Package 'EpiContactTrace'

October 4, 2024

Title Epidemiological Tool for Contact Tracing

Version 0.18.0

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

License EUPL

URL https://github.com/stewid/EpiContactTrace

BugReports https://github.com/stewid/EpiContactTrace/issues

Type Package

LazyData true

Depends R(>= 3.0.2)

Imports graphics, methods, tools, utils

Collate 'Contacts.R' 'ContactTrace.R' 'EpiContactTrace-package.R'
'in-degree.R' 'ingoing-contact-chain.R' 'network-structure.R'
'network-summary.R' 'out-degree.R' 'outgoing-contact-chain.R'
'plot.R' 'report.R' 'shortest-paths.R' 'show.R' 'trace.R'
'tree.R'

Encoding UTF-8

RoxygenNote 7.3.2

NeedsCompilation yes

Author Maria Noremark [aut] (<https://orcid.org/0000-0003-2555-8476>),
Stefan Widgren [aut, cre] (<https://orcid.org/0000-0001-5745-2284>)

Maintainer Stefan Widgren <stefan.widgren@gmail.com>

Repository CRAN

Date/Publication 2024-10-04 18:20:02 UTC

Contents

`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.

Author(s)

Maintainer: Stefan Widgren <stefan.widgren@gmail.com> (ORCID)

Authors:

- Maria Noremark (ORCID)

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

See Also

Useful links:

- https://github.com/stewid/EpiContactTrace
- Report bugs at https://github.com/stewid/EpiContactTrace/issues

Examples

```
## Load data
data(transfers)

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

contactTrace

## Generate an html report showing details of the contact tracing
## for root 2645. Note: Creates the files 2645.html and 2645.png
## in the temporary directory.
Report(contactTrace, dir = tempdir())
```

Contacts-class *Class* "Contacts"

Description

Class to handle contacts.

Details

The `Contacts` class keeps track of all ingoing or outgoing livstock 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.

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 transfered.

`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"

Objects from the Class

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

Examples

```
## 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

```
## Load data
data(transfers)

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

## Show structure
str(contactTrace)
```

InDegree InDegree

Description

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

Usage

```
InDegree(x, ...)

## S4 method for signature 'Contacts'
InDegree(x)

## S4 method for signature 'ContactTrace'
InDegree(x)

## S4 method for signature 'data.frame'
InDegree(x, root, tEnd = NULL, days = NULL, inBegin = NULL, inEnd = NULL)
```

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.
...	Additional arguments to the method
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 = "data.frame")` Get the InDegree for a data.frame with movements, see details and examples.

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

See Also

`NetworkSummary`

Examples

```
## Not run:

## Load data
data(transfers)

## Perform contact tracing using tEnd and days
contactTrace <- 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)

## 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 IngoingContactChain

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.

Usage

```
IngoingContactChain(x, ...)

## S4 method for signature 'Contacts'
IngoingContactChain(x)

## S4 method for signature 'ContactTrace'
IngoingContactChain(x)

## S4 method for signature 'data.frame'
IngoingContactChain(
  x,
  root,
  tEnd = NULL,
  days = NULL,
  inBegin = NULL,
  inEnd = NULL
)
```

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.
...	Additional arguments to the method
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 = "data.frame") Get the IngoingContactChain for a data.frame with movements, see details and examples.

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

See Also

NetworkSummary

Examples

```
## Not run:

## 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)

## 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)
```

NetworkStructure NetworkStructure

Description

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

Usage

```
NetworkStructure(object)

## S4 method for signature 'Contacts'
NetworkStructure(object)

## S4 method for signature 'ContactTrace'
NetworkStructure(object)

## S4 method for signature 'list'
NetworkStructure(object)
```

Arguments

object	A Contacts or linkS4class{ContactTrace} object.

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

```
## Load data
data(transfers)

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

NetworkStructure(contactTrace)
```

`NetworkSummary` `NetworkSummary`

Description

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

Usage

```
NetworkSummary(x, ...)

## S4 method for signature 'ContactTrace'
NetworkSummary(x)

## S4 method for signature 'list'
NetworkSummary(x)

## S4 method for signature 'data.frame'
NetworkSummary(
  x,
  root,
  tEnd = NULL,
```

```
  days = NULL,
  inBegin = NULL,
  inEnd = NULL,
  outBegin = NULL,
  outEnd = NULL
)
```

Arguments

x	a ContactTrace object, a data.frame with movements of animals between holdings (see Trace for details), or a list of ContactTrace objects where each item in the list must be a ContactTrace object.
...	Additional arguments to the method
root	vector of roots to calculate network summary 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
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 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

`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

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

Examples

```
## Not run:

## 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)

## 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(1, 2, 3)
tEnd <- c("2005-09-01", "2005-10-01")
days <- c(30, 45)

## The network summary are calculated at the following
## 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)
```

```
## 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 OutDegree

Description

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

Usage

```
OutDegree(x, ...)

## S4 method for signature 'Contacts'
OutDegree(x)

## S4 method for signature 'ContactTrace'
OutDegree(x)

## S4 method for signature 'data.frame'
OutDegree(x, root, tEnd = NULL, days = NULL, outBegin = NULL, outEnd = NULL)
```

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.
...	Additional arguments to the method
root	vector of roots to calculate outdegree 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

Details

The time period used for `OutDegree` can either be specified using `tEnd` and `days` or `outBegin` and `outEnd`.

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

An alternative way is to use `outBegin` and `outEnd`. The time period for outgoing contacts starts at outBegin and ends at outEndDate. The vectors `root` `outBegin`, `outEnd` must have the same lengths and the outdegree will be calculated for each index of them.

The movements in `OutDegree` 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

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 = "data.frame")` Get the OutDegree for a data.frame with movements, see details and examples.

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

See Also

`NetworkSummary`

Examples

```
## Not run:

## 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)

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

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

## Calculate outdegree
result <- OutDegree(transfers,
                    root = root,
                    tEnd = "2005-10-31",
                    days = 91)

## End(Not run)
```

`OutgoingContactChain` `OutgoingContactChain`

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.

Usage

```
OutgoingContactChain(x, ...)
```

```
## S4 method for signature 'Contacts'
OutgoingContactChain(x)

## S4 method for signature 'ContactTrace'
OutgoingContactChain(x)

## S4 method for signature 'data.frame'
OutgoingContactChain(
  x,
  root,
  tEnd = NULL,
  days = NULL,
  outBegin = NULL,
  outEnd = NULL
)
```

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.
...	Additional arguments to the method
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 = "data.frame") Get the OutgoingContactChain for a data.frame with movements, see examples.

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

See Also

NetworkSummary

Examples

```
## Not run:

## Load data
data(transfers)

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

## Calculate outgoing contact chain from a ContactTrace object
oc_1 <- OutgoingContactChain(contactTrace)

## Calculate outgoing contact chain using tEnd and days
oc_2 <- OutgoingContactChain(transfers,
                             root = 2645,
                             tEnd = "2005-10-31",
                             days = 91)

## Check that the result is identical
identical(oc_1, oc_2)

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

plot-methods *plot,-method*

Description

The contact structure can be visualized graphically with a plot. The plot gives an overview of the number of ingoing and outgoing holdings connected to the root holding. The black node is the root holding and all white nodes represent holdings that are direct or indirect holdings with ingoing contacts to root. Grey nodes represent holdings that are direct or indirect holdings with outgoing contacts from root.

Usage

```
## S4 method for signature 'ContactTrace'
plot(x, y, ...)
```

Arguments

x	The ContactTrace object to plot
y	Not used
...	Additional arguments affecting the plot

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

See Also

show.

Examples

```
## Not run:

## Load data
data(transfers)

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

```
## Plot in- and outgoing contact chain for the root 2645
plot(contactTrace)

## End(Not run)
```

`Report` *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.

Usage

```
Report(object, format = c("html", "pdf"), dir = ".", template = NULL)

## S4 method for signature 'ContactTrace'
Report(object, format = c("html", "pdf"), dir = ".", template = NULL)

## S4 method for signature 'list'
Report(object, format = c("html", "pdf"), dir = ".", template = NULL)
```

Arguments

`object`	the object
`format`	the format to use, can be either 'html' or 'pdf'. The default is 'html'
`dir`	the generated report is written to the directory folder. The default (`"."`) is the current working directory.
`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.

References

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009
- Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Hardle and Bernd Ronz, editors, Compstat 2002 - Proceedings in Computational Statistics, pages 575-580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

See Also

Sweave, texi2pdf.

Examples

```
## Load data
data(transfers)

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

## Generate an html report showing details of the contact tracing for
## root 2646.
## Creates the file 2645.html in the temporary directory.
Report(contactTrace, dir = tempdir())

## 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
## Creates the files 1.html, 2.html, ..., 10.html
## in the temporary directory
Report(contactTrace, dir = tempdir())
```

ReportObject *Get current* ContactTrace *object when generating a 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. This method enables communication of the current ContactTrace object to the report.

Usage

```
ReportObject()
```

Value

The current ContactTrace object when generating a report

ShortestPaths ShortestPaths

Description

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

Usage

```
ShortestPaths(x, ...)

## S4 method for signature 'ContactTrace'
ShortestPaths(x)

## S4 method for signature 'data.frame'
ShortestPaths(
  x,
  root,
  tEnd = NULL,
  days = NULL,
  inBegin = NULL,
  inEnd = NULL,
  outBegin = NULL,
  outEnd = NULL
)
```

Arguments

x	a ContactTrace object, or a data.frame with movements of animals between holdings, see Trace for details.
...	Additional arguments to the method
root	vector of roots to calculate shortest path 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
outBegin	the first date to include outgoing movements. Defaults to NULL
outEnd	the last date to include outgoing movements. Defaults to NULL

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 contact at distance from root

destination The destination of the contact at distance from root

distance The shortest distance from/to root in the depth first search

Methods

signature(object = "ContactTrace") Get the shortest paths for the ingoing and outgoing Contacts of a ContactTrace object.

signature(x = "data.frame") Get the shortest paths for a data.frame with movements, see details and examples.

See Also

show and NetworkStructure.

Examples

```
## Not run:

## Load data
data(transfers)

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

ShortestPaths(contactTrace)

## Calculate shortest paths for all included herds
## First extract all source and destination from the dataset
root <- sort(unique(c(transfers$source, transfers$destination)))

sp <- ShortestPaths(transfers,
                    root = root,
                    tEnd = "2005-10-31",
                    days = 90)

## End(Not run)
```

`show-methods` *Show*

Description

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

Usage

```
## S4 method for signature 'Contacts'
show(object)
```

Arguments

`object`	The `Contacts` or `ContactTrace` `object`

Value

None (invisible 'NULL').

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

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

Examples

```
## Not run:

## Load data
data(transfers)

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

show(contactTrace)

## End(Not run)
```

Trace *Trace Contacts.*

Description

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

Usage

```
Trace(
  movements,
  root,
  tEnd = NULL,
  days = NULL,
  inBegin = NULL,
```

```
  inEnd = NULL,
  outBegin = NULL,
  outEnd = NULL,
  maxDistance = NULL
)
```

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
maxDistance	stop contact tracing at maxDistance (inclusive) from root. Default is NULL i.e. don't use the maxDistance stop criteria.

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

- Dube, C., et al., A review of network analysis terminology and its application to foot-and-mouth disease modelling and policy development. Transbound Emerg Dis 56 (2009) 73-85, doi: 10.1111/j.1865-1682.2008.01064.x
- Noremark, M., et al., Network analysis of cattle and pig movements in Sweden: Measures relevant for disease control and riskbased surveillance. Preventive Veterinary Medicine 99 (2011) 78-90, doi: 10.1016/j.prevetmed.2010.12.009

Examples

```
## 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
trace_1

## Create a network summary for 10 of the included herds
## First extract all source and destination from the dataset,
## then select the first ten.
root <- sort(unique(c(transfers$source,
                      transfers$destination)))
root <- root[1:10]

## Perform contact tracing using tEnd and days.
trace_3 <- Trace(movements = transfers,
                 root = root,
                 tEnd = "2005-10-31",
                 days = 91)

## Perform contact tracing using inBegin, inEnd
## outBegin and outEnd
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
identical(trace_3, trace_4)
```

`transfers` *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.

Usage

```
data(transfers)
```

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

```
data(transfers)

Trace(movements = transfers, root = 2645, tEnd = "2005-10-31", days = 90)
```

Index