of roots to calculate outdegree for.</td>
</tr>
<tr>
<td><code>tEnd</code></td>
</tr>
<tr>
<td>the last date to include outgoing movements. Defaults to <code>NULL</code>.</td>
</tr>
<tr>
<td><code>days</code></td>
</tr>
<tr>
<td>the number of previous days before <code>tEnd</code> to include outgoing movements. Defaults to <code>NULL</code>.</td>
</tr>
<tr>
<td><code>outBegin</code></td>
</tr>
<tr>
<td>the first date to include outgoing movements. Defaults to <code>NULL</code>.</td>
</tr>
<tr>
<td><code>outEnd</code></td>
</tr>
<tr>
<td>the last date to include outgoing movements. Defaults to <code>NULL</code>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>The time period used for <code>OutDegree</code> can either be specified using <code>tEnd</code> and <code>days</code> or <code>outBegin</code> and <code>outEnd</code>.</td>
</tr>
<tr>
<td>If using <code>tEnd</code> and <code>days</code>, the time period for outgoing contacts ends at <code>tEnd</code> and starts at <code>days</code> prior to <code>tEnd</code>. The outdegree will be calculated for each combination of <code>root</code>, <code>tEnd</code> and <code>days</code>.</td>
</tr>
<tr>
<td>An alternative way is to use <code>outBegin</code> and <code>outEnd</code>. The time period for outgoing contacts starts at <code>outBegin</code> and ends at <code>outEndDate</code>. The vectors <code>root</code>, <code>outBegin</code>, <code>outEnd</code> must have the same lengths and the outdegree will be calculated for each index of them.</td>
</tr>
<tr>
<td>The movements in <code>OutDegree</code> is a data.frame with the following columns:</td>
</tr>
<tr>
<td><code>source</code></td>
</tr>
<tr>
<td><code>destination</code></td>
</tr>
<tr>
<td><code>t</code></td>
</tr>
<tr>
<td><code>id</code></td>
</tr>
<tr>
<td><code>n</code></td>
</tr>
<tr>
<td><code>category</code></td>
</tr>
</tbody>
</table>
OutDegree-methods

Value

A data.frame with the following columns:

- root The root of the contact tracing
- outBegin The first date to include outgoing movements
- outEnd The last date to include outgoing movements
- outDays The number of days in the interval outBegin to outEnd
- outDegree The OutDegree of the root within the time-interval

Methods

signature(x = "ContactTrace") Get the OutDegree of a ContactTrace object.
signature(x = "list") Get the OutDegree for a list of ContactTrace objects. Each item in the list must be a ContactTrace object.
signature(x = "data.frame") Get the OutDegree for a data.frame with movements, see details and examples.

References


See Also

NetworkSummary

Examples

```r
## Load data
data(transfers)

## Perform contact tracing using tEnd and days
contactTrace <- Trace(movements=transfers,
                      root=2645,
                      tEnd='2005-10-31',
                      days=91)

## Calculate outdegree from a ContactTrace object
od.1 <- OutDegree(contactTrace)

## Calculate outdegree using tEnd and days
od.2 <- OutDegree(transfers,
                   root=2645,
                   tEnd='2005-10-31',
                   days=91)
```
OutgoingContactChain-methods

Description

The outgoing contact chain is the number of holdings in the network of direct and indirect contacts from the root holding, with regard to temporal and order of the contacts during the defined time window used for contact tracing.

Arguments

- **x**: a ContactTrace object, or a list of ContactTrace objects or a data.frame with movements of animals between holdings, see Trace for details.
- **root**: vector of roots to calculate outgoing contact chain for.
- **tEnd**: the last date to include outgoing movements. Defaults to NULL.
- **days**: the number of previous days before tEnd to include outgoing movements. Defaults to NULL.
- **outBegin**: the first date to include outgoing movements. Defaults to NULL.
- **outEnd**: the last date to include outgoing movements. Defaults to NULL.

Value

A data.frame with the following columns:

- **root**: The root of the contact tracing
- **outBegin**: The first date to include outgoing movements
- **outEnd**: The last date to include outgoing movements
- **outDays**: The number of days in the interval outBegin to outEnd
- **outDegree**: The OutgoingContactChain of the root within the time-interval
### Methods

- **signature(x = "ContactTrace")** Get the OutgoingContactChain of a ContactTrace object.
- **signature(x = "list")** Get the OutgoingContactChain for a list of ContactTrace objects. Each item in the list must be a ContactTrace object.
- **signature(x = "data.frame")** Get the OutgoingContactChain for a data.frame with movements, see examples.

### References


### See Also

- NetworkSummary

### Examples

#### Load data

```r
data(transfers)
```

#### Perform contact tracing using tEnd and days

```r
contactTrace <- Trace(movements=transfers, 
  root=2645, 
  tEnd='2005-10-31',
  days=91)
```

#### Calculate outgoing contact chain from a ContactTrace object

```r
oc.1 <- OutgoingContactChain(contactTrace)
```

#### Calculate outgoing contact chain using tEnd and days

```r
oc.2 <- OutgoingContactChain(transfers, 
  root=2645, 
  tEnd='2005-10-31',
  days=91)
```

#### Check that the result is identical

```r
identical(oc.1, oc.2)
```

#### Not run:

#### Calculate outgoing contact chain for all included herds

```r
# First extract all source and destination from the dataset 
root <- sort(unique(c(transfers$source, transfers$destination)))

# Calculate outgoing contact chain
```
result <- OutgoingContactChain(transfers,
    root=root,
    tEnd='2005-10-31',
    days=91)

## End(Not run)

---

### Generate a contact tracing Report

#### Description

EpiContatTrace contains report templates to generate pdf- or html reports for the farm specific contacts. These reports can be useful for hands-on disease tracing in the field. The templates are used by Sweave and can be adapted by the end user. However, in the default setting the report has the following layout: first the contacts are visualised graphically in a plot, as to give an immediate signal to the reader of the report of the number of contacts. In the following, the contact data are presented with different levels of detail split by ingoing and outgoing contacts. The first includes collapsed data and the sequential contact structure at group level (i.e. no information on individuals or dates). In this summary, the sequential structure of each part of the chain is included, and a holding that appears in several different parts of the chain can therefore be included more than once in the summary. The reason for this is to facilitate sequential tracing and getting an overview of each part of the chain. After the summary all details of all contacts included in the contact chains is presented, i.e. date of contact and data on individual level when available. To generate pdf files a TeX installation must exist to compile the latex file. The report is saved in the working directory with the name of the root as filename.

#### Arguments

- **object**
  - the object
- **format**
  - the format to use, can be either 'html' or 'pdf'. The default is 'html'
- **template**
  - the Sweave template file to use. If none is provided, the default is used.

#### Methods

- signature(object = "ContactTrace" ) Generate a report for a ContactTrace object.
- signature(object = "list") Generate reports for a list of ContactTrace objects.

#### Note

Plots are not supported in version 0.8.6 since igraph0 has been archived. We intend to resolve the issue in a future version. Install version 0.8.5 and igraph0 manually from the archive if plots are required. See section 6.3 in ‘R Installation and Administration’ on how to install packages from source.
References


See Also

Sweave, R2HTML, texi2pdf.

Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contactTrace <- trace(movements=transfers,
  root=2645,
  tEnd='2005-10-31',
  days=90)

## Not run:
## Generate an html report showing details of the contact tracing for
## root 2646.
## Note: Creates the files 2645.html and 2645.png in the working
## directory.
Report(contactTrace)

## It's possible to generate reports for a list of ContactTrace objects.
## Perform contact tracing for ten of the included herds
root <- sort(unique(c(transfers$source, transfers$destination)))[1:10]

## Perform contact tracing
contactTrace <- trace(movements=transfers,
  root=root,
  tEnd='2005-10-31',
  days=90)

## Generate reports
## Note: Creates the files 1.html, 2.html, ..., 10.html and
## 1.png, 2.png, ..., 10.png in the working directory
Report(contactTrace)

## End(Not run)
```
**Description**

EpiContaTrace contains report templates to generate pdf- or html reports for the farm specific contacts. These reports can be useful for hands-on disease tracing in the field. The templates are used by Sweave and can be adapted by the end user. This method enables communication of the current ContactTrace object to the report.

**Usage**

```r
ReportObject()
```

**Value**

The current ContactTrace object when generating a report

---

**ShortestPaths-methods**

**Description**

Methods for function ShortestPaths in package `EpiContactTrace` to get the shortest distance from/to the root given by the contact tracing.

**Details**

The contact tracing performs a depth first search starting at the root. The ShortestPaths gives the shortest distance from root at each node. The network tree structure given by the depth first search is shown by `show`.

**Value**

A data.frame with the following columns:

- **root** The root of the contact tracing
- **inBegin** If the direction is ingoing, then inBegin equals inBegin in Trace else NA.
- **inEnd** If the direction is ingoing, then inEnd equals inEnd in Trace else NA.
- **outBegin** If the direction is outgoing, then outBegin equals outBegin in Trace else NA.
- **outEnd** If the direction is outgoing, then outEnd equals outEnd in Trace else NA.
- **direction** If the direction is ingoing, then direction equals 'in' else 'out'
- **source** The source of the contacts in the depth first search
- **destination** The destination of the contacts in the depth first search
- **distance** The shortest distance from/to root in the depth first search
Methods

signature(object = "Contacts") Get the shortest paths for the Contacts object.
signature(object = "ContactTrace") Get the shortest paths for the ingoing and outgoing Contacts of a ContactTrace object.
signature(object = "list") Get the shortest paths for a list of ContactTrace objects. Each item in the list must be a ContactTrace object.

See Also

show and NetworkStructure.

Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contactTrace <- Trace(movements=transfers, 
  root=2645, 
  tEnd='2005-10-31', 
  days=90)

ShortestPaths(contactTrace)
```

Description

Shows information of the time-window used for contact tracing and summary of network parameters. It also visualize the contact structure.

Methods

signature(object = "Contacts") Show information for the Contacts object.
signature(object = "ContactTrace") Show information for the ingoing and outgoing Contacts of a ContactTrace object.

References

Examples

```r
## Load data
data(transfers)

## Perform contact tracing
contactTrace <- Trace(movements=transfers, root=2645, tEnd='2005-10-31', days=90)
show(contactTrace)
```

### Trace

**Trace Contacts.**

### Description

Contact tracing for a specified node(s) (root) during a specified time period. The time period is divided into two parts, one for ingoing contacts and one for outgoing contacts.

### Usage

```r
Trace(movements, root, tEnd, days, inBegin, inEnd, outBegin, outEnd)
```

### Arguments

- `movements` a data frame data.frame with movements, see details.
- `root` vector of roots to perform contact tracing for.
- `tEnd` the last date to include ingoing and outgoing movements. Defaults to NULL.
- `days` the number of previous days before tEnd to include ingoing and outgoing movements. Defaults to NULL.
- `inBegin` the first date to include ingoing movements. Defaults to NULL.
- `inEnd` the last date to include ingoing movements. Defaults to NULL.
- `outBegin` the first date to include outgoing movements. Defaults to NULL.
- `outEnd` the last date to include outgoing movements. Defaults to NULL.

### Details

The time period used for Trace can either be specified using `tEnd` and `days` or `inBegin`, `inEnd`, `outBegin` and `outEnd`.

If using `tEnd` and `days`, the time period for ingoing and outgoing contacts ends at `tEnd` and starts at `days` prior to `tEnd`. The tracing will be performed for each combination of `root`, `tEnd` and `days`.

An alternative way is to use `inBegin`, `inEnd`, `outBegin` and `outEnd`. The time period for ingoing contacts starts at `inBegin` and ends at `inEndDate`. For outgoing contacts the time period starts at
outBegin and ends at outEnd. The vectors root inBegin, inEnd, outBegin and outEnd must have the same lengths and the tracing will be performed for each index of them.

The argument movements in Trace is a data.frame with the following columns:

- **source** an integer or character identifier of the source holding.
- **destination** an integer or character identifier of the destination holding.
- **t** the Date of the transfer
- **id** an optional character vector with the identity of the animal.
- **n** an optional numeric vector with the number of animals moved.
- **category** an optional character or factor with category of the animal e.g. Cattle.

**References**


**Examples**

```r
## Load data
data(transfers)

## Perform contact tracing using tEnd and days
trace.1 <- Trace(movements=transfers, 
  root=2645, 
  tEnd='2005-10-31',
  days=91)

## Perform contact tracing using inBegin, inEnd
## outBegin and outEnd
trace.2 <- Trace(movements=transfers, 
  root=2645, 
  inBegin='2005-08-01',
  inEnd='2005-10-31',
  outBegin='2005-08-01',
  outEnd='2005-10-31')

## Check that the result is identical
identical(trace.1, trace.2)

## Show result of contact tracing
show(trace.1)

## Not run:

## Create a network summary for all included herds
```
## First extract all source and destination from the dataset

```r
root <- sort(unique(c(transfers$source,
                      transfers$destination)))
```

## Perform contact tracing using \( tEnd \) and \( \text{days} \)

```r
trace.3 <- Trace(movements=transfers,
                 root=root,
                 tEnd='2005-10-31',
                 days=91)
```

## Perform contact tracing using \( \text{inBegin}, \text{inEnd} \)
## \( \text{outBegin} \) and \( \text{outEnd} \)

```r
trace.4 <- Trace(movements=transfers,
                 root=root,
                 inBegin=rep('2005-08-01', length(root)),
                 inEnd=rep('2005-10-31', length(root)),
                 outBegin=rep('2005-08-01', length(root)),
                 outEnd=rep('2005-10-31', length(root)))
```

## Check that the result is identical

```r
identical(trace.3, trace.4)
```

NetworkSummary(trace.3)

## End(Not run)

### Movement Example Data

#### Description

Movement data included in the package. The data contains fictitious example data of cattle movements during the period 2005-08-01 – 2005-10-31.

#### Format

A data frame with 70190 observations on the following 6 variables.

- **source** a numeric vector with the holding identifier of the source
- **destination** a numeric vector with holding identifier of the destination.
- **id** a character vector with the identity of the animal. In this dataset an 5 character hexadecimal vector.
- **t** a Date of the transfers
- **n** a numeric vector with the number of animals moved. Always 1 in this dataset.
- **category** a factor describing the category of the animal. Always Cattle in this dataset.
Examples

```r
## Not run:
data(transfers)

contactTrace <- Trace(movements=transfers, 
                      root=2645, 
                      tEnd='2005-10-31', 
                      days=90)

show(contactTrace)

## End(Not run)
```
Index

*Topic **classes**
- Contacts-class, 5
- ContactTrace-class, 6

*Topic **datasets**
- transfers, 26

*Topic **methods**
- InDegree-methods, 7
- IngoingContactChain-methods, 9
- NetworkStructure-methods, 12
- NetworkSummary-methods, 13
- OutDegree-methods, 16
- OutgoingContactChain-methods, 18
- Report-methods, 20
- ShortestPaths-methods, 22
- show-methods, 23

*Topic **package**
- EpiContactTrace-package, 2
  - ani.options, 4
  - Animate, 3
  - Contacts-class, 5
  - ContactTrace-class, 6
  - EpiContactTrace
    - (EpiContactTrace-package), 2
  - EpiContactTrace-package, 2
  - get_map, 4
  - InDegree, 8, 13, 14
  - InDegree (InDegree-methods), 7
  - InDegree,Contacts-method (InDegree-methods), 7
  - InDegree,ContactTrace-method (InDegree-methods), 7
  - InDegree,data.frame-method (InDegree-methods), 7
  - InDegree,list-method (InDegree-methods), 7
  - InDegree-methods, 7
  - IngoingContactChain, 10, 13, 14
  - IngoingContactChain
    - (IngoingContactChain-methods), 9
  - IngoingContactChain,Contacts-method
    - (IngoingContactChain-methods), 9
  - IngoingContactChain,ContactTrace-method
    - (IngoingContactChain-methods), 9
  - IngoingContactChain,data.frame-method
    - (IngoingContactChain-methods), 9
  - IngoingContactChain,list-method
    - (IngoingContactChain-methods), 9
  - IngoingContactChain-methods, 9
  - NetworkStructure, 23
  - NetworkStructure
    - (NetworkStructure-methods), 12
  - NetworkStructure,Contacts-method
    - (NetworkStructure-methods), 12
  - NetworkStructure,ContactTrace-method
    - (NetworkStructure-methods), 12
  - NetworkStructure,list-method
    - (NetworkStructure-methods), 12
  - NetworkStructure-methods, 12
  - NetworkSummary, 9, 11, 17, 19
  - NetworkSummary
    - (NetworkSummary-methods), 13
  - NetworkSummary,ContactTrace-method
    - (NetworkSummary-methods), 13
  - NetworkSummary,data.frame-method
    - (NetworkSummary-methods), 13
  - NetworkSummary,list-method
    - (NetworkSummary-methods), 13
  - NetworkSummary-methods, 13
  - OutDegree, 13, 14, 17

28
INDEX

OutDegree (OutDegree-methods), 16
OutDegree, Contacts-method
  (OutDegree-methods), 16
OutDegree, ContactTrace-method
  (OutDegree-methods), 16
OutDegree, data.frame-method
  (OutDegree-methods), 16
OutDegree, list-method
  (OutDegree-methods), 16
OutDegree-methods, 16
OutgoingContactChain, 13, 14, 18
OutgoingContactChain
  (OutgoingContactChain-methods), 18
OutgoingContactChain, Contacts-method
  (OutgoingContactChain-methods), 18
OutgoingContactChain, ContactTrace-method
  (OutgoingContactChain-methods), 18
OutgoingContactChain, data.frame-method
  (OutgoingContactChain-methods), 18
OutgoingContactChain, list-method
  (OutgoingContactChain-methods), 18
OutgoingContactChain-methods, 18

Report (Report-methods), 20
Report, ContactTrace-method
  (Report-methods), 20
Report, list-method (Report-methods), 20
Report-methods, 20
ReportObject, 22

ShortestPaths (ShortestPaths-methods), 22
ShortestPaths, Contacts-method
  (ShortestPaths-methods), 22
ShortestPaths, ContactTrace-method
  (ShortestPaths-methods), 22
ShortestPaths, list-method
  (ShortestPaths-methods), 22
ShortestPaths-methods, 22
show, 12, 22, 23
show (show-methods), 23
show, Contacts-method (show-methods), 23
show, ContactTrace-method
  (show-methods), 23
show-methods, 23
Trace, 7, 10, 12, 14, 16, 18, 22, 24
transfers, 26
transfers, 26

outdegree
(outdegree-methods), 16
outdegreeLcontacts-method
(outdegree-methods), 16
outdegreeLcontacttrace-method
(outdegree-methods), 16
outdegreeLdataNframe-method
(outdegree-methods), 16
outdegreeLlist-method
(outdegree-methods), 16
outdegree-methods, 16
outgoingcontactchain
(outgoingcontactchain-methods), 18
outgoingcontactchainLcontacts-method
(outgoingcontactchain-methods), 18
outgoingcontactchainLcontacttrace-method
(outgoingcontactchain-methods), 18
outgoingcontactchainLdataNframe-method
(outgoingcontactchain-methods), 18
outgoingcontactchainLlist-method
(outgoingcontactchain-methods), 18
outgoingcontactchain-methods, 18
report
(report-methods), 20
reportLcontacttrace-method
(report-methods), 20
reportLlist-method
(report-methods), 20
report-methods, 20
reportobject, 22
shortestpaths
(shortestpaths-methods), 22
shortestpathsLcontacts-method
(shortestpaths-methods), 22
shortestpathsLcontacttrace-method
(shortestpaths-methods), 22
shortestpathsLlist-method
(shortestpaths-methods), 22
shortestpaths-methods, 22
show, 12, 22, 23
show (show-methods), 23
show, Contacts-method (show-methods), 23
show, ContactTrace-method
  (show-methods), 23
show-methods, 23
Trace, 7, 10, 12, 14, 16, 18, 22, 24
transfers, 26