Package 'o2geosocial'

October 14, 2022

Type Package

Title Reconstruction of Transmission Chains from Surveillance Data

Version 1.1.0

URL https://github.com/alxsrobert/o2geosocial

Encoding UTF-8

Maintainer Alexis Robert <alexis.robert@lshtm.ac.uk>

Description Bayesian reconstruction of who infected whom during past outbreaks using routinely-collected surveillance data. Inference of transmission trees using genotype, age specific social contacts, distance between cases and onset dates of the reported cases. (Robert A, Kucharski AJ, Gastanaduy PA, Paul P, Funk S. 2020 <doi:10.1098/rsif.2020.0084>).

License MIT + file LICENSE

Imports Rcpp, methods, stats, grDevices, geosphere, ggplot2, visNetwork, data.table, outbreaker2

LazyData true

Depends R(>= 3.5.0)

LinkingTo Rcpp

RoxygenNote 7.1.1

Suggests testthat, tigris, sf, knitr, socialmixr, rmarkdown

VignetteBuilder knitr

BugReports https://github.com/alxsrobert/o2geosocial/issues

NeedsCompilation yes

Author Alexis Robert [aut, cre, cph] (<https://orcid.org/0000-0002-4516-2965>), Sebastian Funk [aut] (<https://orcid.org/0000-0002-2842-3406>), Adam J Kucharski [aut] (<https://orcid.org/0000-0001-8814-9421>), Thibaut Jombart [ctb]

Repository CRAN

Date/Publication 2021-09-11 04:40:02 UTC

R topics documented:

`create_config` *Set and check parameter settings*

Description

This function defines settings. It takes a list of named items as input, performs various checks, set defaults where arguments are missing, and return a correct list of settings. If no input is given, it returns the default settings.

Usage

```
create_config(..., data = NULL)
```

Arguments

`...`	a list of config items to be processed (see description)
`data`	an optional list of data items as returned by `outbreaker_data`; if provided, this allows for further checks of the outbreaker settings.

Details

Acceptable arguments for ... are:

init_tree the tree used to initialize the MCMC. It can be a vector of integers corresponding to the tree itself, where the i-th value corresponds to the index of case `i`. Otherwise, it should be defined as the character string "star" and the function `create_config()` will generate the initial tree.

spatial_method a character string indicating the method used to evaluate the spatial likelihood. Can be either "exponential" or "power-law".

gamma a double indicating the spatial threshold for pre clustering; defaults to NULL.

delta a double indicating the temporal threshold for pre clustering; defaults to NULL.

init_kappa a vector of integers indicating the initial values of kappa; defaults to 1.

init_a initial value of the first spatial parameter (population).

init_b initial value of the second spatial parameter (distance).

init_alpha a vector of integers indicating the initial values of alpha, where the i-th value indicates the ancestor of case 'i'; defaults to `NULL`, in which ancestries are defined from `init_tree`.

init_t_inf a vector of integers indicating the initial values of `t_inf`, i.e. dates of infection; defaults to `NULL`, in which case the most likely `t_inf` will be determined from the delay to reporting/symptoms distribution, and the dates of reporting/symptoms, provided in `data`.

init_pi initial value for the reporting probability.

n_iter an integer indicating the number of iterations in the MCMC, including the burnin period.

move_alpha a vector of logicals indicating, for each case, if the ancestry should be estimated ('moved' in the MCMC), or not, defaulting to TRUE; the vector is recycled if needed.

move_t_inf a vector of logicals indicating, for each case, if the dates of infection should be estimated ('moved' in the MCMC), or not, defaulting to TRUE; the vector is recycled if needed.

move_pi a logical indicating whether the reporting probability should be estimated ('moved' in the MCMC), or not, all defaulting to TRUE.

move_kappa a logical indicating whether the number of generations between two successive cases should be estimated ('moved' in the MCMC), or not, all defaulting to TRUE.

move_a a logical indicating whether the first spatial parameter should be estimated ('moved' in the MCMC), or not, all defaulting to TRUE.

move_b a logical indicating whether the second spatial parameter should be estimated ('moved' in the MCMC), or not, all defaulting to TRUE.

move_swap_cases a logical indicating whether the movement to swap cases should be used, or not, all defaulting to TRUE.

sample_every the frequency at which MCMC samples are retained for the output.

sd_pi the standard deviation for the Normal proposal for the reporting probability.

sd_a the standard deviation for the Normal proposal for the first spatial parameter.

sd_b the standard deviation for the Normal proposal for the second spatial parameter.

find_import a logical indicating whether the import status of cases should be estimated.

outlier_threshold a numeric value indicating the probability that should be used to compute the threshold when estimating the import status.

outlier_relative a logical indicating whether the threshold is an absolute or relative value, default to FALSE (absolute value).

outlier_plot a logical indicating whether to plot the comparison between the likelihoods of connection in the short run and the threshold.

n_iter_import Number of iterations of the first short run.

sample_every_import the frequency at which MCMC samples are retained for the output during the first run.

burnin The number of iterations that should be removed when estimating import.

max_kappa an integer indicating the largest number of generations between any two linked cases; defaults to 5.

prior_pi a numeric vector of length 2 indicating the first and second parameter of the beta prior for the reporting probability 'pi'.

prior_a a numeric vector of length 2 indicating the first and second parameter of the uniform prior for the first spatial parameter 'a'.

prior_b a numeric vector of length 2 indicating the first and second parameter of the uniform prior for the second spatial parameter 'b'.

verbatim Logical, should the number of iteration be printed.

Value

A named list containing the value of each elements listed in the 'Details' section. This list describes the settings of the `outbreaker()` function. The class of this list is set to `outbreaker_config`.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

See Also

`outbreaker_data` to check and process data for outbreaker

Examples

```
## see default settings
create_config()

## change defaults
create_config(move_alpha = FALSE, n_iter = 2e5, sample_every = 1000)
```

---

create_param *Initializes outputs for outbreaker*

Description

This function creates initial outputs and parameter states for outbreaker.

Usage

```
create_param(data = outbreaker_data(), config = create_config())
```

Arguments

`data`	A list of data items as returned by `outbreaker_data`, or arguments passed to this function.
`config`	A list of settings as returned by `create_config`, or arguments passed to this function.

Value

A named list containing two components `$store` and `$current`. `store` is a list with the class `outbreaker_store`, used for storing 'saved' states of the MCMC. `current` is a list with the class `outbreaker_param`, used for storing 'current' states of the MCMC.

`outbreaker_store` class content:

- `size`The length of the list, corresponding to the number of samples saved from the MCMC.
- `step`A vector of integers of length `size`, storing the steps of the MCMC corresponding to the saved samples.
- `post`A numeric vector of length `size`, storing log-posterior values.
- `like`A numeric vector of length `size`, storing log-likelihood values.
- `prior`A numeric vector of length `size`, storing log-prior values.
- `alpha`A list of length `size`. Each item of the list is an integer vector of length `data$N`, storing indices (from 1 to N) of infectors for each case.
- `t_inf`A list of length `size`. Each item of the list is an integer vector of length `data$N`, storing dates of infections for each case.
- `kappa`A list of length `size`. Each item of the list is an integer vector of length `data$N`, storing the number of generations before the last sampled ancestor for each case.
- `pi`A numeric vector of length `size`, storing values of the reporting probability.
- `a`A numeric vector of length `size`, storing values of the first spatial parameter (population).
- `b`A numeric vector of length `size`, storing values of the second spatial parameter (distance).
- `counter`A counter used to keep track of the current iteration of the MCMC (used internally).

`outbreaker_param` class content:

- `alpha`An integer vector of length `data$N`, storing indices (from 1 to N) of infectors for each case.
- `t_inf`An integer vector of length `data$N`, storing dates of infections for each case.
- `kappa`An integer vector of length `data$N`, storing the number of generations before the last sampled ancestor for each case.
- `pi`The value of the reporting probability.
- `a`The value of the first spatial parameter (population).
- `b`The value of the second spatial parameter (distance).
- `log_s_dens`The spatial likelihood matrix, calculated at each step from a and b if move_a == TRUE or move_b == TRUE.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

Examples

```
## load data
data("toy_outbreak_short")
dt_cases <- toy_outbreak_short$cases
dt_cases <- dt_cases[order(dt_cases$Date), ]
dt_regions <- toy_outbreak_short$dt_regions
all_dist <- geosphere::distGeo(matrix(c(rep(dt_regions$long, nrow(dt_regions)),
                                        rep(dt_regions$lat, nrow(dt_regions))),
                                      ncol = 2),
                               matrix(c(rep(dt_regions$long, each = nrow(dt_regions)),
                                        rep(dt_regions$lat, each = nrow(dt_regions))),
                                      ncol = 2))

dist_mat <- matrix(all_dist/1000, nrow = nrow(dt_regions))
pop_vect <- dt_regions$population
names(pop_vect) <- rownames(dist_mat) <- colnames(dist_mat) <- dt_regions$region

data <- outbreaker_data(dates = dt_cases$Date, age_group = dt_cases$age_group,
                        region = dt_cases$Cens_tract, population = pop_vect,
                        distance = dist_mat)

## modify config settings
config <- create_config(move_alpha = FALSE, n_iter = 2e5, sample_every = 1000)

## create param object
param <- create_param(data = data, config = config)
```

`custom_likelihoods` *Customise likelihood functions for o2geosocial*

Description

This function is used to specify customised likelihood functions for o2geosocial Custom functions are specified as a named list or series of comma-separated, named arguments, indicating which log-likelihood component they compute. Values currently available are:

Usage

```
custom_likelihoods(...)

## S3 method for class 'custom_likelihoods'
print(x, ...)
```

Arguments

`...`	a named list of functions, each computing a log-likelihood component.
`x`	an `outbreaker_config` object as returned by `create_config`.

Details

- `timing_sampling`: the likelihood of sampling times; by default, the function `cpp_ll_timing_sampling` is used.
- `timing_infections`: the likelihood of infection times; by default, the function `cpp_ll_timing_infections` is used.
- `reporting`: the likelihood of the reporting process; by default, the function `cpp_ll_reporting` is used.
- `space`: the likelihood of spatial distances; by default, the function `cpp_ll_space` is used.
- `age`: the likelihood of the age contacts; by default, the function `cpp_ll_age` is used.

All log-likelihood functions should have the following arguments, in this order:

- `data`: a list of named items containing input data as returned by `outbreaker_data`
- `param`: a list of parameters with the class `create_param`

Value

A named list of functions with the class `custom_likelihood`, each implementing a customised log-likelihood components of outbreaker. Functions which are not customised will result in a NULL component.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

Examples

```
## specify a null model by disabling all likelihood components
f_null <- function(data, config = NULL, param, i) {
  return(0.0)
}


null_model <- custom_likelihoods(timing_sampling = f_null,
                                 timing_infections = f_null,
                                 reporting = f_null,
                                 space = f_null,
                                 age = f_null)

null_config <- list(find_import = FALSE,
                    n_iter = 200, gamma = 100, delta = 30,
                    sample_every = 1)

## load data
data("toy_outbreak_short")
dt_cases <- toy_outbreak_short$cases
dt_cases <- dt_cases[order(dt_cases$Date), ][1:15,]
dt_regions <- toy_outbreak_short$dt_regions
all_dist <- geosphere::distGeo(matrix(c(rep(dt_regions$long, nrow(dt_regions)),
```

```
                                        rep(dt_regions$lat, nrow(dt_regions))),
                                      ncol = 2),
                                matrix(c(rep(dt_regions$long, each = nrow(dt_regions)),
                                        rep(dt_regions$lat, each = nrow(dt_regions))),
                                      ncol = 2))

dist_mat <- matrix(all_dist/1000, nrow = nrow(dt_regions))
pop_vect <- dt_regions$population
names(pop_vect) <- rownames(dist_mat) <- colnames(dist_mat) <- dt_regions$region

data <- outbreaker_data(dates = dt_cases$Date, age_group = dt_cases$age_group,
                        region = dt_cases$Cens_tract, population = pop_vect,
                        distance = dist_mat)

res_null <- outbreaker(data = data,
                       config = null_config,
                       likelihoods = null_model)
```

`custom_moves` *Customise samplers for outbreaker*

Description

This function is used to specify customised movement functions (a.k.a. samplers) for outbreaker. Custom functions are specified as a named list or series of comma-separated, named arguments, indicating which type of movement they implement. Values currently available are:

Usage

```
custom_moves(...)
```

Arguments

`...`	A list or a series of named, comma-separated functions implementing movements of parameters or augmented data.

Details

- `pi`: movement of the reporting probability; by default, the function `cpp_move_pi` is used.
- `a`: movement of the first spatial parameter; by default, the function `cpp_move_a` is used.
- `b`: movement of the second spatial parameter; by default, the function `cpp_move_b` is used.
- `alpha`: movement of the transmission tree, by randomly proposing infectors in the pool of cases infected before; by default, the function `cpp_move_alpha` is used.
- `swap_cases`: movement of the transmission tree, by swapping infectors and infected cases; by default, the function `cpp_move_swap_cases` is used.
- `ancestors`: movement of the transmission tree, by changing the ancestors of the different trees in a cluster; by default, the function `cpp_move_ancestors` is used.

- t_inf: movement of the date of infection; by default, the function cpp_move_t_inf is used.
- kappa: movement of the number generations between cases; by default, the function cpp_move_kappa is used.

Movement functions must have an argument param, which is a list of parameters and augmented data of the class create_param. Each movement function will be enclosed with its other arguments, so that the resulting function will have a single argument 'param'. For non-standard movements (i.e. none of the names specified above), the closure will contain:

- data: a list of named items containing input data as returned by outbreaker_data
- config: a list of named items containing input data as returned by create_config
- likelihoods: a list of named custom likelihood functions as returned by custom_likelihoods
- priors: a list of named custom prior functions as returned by custom_priors

Value

A named list of movement functions with a single argument 'param', with class outbreaker_moves.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

---

custom_priors *Customise priors for outbreaker*

Description

Priors can be specified in several ways in o2geosocial (see details and examples). The most flexible way to specify a prior is to provide a prior function directly. This function must take an argument 'param', which is a list which contains all the states of the parameters and augmented data. See the documentation of create_param for more information.

Usage

```
custom_priors(...)

## S3 method for class 'custom_priors'
print(x, ...)
```

Arguments

...	A list or a series of named, comma-separated functions implementing priors. Each function must have a single argument, which corresponds to a 'outbreaker_param' list.
x	an outbreaker_config object as returned by create_config.

Details

There are three ways a user can specify priors:

1) Default: this is what happens when the 'config' has default values of prior parameters.
2) Customized parameters: in this case, the prior functions are the default ones from the package, but will use custom parameters, specified by the user through `create_config`.

3) Customized functions: in this case, prior functions themselves are specified by the user, through the '...' argument of 'custom_priors'. The requirements is that such functions must have either hard-coded parameters or enclosed values. They will take a single argument which is a list containing all model parameters with the class 'outbreaker_param'. ALL PRIORS functions are expected to return values on a LOG SCALE.

Priors currently used for the model are:

- `pi` (reporting probability): default function is a beta distribution implemented in `cpp_prior_pi`. New prior functions should use `x$pi` to refer to the current value of `pi`, assuming their argument is called `x`.
- `a` (first spatial parameter (population)): default function is a uniform distribution implemented in `cpp_prior_a`. New prior functions should use `x$a` to refer to the current value of `a`, assuming their argument is called `x`.
- `b` (second spatial parameter (distance)): default function is a uniform distribution implemented in `cpp_prior_b`. New prior functions should use `x$b` to refer to the current value of `b`, assuming their argument is called `x`.

Value

A named list of custom functions with class `custom_priors`. Values set to `NULL` will be ignored and default functions will be used instead.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

Examples

```
## SPECIFYING PRIOR PARAMETERS
## Default values: pi follows a beta distribution (parameters 10, 1),
## a and b follow a uniform distribution (parameters 0, 5)
default_config <- create_config()
## Use the variables prior_a, prior_b and prior_pi to change the parameters
## of the prior distributions can be
new_config <- create_config(prior_a = c(0,5), prior_b = c(0,5),
                        prior_pi = c(2, 1))


## SPECIFYING A NEW PRIOR FUNCTION
## Example: flat prior for pi between 0.5 and 1
```

```
f <- function(x) {ifelse(x$pi > 0.5, log(2), log(0))}
priors <- custom_priors(pi = f)
## test the new prior distribution
priors$pi(list(pi=1))
priors$pi(list(pi=.6))
priors$pi(list(pi=.2))
priors$pi(list(pi=.49))
```

`outbreaker` *outbreaker: main function for reconstructing disease outbreaks*

Description

The function `outbreaker` is the main function of the package. It runs processes various inputs (data, configuration settings, custom priors, likelihoods and movement functions) and explores the space of plausible transmission trees of a densely sampled outbreaks.

Usage

```
outbreaker(
  data = outbreaker_data(),
  config = create_config(),
  priors = custom_priors(),
  likelihoods = custom_likelihoods(),
  moves = custom_moves()
)
```

Arguments

`data`	a list of named items containing input data as returned by `outbreaker_data`
`config`	a set of settings as returned by `create_config`
`priors`	a set of log-prior functions as returned by `custom_priors`
`likelihoods`	a set of log-likelihood functions as returned by `custom_likelihoods`
`moves`	a set of movement functions as returned by `custom_moves`

Value

A data frame of `n_iter / sample_every` rows (as defined in the function`create_config()`). For each row, the data frame contains:

- `post`: The posterior value of the transmission tree at this iteration.
- `like`: The likelihood value of the transmission tree at this iteration.
- `post`: The posterior value of the transmission tree at this iteration.
- `a`: The estimate of the spatial parameter `a` at this iteration,

- a: The estimate of the spatial parameter b at this iteration,
- pi: The estimate of the conditional report ratio pi at this iteration,
- alpha_1 to alpha_N: The infector of each case at this iteration.
- t_inf_1 to t_inf_N: The infection date of each case at this iteration.
- kappa_1 to kappa_N: The number of generation between each case and their infector at this iteration.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

See Also

outbreaker_data to process input data, and create_config to process/set up parameters

- outbreaker_data: function to process input data
- create_config: function to create default and customise configuration settings
- custom_priors: function to specify customised prior functions
- custom_likelihoods: function to specify customised likelihoods functions
- custom_moves: function to create default and customise movement functions

Examples

```
## get data
data(toy_outbreak_short)

## run outbreaker
dt_cases <- toy_outbreak_short$cases
dt_cases <- dt_cases[order(dt_cases$Date), ]
dt_regions <- toy_outbreak_short$dt_regions
all_dist <- geosphere::distGeo(matrix(c(rep(dt_regions$long, nrow(dt_regions)),
                                        rep(dt_regions$lat, nrow(dt_regions))),
                                      ncol = 2),
                               matrix(c(rep(dt_regions$long, each = nrow(dt_regions)),
                                        rep(dt_regions$lat, each = nrow(dt_regions))),
                                      ncol = 2))

dist_mat <- matrix(all_dist/1000, nrow = nrow(dt_regions))
pop_vect <- dt_regions$population
names(pop_vect) <- rownames(dist_mat) <- colnames(dist_mat) <- dt_regions$region

data <- outbreaker_data(dates = dt_cases$Date, age_group = dt_cases$age_group,
                        region = dt_cases$Cens_tract, population = pop_vect,
                        distance = dist_mat, a_dens = toy_outbreak_short$age_contact,
                        f_dens = dgamma(x = 1:300, scale = 0.43, shape = 27),
                        w_dens = dnorm(x = 1:300, mean = 11.7, sd = 2.0))
out <- outbreaker(data = data, config = list(n_iter = 200, sample_every = 5,
                                             n_iter_import = 100, sample_every_import = 5,
```

```
                                                gamma = 100, delta = 30, burnin = 20))
plot(out)
```

`outbreaker_data` *Process input data for outbreaker*

Description

This function performs various checks on input data given to outbreaker. It takes a list of named items as input, performs various checks, set defaults where arguments are missing, and return a correct list of data input. If no input is given, it returns the default settings.

Usage

```
outbreaker_data(..., data = list(...))
```

Arguments

`...`	a list of data items to be processed (see description)
`data`	optionally, an existing list of data item as returned by `outbreaker_data`.

Details

Acceptable arguments for ... are:

dates a vector indicating the collection dates, provided either as integer numbers or in a usual date format such as `Date` or `POSIXct` format. By convention, zero will indicate the oldest date. Cases must be ordering by ascending onset date.

age_group a vector indicating the age group of the cases, provided as integer numbers. The value of age group corresponds to the position of this age group in a_dens.

region a vector indicating the region of the cases, provided as integer numbers or characters. If numeric, the value of the region corresponds to the position of the region in the distance matrix and the population vector. Otherwise, the value corresponds to the region and will be matched to the distance matrix and the population vector.

w_dens a vector of numeric values indicating the generation time distribution, reflecting the infectious potential of a case t = 1, 2, ... time steps after infection. By convention, it is assumed that newly infected patients cannot see new infections on the same time step. If not standardized, this distribution is rescaled to sum to 1.

f_dens similar to `w_dens`, except that this is the distribution of the incubation period, i_e. time interval between the reported onset date and the infection date.

a_dens a matrix of numeric values indicating the contact between age groups, reflecting on the infectious potential of a case for a given age group.

genotype a character vector showing the genotype in each case.

is_cluster an integer vector indicating which group of cases each case belongs to.

s_dens a matrix of numeric values indicating the initial value of the connectivity between region. Only needed if a and b are fixed in the model, otherwise NULL.

population a double vector indicating the population in every region considered in the study.

distance a double matrix indicating the distance between each region.

import a logical vector indicating whether each case is an import (TRUE) or not (FALSE).

Value

A named list containing the value of each elements listed in the 'Details' section. This list describes the data that will be used by the function `outbreaker()`.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

Examples

```
data("toy_outbreak_short")
dt_cases <- toy_outbreak_short$cases
dt_cases <- dt_cases[order(dt_cases$Date), ]
dt_regions <- toy_outbreak_short$dt_regions
all_dist <- geosphere::distGeo(matrix(c(rep(dt_regions$long, nrow(dt_regions)),
                                        rep(dt_regions$lat, nrow(dt_regions))),
                                      ncol = 2),
                               matrix(c(rep(dt_regions$long, each = nrow(dt_regions)),
                                        rep(dt_regions$lat, each = nrow(dt_regions))),
                                      ncol = 2))

dist_mat <- matrix(all_dist/1000, nrow = nrow(dt_regions))
pop_vect <- dt_regions$population
names(pop_vect) <- rownames(dist_mat) <- colnames(dist_mat) <- dt_regions$region
data <- outbreaker_data(dates = dt_cases$Date, age_group = dt_cases$age_group,
                        region = dt_cases$Cens_tract, population = pop_vect,
                        distance = dist_mat)
```

`pre_clustering` *Pre cluster cases in groups according using the genotypes and the arbitrary thresholds gamma (spatial) and delta (temporal).*

Description

This function updates the clusters and the initial tree in the lists data and config

Usage

```
pre_clustering(data, config)
```

Arguments

data	A list of data items as returned by outbreaker_data, or arguments passed to this function.
config	A list of settings as returned by create_config, or arguments passed to this function.

Value

A named list containing two components $data and $config. data data items as returned by outbreaker_data. config is a list of settings as returned by create_config.

Author(s)

Alexis Robert (<alexis.robert@lshtm.ac.uk>)

---

print.outbreaker_chains

Basic methods for processing outbreaker results

Description

Several methods are defined for instances of the class outbreaker_chains, returned by outbreaker, including: print, plot

Usage

```
## S3 method for class 'outbreaker_chains'
print(x, n_row = 3, n_col = 8, type = "chain", ...)

## S3 method for class 'outbreaker_chains'
plot(
  x,
  y = "post",
  type = c("trace", "hist", "density", "cluster", "alpha", "t_inf", "kappa", "network"),
  burnin = 0,
  min_support = 0.1,
  labels = NULL,
  group_cluster = NULL,
  ...
)

## S3 method for class 'outbreaker_chains'
summary(object, burnin = 0, group_cluster = NULL, ...)
```

Arguments

x	an outbreaker_chains object as returned by outbreaker.
n_row	the number of rows to display in head and tail; defaults to 3.
n_col	the number of columns to display; defaults to 8.
type	a character string indicating the kind of plot to be used (see details)
...	further arguments to be passed to other methods
y	a character string indicating which element of an outbreaker_chains object to plot
burnin	the number of iterations to be discarded as burnin
min_support	a number between 0 and 1 indicating the minimum support of ancestries to be plotted; only used if 'type' is 'network'
labels	a vector of length N indicating the case labels (must be provided in the same order used for dates of symptom onset)
group_cluster	a numeric vector indicating the breaks to aggregate the cluster size distribution.
object	an outbreaker_chains object as returned by outbreaker.

Details

type indicates the type of graphic to plot:

- trace to visualise MCMC traces for parameters or augmented data (plots the log-likelihood by default)
- hist to plot histograms of quantitative values
- density to plot kernel density estimations of quantitative values
- alpha to visualise the posterior frequency of ancestries
- network to visualise the transmission tree; note that this opens up an interactive plot and requires a web browser with Javascript enabled; the argument 'min_support' is useful to select only the most supported ancestries and avoid displaying too many links
- kappa to visualise the distributions generations between cases and their ancestor/infector
- cluster to visualise the cluster size distribution, grouped by the value in group_cluster

Value

The form of the value returned by plot depends on the type. If the type is set as network, plot returns a visNetwork object containing the details of the inferred transmission trees. Otherwise, it returns a ggplot object containing the elements of the plot.

The function summary returns a list containing 9 elements:

- stepcontains the first and last values of the iteration number; the interval between each iteration retained for the output (defined by the parameter sample_every in create_config), and the number of iterations in the output,
- postcontains the minimum, maximum, mean, median and quartiles of the posterior distribution.

- likecontains the minimum, maximum, mean, median and quartiles of the likelihood distribution.
- priorcontains the minimum, maximum, mean, median and quartiles of the prior distribution.
- picontains the minimum, maximum, mean, median and quartiles of the conditional report ratio.
- acontains the minimum, maximum, mean, median and quartiles of the spatial parameter a.
- bcontains the minimum, maximum, mean, median and quartiles of the spatial parameter b.
- treea data.frame that contains the most likely infector, the infection date, and the number of missing generations of each case. It also contains the support of the most likely branch (i.e. the proportion of iterations where the infector of a case is its most likely infector), and import, the proportion of iteration where the case was classified as an importation.
- clustera data frame listing the minimum, maximum, median, mean and quartile of the cluster size distribution.

Author(s)

Initial version by Thibaut Jombart, rewritten by Alexis Robert (<alexis.robert@lshtm.ac.uk>)

---

toy_outbreak_long *Simulated outbreaks*

Description

We generated two datasets used to illustrate o2geosocial. The first one (toy_outbreak_long) contains 1,940 cases, from simulated outbreaks nationwide between 2010 and 2017. The list contains the following:

Usage

```
toy_outbreak_long
```

Format

An object of class list of length 3.

Details

- $cases: A data table summarising the epidemiological features of the 1,940 cases. It contains the ID, State, onset date, genotype, county, age group, import status, cluster, generation and infector of the cases.
- $dt_regions: A data table containing the ID, population, longitude and latitude of each region. Should be used to compute the distance matrix, using the package geosphere.
- $age_contact: A matrix indicating the number of contacts between age groups

Author(s)

Alexis Robert <alexis.robert@lshtm.ac.uk>

Examples

```
data("toy_outbreak_long")
names(toy_outbreak_long)
toy_outbreak_long
```

---

toy_outbreak_short *Simulated outbreaks*

Description

Second dataset used to illustrate o2geosocial. (toy_outbreak_short) is a smaller data set (75 cases), spread across different Census tracks in Ohio (population and location of each region taken from https://www.census.gov/geographies/reference-files/2010/geo/2010-centers-population.html). The list contains the following:

Usage

```
toy_outbreak_short
```

Format

An object of class list of length 3.

Details

- $cases: A data table summarising the epidemiological features of the 75 cases. It contains the ID, state, onset date, genotype, Census tract, age group, import status, cluster, generation and infector of the cases.
- $dt_regions: A data table containing the ID, population, longitude and latitude of each region. Should be used to compute the distance matrix, using the package geosphere.
- $age_contact: A matrix indicating the number of contacts between age groups

Author(s)

Alexis Robert <alexis.robert@lshtm.ac.uk>

Examples

```
data("toy_outbreak_short")
names(toy_outbreak_short)
toy_outbreak_short
```

Index