Package ‘EpiILMCT’

January 20, 2025

Title Continuous Time Distance-Based and Network-Based Individual Level Models for Epidemics

Version 1.1.7

Date 2021-06-29

Author Waleed Almutiry [aut, cre],
Rob Deardon [aut, ths],
Vineetha Warriyar K. V. [ctb]

Maintainer Waleed Almutiry <wkmtierie@qu.edu.sa>

Description Provides tools for simulating from continuous-time individual level models of disease transmission, and carrying out infectious disease data analyses with the same models. The epidemic models considered are distance-based and/or contact network-based models within Susceptible-Infectious-Removed (SIR) or Susceptible-Infectious-Notified-Removed (SINR) compartmental frameworks. <doi:10.18637/jss.v098.i10>.

Depends graphics, stats, utils, coda, parallel, R (>= 3.5.0)

Imports methods, igraph

License GPL (>= 2)

URL https://github.com/waleedalmutiry/EpiILMCT/

NeedsCompilation yes

Repository CRAN

Date/Publication 2021-06-29 06:10:06 UTC

Contents

as.epidat *Continuous Time level information of an Epidemic*

Description

The function `as.epidat` is used to generate objects of class `"datagen"`. These objects contain the individual event history of an epidemic along with other individual level information. The output of this function provides information required to fit continuous time individual level models.

Usage

```
as.epidat(type, kerneltype, incub.time = NULL, inf.time, rem.time,
id.individual, location = NULL, network = NULL, network.type = NULL)
```

Arguments

`type` type of compartmental framework, with the choice of “SIR” for Susceptible-Infectious-Removed and “SINR” for Susceptible-Infectious-Notified-Removed.

`kerneltype` type of kernel function with choice of “distance” for a distance-based ILM, “network” for a contact network-based ILM and “both” for a combination of network- and distance-based ILM.

`incub.time` a vector of the incubation times of individuals. It is required when `type` = “SINR”.

`inf.time` a vector of the infection times of individuals.

`rem.time` a vector of the removal times of individuals.

`id.individual` a vector of the id number of individuals.

`location` a matrix of the XY coordinates of individuals. It is required when `kerneltype` set to either “distance” or “both”.

`network` a contact network matrix. It is required when `kerneltype` set to either “network” or “both”.

`network.type` type of the contact network model, see the argument `type` in `contactnet` function.

Details

The id.individual is used here to relate individuals to their corresponding row numbers of the location, and contact network, matrices. The elements of the incub.time, inf.time, and rem.time vectors must be in sync with those elements of the id.individual vector. Then, each individual will have an id number and corresponding event times. Any mistake in entering these vectors can seriously affect the use of other functions like epictmcmc and plot.datagen which can highly lead to incorrect results and conclusion.

Value

A class of “datagen” which has a list of type, kerneltype, a sorted matrix of the epidemic with respect to the infection times in ascending order, the XY coordinates of individuals, and/or the contact network.

See Also

epictmcmc, datagen, plot.datagen.

Examples

```
# Just for illustation:
# SIR distance-based ILM for a population of 10 individuals:
loc <- cbind(runif(10, 0, 10), runif(10,0,10))
net <- contactnet(type = "random", num.id = 10, location = loc, beta = 0.3)
infection <- c(2.5, 1, 0, 0, 0.5, 0, 2, 1.5, 0, 3)
removal <- c(3.5, 2, 0, 1, 1.5, 0, 3, 2.5, 0, 4)
id <- c(2, 1, 4, 7, 10, 3, 5, 9, 8, 6)
epi <- as.epidat(type = "SIR", kerneltype = "distance", inf.time = infection,
rem.time = removal, id.individual = id, location = loc)
epi
# using the data set generated by using the SINR network-based ILM:
data(NetworkDataSINR)
names(NetworkDataSINR)
netSINR<-as.epidat(type = "SINR", kerneltype = "network",
incub.time = NetworkDataSINR$epi[,4], inf.time = NetworkDataSINR$epi[,6],
rem.time = NetworkDataSINR$epi[,2], id.individual = NetworkDataSINR$epi[,1],
location  = NetworkDataSINR$loc, network = NetworkDataSINR$net,
network.type = "powerlaw")
names(netSINR)
plot(netSINR, plottype = "history")
netSIR<-as.epidat(type = "SIR", kerneltype = "network",
inf.time = NetworkDataSINR$epi[,6], rem.time = NetworkDataSINR$epi[,2],
id.individual = NetworkDataSINR$epi[,1], location  = NetworkDataSINR$loc,
network = NetworkDataSINR$net, network.type = "powerlaw")
names(netSIR)
plot(netSIR, plottype = "history")
```

`contactnet` *Generate binary adjacency matrix contact networks*

Description

This function allows the user to generate contact network with binary adjacency matrices from spatial (power-law or Cauchy) and random network models.

Usage

```
contactnet(type, num.id = NULL, location = NULL, beta, nu = NULL)
```

Arguments

`type` type of contact network model, with the choice of “powerlaw” for the power-law model, “Cauchy” for the Cauchy model, and “random” for randomly generated contact network from a Bernoulli distribution with parameter beta.

`num.id` the number of individuals (nodes). It has to be specified when `type` is set to “random”.

`location` a matrix of the XY coordinates of individuals. Required when `type` is set to “powerlaw” or “Cauchy”. Default is NULL.

`beta` spatial parameter of the spatial contact network model or probability parameter of the random network model (>0).

`nu` scale parameter of the power-law contact network model (>0). Default value is 1.

Details

The contact networks considered here are undirected, “unweighted” and have binary adjacency matrix, so that $c_{ij} = c_{ji}$ for $i \neq j; i, j = 1, .., N$, and each element of the contact network is defined as $c_{ij} = 1$ if a connection exist between individuals i and j, and 0 otherwise. We can either generate spatial networks which have connections more likely to be present between two individuals closed together than far apart, or random contact networks.

For spatial contact networks, there are two options of specifying the model for the probability of connections between individuals. We use a generalized version of the power-law contact network model of Bifolchi et al. (2013) for the first option, in which the probability of a connection between individuals i and j is given by:

$$p(c_{ij} = 1) = 1 - e^{-\nu(d_{ij}^{-\beta})}, \quad \nu, \beta > 0,$$

where d_{ij} is the Euclidean distance between individuals i and j; β is the spatial parameter; and ν is the scale parameter.

The second option is a Cauchy model as used in Jewell et al., (2009). The probability of a connection between individuals i and j is given by:

$$p(c_{ij} = 1) = 1 - e^{-\beta/(d_{ij}^2 + \beta^2)}, \quad \beta > 0,$$

where d_{ij} is the Euclidean distance between individual i and j; and β is the spatial parameter.

The final option is a random contact network in which the probability of a connection is generated from a Bernoulli distribution with probability equal to β.

Value

An object of class “contactnet” that has a list of:

location: A matrix of the XY coordinates of individuals.

contact.network: Binary contact network adjacency matrix.

type: The type of contact network model.

References

Jewell, C. P., Kypraios, T., Neal, P., and Roberts, G. O. (2009). Bayesian analysis for emerging infectious diseases. *Bayesian Analysis*, 4(3):465-496.

Bifolchi, N., Deardon, R., and Feng, Z. (2013). Spatial approximations of network-based individual level infectious disease models. *Spatial and Spatio-temporal Epidemiology*, 6:59-70.

Examples

```
set.seed(12345)
loc<- matrix(cbind(runif(50, 0, 10),runif(50, 0, 10)), ncol = 2, nrow = 50)
net1<- contactnet(type = "powerlaw", location = loc, beta = 1.5, nu = 0.5)
net2<- contactnet(type = "Cauchy", location = loc, beta = 0.5)
net3<- contactnet(type = "random", num.id = 50, beta = 0.08)
```

datagen *Generate epidemics from distance-based and/or network-based continuous-time ILMs.*

Description

Generate epidemics from distance-based and/or network-based ILMs.

Usage

```
datagen(type, kerneltype, kernelmatrix, distancekernel = NULL,

initialepi = NULL, tmax = NULL, suspar = NULL, transpar = NULL,

powersus = NULL, powertrans = NULL, kernel.par = NULL, spark = NULL,

gamma = NULL, delta, suscov = NULL, transcov = NULL)
```

Arguments

`type` type of compartmental framework, with the choice of “SIR” for Susceptible-Infectious-Removed and “SINR” for Susceptible-Infectious-Notified-Removed.

`kerneltype` type of kernel function with choice of “distance” for a distance-based ILM, “network” for a contact network-based ILM and “both” for a combination of network- and distance-based ILM.

`kernelmatrix` a matrix of the XY coordinates of individuals when `kerneltype` is set to “distance”. An object of class “contactnet”, or a matrix of incidence (0 or 1) or weights $[0,\infty)$ representing contact network when `kerneltype` is set to “network”. A list of two matrices (1: the XY coordinates of individuals, 2: contact network matrix or an object of class “contactnet”) when `kerneltype` is set to “both”.

`distancekernel` the spatial kernel type when `kerneltype` is set to “distance” or “both”. Choices are “powerlaw” for a power-law distance kernel or “Cauchy” for a Cauchy distance kernel.

`initialepi` a matrix of the initial n_{ini} infected individuals. When `type` is set to “SIR”, it is an $n_{ini} \times 4$ matrix representing the id numbers of the individuals, removal times, infectious periods, and infection times. When `type` is set to “SINR”, it is an $n_{ini} \times 6$ matrix representing the id numbers of the individuals, removal times, delay periods, notification times, incubation periods, and infection times. If set to the default = NULL, one randomly chosen individual is infected with infection time set to zero.

`tmax` the time at which the epidemic simulation ends. If not set the simulation is run until the epidemic dies out or all individuals are infected and then enter the removed state.

`suspar` parameter(s) for the susceptibility function (>0). Default value(s) is 1.

`transpar` parameter(s) for the transmissibility function (>0). Default value(s) is 1.

`powersus` the power parameter(s) of the susceptibility function (>0). Default value(s) is 1.

`powertrans` the power parameter(s) of the transmissibility function (>0). Default value(s) is 1.

`kernel.par` a scalar spatial parameter when `kerneltype` is set to “distance” (>0), or a vector of the spatial and network effect parameters when `kerneltype` is set to “both” (>0). Default = NULL means that it is not required, as used when the `kerneltype` is set to “network”.

`spark` spark parameter (>=0), representing random infections that are unexplained by other parts of the model. Default value is 0.

`gamma` the notification effect parameter for SINR model. The default value is 1.

`delta` a vector of the shape and rate parameters of the gamma-distributed infectious period (SIR) or a 2×2 matrix of the shape and rate parameters of the gamma-distributed incubation and delay periods (SINR). Note that an exponential distribution can be assigned as it is a special case of the gamma distribution where the shape parameter is equal to 1.

`suscov` covariate matrix of the susceptibility function, representing each covariate in a column. If intercept exists or in the case of no covariates, the first column is set to unity.

`transcov` covariate matrix of the transmissibility function, representing each covariate in a column. If intercept exists or in the case of no covariates, the first column is set to unity.

Details

We consider the simulation of epidemics from continuous-time ILMs in the following way. Each infected individual has an infection life history defined by their time of infection and the length of time spent in the infectious state. We are assuming the conditional intensity functions stay constant between events, so that the time to the next infected individual, given that the last infection occurred at time t, is given by the minimum $W_j \sim Exp(\lambda_{jt})$, where W_j represents the "waiting time" for susceptible individual j becoming infected.

Under both SIR and SINR ILMs option, an epidemic is simulated using the specified model starting with a randomly chosen infected individual k at time I_1, or as specified via the `initialepi` option. Under the SIR model, the rate of infection of each susceptible individual j at time $t \in \mathbb{R}^+$ follows the framework of Deardon et al. (2010):

$$\lambda_{jt} = \left[\Omega_S(j) \sum_{i \in \mathcal{I}(t)} \Omega_T(i)\kappa(i,j)\right] + \epsilon(j,t),$$

where $\mathcal{I}(t)$ is the set of infectious individual at time t. Under the SINR model, the rate of infection of each susceptible individual j at time t follows Jewell et al. (2009):

$$\lambda_{jt} = \left[\Omega_S(j) \sum_{i \in \mathcal{N}^-(t)} \Omega_T(i)\kappa(i,j) + \gamma \left(\Omega_S(j) \sum_{i \in \mathcal{N}^+(t)} \Omega_T(i)\kappa(i,j)\right)\right] + \epsilon(j,t),$$

where $\mathcal{N}^-(t)$ is the set of infectious individual at time t who have been infected but have not reached the notified state; $\mathcal{N}^+(t)$ is the set of infectious individual at time t who have been infected and reached the notified state; γ is the notification effect parameter usually allowing for infectious risk reduction after notification (Jewell et al., 2009); $\Omega_S(j)$ is a function of risk factors associated with susceptible individual j (i.e., susceptibility); and $\Omega_T(i)$ is a function of risk factors associated with infectious individual i (i.e., transmissibility). The $\Omega_S(j)$ and $\Omega_T(i)$ are defined as:

$$\Omega_S(j) = \mathbf{S}\mathbf{X}_j^{\phi}; \text{ and } \Omega_T(i) = \mathbf{T}\mathbf{Z}_i^{\xi},$$

where $\mathbf{S}$ and $\mathbf{T}$ are the parameters vectors of the susceptibility and transmissibility functions, respectively; $\mathbf{X}$ and $\mathbf{Z}$ are the susceptibility and transmissibility risk factor matrices, respectively; and ϕ and ξ are vectors of the power parameters of the susceptibility and transmissibility functions, respectively. These power parameters allow for nonlinear relationships between the susceptibility and transmissibility risk factors and the infection rate (Deardon et al., 2010).

The infection kernel $\kappa(i,j)$ allows for distance (spatial) and/or contact network-based mechanisms. In distance-based models, the kernel function $\kappa(i,j)$ is a power-law or Cauchy function of distance. In network-based models, the kernel function $\kappa(i,j)$ is the contact network elements (binary (c_{ij}) or weighted (w_{ij})). In both distance-network-based models, the kernel function $\kappa(i,j)$ is represented by two terms: (1) spatial function, and (2) the contact network.

The optional kernel function formulae are as follows:

- **Distance-based ILMs**:
  - Power-law spatial kernel:
    $\kappa(i,j) = d_{ij}^{-\beta}$
  - Cauchy spatial kernel:
    $\kappa(i,j) = \frac{\beta}{d_{ij}^2+\beta^2}$
- **Network-based ILMs**:
  - $\kappa(i,j) = c_{ij}$ for binary contact network or,
  - $\kappa(i,j) = w_{ij}$ for weighted contact network.
- **Both distance- and network-based ILMs**:
  - With power-law spatial kernel:
    - $\kappa(i,j) = d_{ij}^{-\beta_1} + \beta_2 c_{ij}$ with binary contact network or,
    - $\kappa(i,j) = d_{ij}^{-\beta_1} + \beta_2 w_{ij}$ with weighted contact network.
  - With Cauchy spatial kernel:
    - $\kappa(i,j) = \frac{\beta_1}{(d_{ij}^2+\beta_1^2)} + \beta_2 c_{ij}$ with binary contact network or,
    - $\kappa(i,j) = \frac{\beta_1}{(d_{ij}^2+\beta_1^2)} + \beta_2 w_{ij}$ with weighted contact network.

At time I_s, the waiting time until being infected is drawn for each susceptible individual j from $W_j \sim Exp(\lambda_{jI_s})$. The individual with minimum W is chosen as the next infected individual, and assigned infection time $I_{s+1} = I_s + min(W_j)$. Under SIR model, this newly-infected individual is also assigned an infectious period D_j generated from the infectious period (exponential or gamma) distribution $f(D_j;\delta)$. Thus, its removal time becomes $R_{s+1} = I_{s+1} + D_j$. Under SINR model, each infected individual is assigned an incubation period defining the time from infection to notification, and a delay period defining the time from notification to removal. Thus, this newly-infected individual is assigned an incubation and delay period $D_j^{(inc)}$ and $D_j^{(delay)}$, generated from the (exponential or gamma) incubation and delay periods distributions $f(D_j^{(inc)};\delta^{(inc)})$ and $f(D_j^{(delay)};\delta^{(delay)})$, respectively. Thus, its notification and removal times become $N_{s+1} = I_{s+1} + D_j^{(inc)}$ and $R_{s+1} = N_{s+1} + D_j^{(delay)}$, respectively. The process is repeated until no infectives remain in the population, or until $I_{s+1} >$ `tmax` if it is specified.

Value

The function `datagen` returns objects of class “datagen” containing the following:

type: The specified type of the compartmental framework.

kerneltype: The specified type of the kernel function.

epidat: Under an SIR ILM, it is a matrix with four columns representing the id numbers of the individuals, removal times, infectious periods, and infection times, respectively. Under an SINR ILM, it is a matrix with six columns: the id numbers of the individuals, removal times, delay periods, notification times, incubation periods, and infection times. Uninfected individuals are assigned infinity values (∞) for both their removal and infection times. Note that the rows of `epidat` are sorted according to the infection times in ascending order. The id numbers of the individuals are used here to represent the corresponding row number of the `location`, contact `network`, and covariate, matrices.

location: A matrix of the XY coordinates of individuals if it is given in the function, otherwise, a zero matrix.

network: The contact network matrix if it is given in the function, otherwise, a zero matrix.

References

Jewell, C. P., Kypraios, T., Neal, P., and Roberts, G. O. (2009). Bayesian analysis for emerging infectious diseases. *Bayesian Analysis*, 4(3):465-496.

Deardon, R., Brooks, S. P., Grenfell, B. T., Keeling, M. J., Tildesley, M. J., Savill, N. J., Shaw, D. J., and Woolhouse, M. E. (2010). Inference for individual-level models of infectious diseases in large populations. *Statistica Sinica*, 20(1):239.

See Also

`loglikelihoodepiILM`, `contactnet`, `plot.datagen`.

Examples

```
set.seed(3245)
loc<- matrix(cbind(runif(100, 1, 100),runif(100, 1, 100)), ncol = 2, nrow = 100)
net1<- contactnet(type = "powerlaw", location = loc, beta = 0.9)
cov <- cbind(rep(1, 100), runif(100, 1, 50))
#To simulate the epidemic using the network kernel:
epi.net <- datagen(type = "SIR", kerneltype = "network",
               kernelmatrix = net1, suspar = c(0.08, 0.05),
               delta = c(4, 2), suscov = cov)
#To simulate the epidemic using the distance kernel:
epi.dis <- datagen(type = "SIR", kerneltype = "distance",
               distancekernel = "powerlaw", kernelmatrix = loc,
               suspar = c(0.8, 0.5), kernel.par = 2,
               delta = c(4, 2), suscov = cov)
```

`epictmcmc` *Markov Chain Monte Carlo-based tool for analyzing (SIR/SINR) distance-based/network-based individual-level disease data.*

Description

Runs a Bayesian data augmented MCMC algorithm for fitting specified models (SIR or SINR) with three possible assumptions regarding the data set: 1. a completely observed epidemic; 2. unknown infection times but known removal times; or 3. unknown removal and infection times but known notification times (SINR only).

Usage

```
epictmcmc(object, distancekernel = NULL, datatype, blockupdate = NULL, nsim,

nchains = NULL, control.sus = NULL, control.trans = NULL, kernel.par = NULL,

spark.par = NULL, delta = NULL, gamma.par = NULL, periodproposal = NULL,

parallel = FALSE)
```

Arguments

object
an object of class “datagen” that can be the output of datagen or as.epidat functions.

distancekernel
the spatial kernel type when kerneltype is set to “distance” or “both”. Choices are “powerlaw” for a power-law distance kernel or “Cauchy” for a Cauchy distance kernel.

datatype
the ILM data set is analyzed based on three possible assumptions. The possible assumptions are “known epidemic” for a completely observed epidemic; “known removal” times for observing only the removal times for the SIR model, or removal and notified times for the SINR model; or “unknown removal” times for observing only the notified times which is applicable only when fitting the SINR model.

blockupdate
a vector of the number of initially observed infection times and the size of blocks for updating the infection times/infectious periods (SIR model), or infection times/incubation periods and removal times/delay periods (SINR model). The default is observing only the first infection time and performing the MCMC updating infection time parameters via single parameter updates.

nsim
the number of MCMC iterations to be carried out for a single chain.

nchains
the number of MCMC chains to be run. Default is 1 (a single MCMC chain).

control.sus
a list of arguments for updating the parameters of the susceptibility function (>0):

1st: the susceptibility parameters (list):

- a vector (of length n_s) or a $n_s \times$ nchains matrix of initial values of the susceptibility parameters,
- a $n_s \times 4$ matrix of the required arguments for updating the susceptibility parameters,

2nd: an $n \times n_s$ matrix of the susceptibility covariates.

3rd: the power parameters of the susceptibility function (list):

- a vector (of length n_s) or a $n_s \times$ nchains matrix of initial values of the power parameters,
- a $n_s \times 4$ matrix of the required arguments for updating the power parameters,

where n and n_s are the number of individuals and number of susceptibility parameters, respectively. Default = NULL means the model does not include these parameters. See details for a description of required arguments.

`control.trans` a list of arguments with the same structure as `control.sus`, but for the transmissibility function (>0).

`kernel.par` a list of arguments for updating the spatial parameter of the distance-based kernel. The first argument is a vector of initial values of the spatial parameter when `kerneltype` is set to “distance” (>0) or a $2 \times$ `nchains` matrix of initial values of the spatial and network effect parameters when `kerneltype` is set to “both” (>0). The second argument is a vector of arguments for updating the spatial parameter of the distance-based kernel when `kerneltype` is set to “distance” or a 2×4 matrix of arguments for updating the spatial and network effect parameters when `kerneltype` is set to “both”. Default = NULL means the model does not include these parameters (i.e., we are fitting a network-based ILM). See details for a description of required arguments.

`spark.par` a list of arguments for updating the spark parameter (>=0). The first argument is a vector of the initial values. The second argument is a vector of the required arguments for updating the spark parameter. It should be defined if `kerneltype` is set to “network” with `datatype`= “known removal” or “unknown removal”. Default = NULL means the model does not include this parameter. See details for a description of required arguments.

`delta` a list of arguments for updating the infectious period rate for an SIR model, or for updating the incubation and delay periods rates for an SINR model (>0). The arguments for updating each rate are a vector of the fixed and known shape parameter(s) for the distribution of the period type, a vector (matrix) of initial values of the rate(s) of the infectious, incubation and/or delay period distributions, and a vector (matrix) of the shape and rate parameters of the gamma prior distribution for the rate parameter(s). Default = NULL means these parameters do not need to be updated, which is the case when `datatype` is set to “known epidemic”.

`gamma.par` a list of arguments for updating the notification effect parameter (SINR) (>=0). The first argument is a vector of initial values. The second argument is a vector of the required arguments for updating the notification effect parameter. Default = NULL means the parameter is not updated and is assigned a value equal to 1. See details for a description of required arguments.

`periodproposal` a vector/matrix of the proposal distribution parameters of the independence sampler for updating the infectious period (SIR model), or the incubation and/or delay periods (SINR model). Here, we use a gamma proposal distribution. It is required when `datatype` is set to “known removal” or “unknown removal”. Default = NULL means the parameters of the gamma proposal distribution are the known shape and updated rate values based on the argument `delta`.

`parallel` if set to “TRUE”, multiple MCMC chains are running in parallel. Default is “FALSE” means chains (>1) are run sequentially.

Details

Bayesian MCMC is performed to obtain posterior estimates of the model parameters and latent variables. When the `datatype` is set to “known removal” or “unknown removal”, we assume the infectious periods (SIR model) or the incubation and/or delay periods (SINR model) follow gamma distributions with fixed shape and unknown rate parameters. We assign a gamma prior distribution for the rate parameter with shape a and rate b. This leads the rate parameter(s) to have a gamma conditional distribution.

Under the SIR model, the conditional distribution is:

$$\delta|\theta, \boldsymbol{I}, \boldsymbol{R} \sim \Gamma(m + a_{\delta}, M + b_{\delta}),$$

where δ is the rate of the infectious period distribution; m is the number of infected individuals; $M = \sum_{i=1}^{m} (R_i - I_i)$; and a_{δ} and b_{δ} are the prior parameters for the infectious period rate, respectively.

Under the SINR model, the conditional distribution is:

$$\delta^{(inc)}|\theta, \boldsymbol{I}, \boldsymbol{N}, \boldsymbol{R} \sim \Gamma(m + a_{\delta^{(inc)}}, M_{inc} + b_{\delta^{(inc)}}),$$

where $\delta^{(inc)}$ is the rate of the incubation period distribution; $M_{inc} = \sum_{i=1}^{m} (N_i - I_i)$; and $a_{\delta^{(inc)}}$ and $b_{\delta^{(inc)}}$ are the prior parameters for the incubation period rate, respectively; and

$$\delta^{(delay)}|\theta, \boldsymbol{I}, \boldsymbol{N}, \boldsymbol{R} \sim \Gamma(m + a_{\delta^{(delay)}}, M_{delay} + b_{\delta^{(delay)}}),$$

where $\delta^{(delay)}$ is the rate parameter of the delay period distribution; $M_{delay} = \sum_{i=1}^{m} (R_i - N_i)$; and $a_{\delta^{(delay)}}$ and $b_{\delta^{(delay)}}$ are the prior parameters for the delay period rate, respectively.

A Gibbs update (i.e., sampling from the conditional posterior distribution) is then used for the infectious period (SIR model) or the incubation and/or delay period rates (SINR model). The required information for each period distribution are entered via the `delta` argument. We assume each period type follows a gamma distribution with fixed shape and unknown rate. Thus, to update the rate parameter of each period, `delta` is a list containing a vector of the fixed shape value(s), a vector (matrix) of the initial values of the rate(s), and a vector (matrix) of the parameters of the prior distribution of the rate parameter(s). In the case of incubation and delay periods being estimated, the input of the initial values is a 2 × `nchains` matrix, and the prior parameters are given by a 2 × 2 matrix in which each row contains the required information for each period rate.

An independence sampler is used for updating the infection times and infectious periods (SIR model), or the infection and/or removal times, and the incubation and/or delay periods (SINR model). Under the SIR model, we update each infection time I_i by proposing infectious period D_i^* from a proposal distribution with tuning parameter, such that $D_i \sim$ f(.). Then, the new infection time is just the difference between the fixed known removal time and the new infectious period of the i^{th} individual. Each infection time/infectious period is updated in this way in turn. The same procedure is applied for updating the missing event times and incubation and delay periods for the SINR model, with their corresponding parameters.

The `control.sus` and `control.trans` arguments provide all the information needed for the susceptibility and transmissibility functions. These arguments must be defined as a list of three objects. The first object is a list of: 1) a vector or matrix of initial values; and 2) a vector or matrix of the following arguments in order: initial value, prior distribution ("gamma", (positive) "half normal" or "uniform"), a vector of the prior distribution parameters, and the proposal variance. The second object is a matrix of the susceptibility or transmissibility risk factors. The third contains the power parameters of the susceptibility or transmissibility functions and has the same structure as the first object. The other model parameters `kernel.par`, `spark.par` and `gamma.par` also have the same structure as the first object. These model parameters, along with the model parameters of the susceptibility and transmissibility functions (`control.sus` and `control.trans`), are updated in turn using a random-walk Metropolis Hastings algorithm with normal proposals. The normal proposals must be tuned, via the proposal variance, by the user to achieve good mixing properties.

Note that, setting the variance of the normal proposal distribution to zero fixes a parameter at its initial value. This option allows the user to fix such a parameter in the MCMC procedure while updating others.

For faster run of the `epictmcmc` function, an option of running multiple chains on parallel can be performed, controlled via the two options `nchains` and `parallel`. If `parallel` is set to TRUE, the number of required cores is set to the minimum of `nchains` and the number of available cores on the user's computer.

Value

Returns an object of class `epictmcmc` that contains:

compart.framework: the compartmental framework model used in the analysis.

kernel.type: the used `kernel.type` in the function.

data.assumption: the `datatype`.

parameter.samples: the MCMC output of the updated model parameters.

log.posterior: the log posterior densities.

acceptance.rate: the acceptance rates for MCMC updates of each parameter.

number.iteration: the number of iterations.

number.parameter: the number of the unknown model parameters.

number.chains: the number of the MCMC chains.

infection.times.samples: the updated infection times when `datatype` is set to "known removal".

Average.infectious.periods: the average infectious period when `type` is set to "SIR" and `datatype` is set to "known removal".

removal.times.samples: the updated removal times when `datatype` is set to "unknown removal".

Average.incubation.periods: the average incubation period when `type` is set to "SINR" and `datatype` is set to either "known removal" or "unknown removal".

Average.delay.periods: the average delay period when `type` is set to "SINR" and `datatype` is set to "unknown removal".

See Also

`print.epictmcmc`, `summary.epictmcmc`, `plot.epictmcmc`, `datagen`, `loglikelihoodepiILM`.

Examples

```
## This is for testing; the number of MCMC iterations needs to be
## mucher higher in practice.

## distance-based SIR continuous-time ILMs ##
data(SpatialData)
## performing the MCMC-tool for analyzing the fully observed spatial data
## under the SIR distance-based continuous ILM:
```

```
suspar <- list(NULL)
suspar[[1]] <- list(2, c("gamma", 1, 0.01, 0.5))
suspar[[2]] <- rep(1, length(SpatialData$epidat[,1]))
kernel1 <- list(2, c("gamma", 1, 0.01, 0.5))
mcmcres2 <- epictmcmc(object = SpatialData,
distancekernel = "powerlaw", datatype = "known epidemic", nsim = 50,
control.sus = suspar, kernel.par = kernel1)
summary(mcmcres2)

## performing the MCMC-tool for analyzing the partially observed spatial
## data (unknown infection times) under the SIR distance-based
## continuous ILM:
suspar <- list(NULL)
suspar[[1]]<-list(2,c("gamma", 1, 0.01, 0.8))
suspar[[2]]<- rep(1,length(SpatialData$epidat[,1]))
kernel1 <- list(2, c("gamma", 1, 0.01, 0.5))
mcmcres22 <- epictmcmc(object = SpatialData,
distancekernel = "powerlaw", datatype = "known removal", nsim = 50,
control.sus = suspar, kernel.par = kernel1, delta = list(1,2,c(4,2)))
summary(mcmcres22)

## distance-based and network-based SIR ILMs ##
data(SpatialNetData)
## performing the MCMC-tool for analyzing the fully observed spatial and
## network data under the SIR distance-based and network-based
## continuous-time ILM:
suspar <- list(NULL)
suspar[[1]] <- list(NULL)
suspar[[1]][[1]] <- c(0.08, 0.2)
suspar[[1]][[2]] <- matrix(c("gamma", "gamma", 1, 1, 0.01, 0.01, 0.1, 0.5),
ncol = 4, nrow = 2)
suspar[[2]] <- SpatialNetData[[2]]
kernel1 <- list(NULL)
kernel1[[1]] <- c(5, 0.5)
kernel1[[2]] <- matrix(c("gamma", "gamma", 1, 1, 0.01, 0.01, 0.5, 1),
ncol = 4, nrow = 2)
mcmcres3 <- epictmcmc(object = SpatialNetData[[1]],
distancekernel = "powerlaw", datatype = "known epidemic", nsim = 50,
control.sus = suspar, kernel.par = kernel1)
summary(mcmcres3)

## network-based SIR ILMs ##
data(NetworkData)
## performing the MCMC-tool for analyzing the fully observed network data
## under the SIR network-based continuous-time ILM:
suspar <- list(NULL)
suspar[[1]] <- list(NULL)
suspar[[1]][[1]] <- c(0.08, 0.5)
suspar[[1]][[2]] <- matrix(c("gamma", "gamma", 1, 1, 1, 1, 0.1, 0.5),
ncol = 4, nrow = 2)
suspar[[2]] <- NetworkData[[2]]
mcmcres4 <- epictmcmc(object = NetworkData[[1]], datatype = "known epidemic",
nsim = 50, control.sus = suspar)
```

```
summary(mcmcres4)
```

loglikelihoodepiILM *Calculates the log likelihood*

Description

Calculates the log likelihood for the specific compartmental framework of the continuous-time ILMs.

Usage

```
loglikelihoodepiILM(object, distancekernel = NULL, control.sus = NULL,

control.trans = NULL, kernel.par = NULL, spark = NULL, gamma = NULL,

delta = NULL)
```

Arguments

`object`	an object of class “datagen” that can be the output of `datagen` or `as.epidat` functions.
`distancekernel`	the spatial kernel type when `kerneltype` is set to “distance” or “both”. Choices are “powerlaw” for a power-law distance kernel or “Cauchy” for a Cauchy distance kernel.
`control.sus`	a list of values of the susceptibility function (>0): **1st:** a vector of values of the susceptibility parameters, **2nd:** an $n \times n_s$ matrix of the susceptibility covariates, **3rd:** a vector of values of the power parameters of the susceptibility function, where, n and n_s are the number of individuals and number of susceptibility parameters, respectively. Default = NULL means the model does not include these parameters.
`control.trans`	it has the same structure as the `control.sus`, but for the transmissibility function (>0).
`kernel.par`	a scalar spatial parameter for the distance-based kernel (>0), or a vector of the spatial and network effect parameters of the network and distance-based kernel (both). It is not required when the `kerneltype` is set to “network”.
`spark`	spark parameter (>=0), representing random infections that are unexplained by other parts of the model. Default value is zero.
`gamma`	the notification effect parameter for SINR model. The default value is 1.
`delta`	a vector of the shape and rate parameters of the gamma-distributed infectious period (SIR) or a 2 × 2 matrix of the shape and rate parameters of the gamma-distributed incubation and delay periods (SINR).

Details

We label the m infected individuals $i = 1, 2, \ldots, m$ corresponding to their infection (I_i) and removal (R_i) times; whereas the $N - m$ individuals who remain uninfected are labeled $i = m + 1, m + 2, \ldots, N$ with $I_i = R_i = \infty$. We then denote infection and removal time vectors for the population as $\boldsymbol{I} = \{I_1, \ldots, I_m\}$ and $\boldsymbol{R} = \{R_1, \ldots, R_m\}$, respectively. We assume that infectious periods follow a gamma distribution with shape and rate δ. The likelihood of the general SIR continuous-time ILMs is then given as follows:

$$\begin{array}{rcl} L(I, R|\theta) & = & \prod_{j=2}^{m} \left(\epsilon + \sum_{i:I_i<I_j\leq R_i} \lambda_{ij}^{-}\right) \\ & \times & exp\left\{-\sum_{i=1}^{m} \left(\sum_{j=1}^{N} ((R_i \wedge I_j) - (I_i \wedge I_j))\lambda_{ij}^{-}\right)\right\} \\ & \times & exp\left(-\epsilon \sum_{i=1}^{N} [(t_{obs} \wedge I_i) - I_1]\right) \prod_{i=1}^{m} f(D_i; \delta) \end{array}$$

where θ is the vector of unknown parameters; f(.;δ) indicates the density of the infectious period distribution; and D_i is the infectious period of infected individual i defined as $D_i = R_i - I_i$. The likelihood of the general SINR continuous-time ILMs is given by:

$$\begin{array}{rcl} L(I, N, R|\theta) & = & \prod_{j=2}^{m} \left(\epsilon + \sum_{i:I_i<I_j\leq N_i} \lambda_{ij}^{-} + \sum_{i:N_i<I_j\leq R_i} \lambda_{ij}^{+}\right) \\ & \times & exp\left\{-\sum_{i=1}^{n_I} \left(\sum_{j=1}^{N} ((t_{obs} \wedge N_i \wedge I_j) - (I_i \wedge I_j))\lambda_{ij}^{-}\right)\right\} \\ & \times & exp\left\{-\sum_{i=1}^{n_N} \left(\sum_{j=1}^{N} ((t_{obs} \wedge R_i \wedge I_j) - (I_i \wedge I_j)) - ((t_{obs} \wedge N_i \wedge I_j) - (I_i \wedge I_j))\lambda_{ij}^{+}\right)\right\} \\ & \times & exp\left(-\epsilon \sum_{i=1}^{N} [(t_{obs} \wedge I_i) - I_1]\right) \\ & \times & \prod_{i=1}^{m} f(D_i^{(inc)}; \delta^{(inc)}) \prod_{i=1}^{m} f(D_i^{(delay)}; \delta^{(delay)}) \end{array}$$

where D_i^{inc} and D_i^{delay} are the incubation and delay periods such that $D_i^{inc} = N_i - I_i$ and $D_i^{delay} = R_i - N_i$, and

$$\lambda_{ij}^{-} = \Omega_S(j)\Omega_T(i)\kappa(i, j),$$

for $i \in I(t), j \in S(t)$, and

$$\lambda_{ij}^{+} = \gamma(\Omega_S(j)\Omega_T(i)\kappa(i, j)),$$

for $i \in N(t), j \in S(t)$.

Note, λ_{ij}^{+} is used only under the SINR model.

Value

Returns the log likelihood value.

See Also

contactnet, datagen, epictmcmc.

`NetworkData` *Simulated epidemic data set from a network-based SIR ILM*

Description

This is a simulated epidemic data set of population size = 50 individuals that was generated using a network-based SIR individual-level model (ILM) with a contact network that was generated using the power-law model with parameters $\beta = 1.8$ and $\alpha = 1$. The model has one binary susceptible covariate and the infectivity rate is given by:

$$\lambda_{jt} = (\alpha_0 + \alpha_1 z_j) \sum_{i \in I_t} c_{ij}$$

The infectious period follows a gamma distribution $\Gamma(4, \delta)$. The epidemic was simulated with the following parameter values: $\alpha_0 = 0.08$, $\alpha_1 = 0.5$ and $\delta = 2$.

The data set file is a list of an object of class "datagen" that contains of `type`, `kerneltype`, `epidat`, `location` and `network`, and the covariate matrix.

Usage

```
data(NetworkData)
```

Format

It is a list of an object of class “datagen” that contains the following:

type: The “SIR” compartmental framework.

kerneltype: The “network” kernel function.

epidat: A matrix of the simulated epidemic with four columns as: the id numbers of individuals, removal times, infectious periods, and infection times.

location: A matrix of the XY coordinates of individuals.

network: The undirected binary contact network matrix.

and a 50×2 matrix of the covariates represents the unity intercept and the binary covariate z.

NetworkDataSINR *Simulated epidemic data set from a network-based SINR ILM*

Description

This is a simulated epidemic data set of population size = 50 individuals that was generated using a network-based SINR individual-level model (ILM) with a contact network that was generated using the power-law model with parameters $\beta = 1.8$ and $\alpha = 1$. The model has one binary susceptible covariate and the infectivity rate is given by:

$$\lambda_{jt} = (\alpha_0 + \alpha_1 z_j)[\sum_{i:\in I_t} c_{ij} + \gamma \sum_{i:\in N_t} c_{ij}]$$

The incubation period is assumed to follow a gamma distribution $\Gamma(4, \delta^{(inc)})$, and the delay period is assumed to follow an exponential distribution $\delta^{(delay)}$. The epidemic was simulated with the following parameter values: $\alpha_0 = 0.08$, $\alpha_1 = 2$, $\gamma = 1$, $\delta^{(inc)} = 2$ and $\delta^{(delay)} = 2$.

The data set file contains the XY coordinates of individuals, the contact network matrix, the covariate matrix and the simulated epidemic data.

Usage

```
data(NetworkDataSINR)
```

Format

A list of the simulated data:

loc: The XY coordinates of individuals.

net: The undirected binary contact network matrix.

cov: A matrix of two columns represents the unity intercept and the binary covariate z.

epi: The simulated epidemic with six columns as: the id numbers of individuals, removal times, delay periods, notification times, incubation periods, and infection times.

plot.contactnet *Plot the contact network of* contactnet *object*

Description

plot.contactnet is an S3 method that plots the contact network of an S3 object of class contactnet.

Usage

```
## S3 method for class 'contactnet'
plot(x, ...)
```

Arguments

x	an S3 object of class contactnet (i.e. the output of the contactnet function).
...	additional arguments that are passed to the generic plot function.

Value

Plot of the contact network.

See Also

contactnet.

Examples

```
set.seed(12345)
loc<- matrix(cbind(runif(50, 0, 10),runif(50, 0, 10)), ncol = 2, nrow = 50)
net1<- contactnet(type = "powerlaw", location = loc, beta = 1.5, nu = 0.5)
plot(net1)
net2<- contactnet(type = "Cauchy", location = loc, beta = 0.5)
plot(net2)
net3<- contactnet(type = "random", num.id = 50, beta = 0.08)
plot(net3)
# adding some options to the igraph plot:
plot(net3, vertex.color = "red", vertex.size = 8, edge.color = "black",
vertex.label.cex = 0.7, vertex.label.color = "black")
```

plot.datagen *S3 method to plot summary of epidemic.*

Description

plot.datagen is an S3 method that provides different plots summarizing epidemic of class datagen.

Usage

```
## S3 method for class 'datagen'
plot(x, plottype, time.index = NULL, ...)
```

Arguments

x	an object that has a class of datagen.
plottype	History or propagation plots of the epidemic.
time.index	used for obtaining propagation plots at specific infection time points rather than at every infection time. A plot at the first time point is included in any chosen set of infection time points. The default is to provide propagation plot at every infection time point. It is not required when plottype is set to “history”.
...	additional arguments that are passed to the generic plot function.

Details

If `plottype` is set to "history", the function produces epidemic curves of infection and removal times (SIR model), or infection, notification, and removal times (SINR model). If it is set to "propagation", the function produces plots of the propagation of epidemic over time based on the setting option of the argument `kerneltype`. With the network kernel, the function plots all the connections between individuals (in gray) and overlays these with the epidemic pathway direction over time. With the distance kernel, the function plots the epidemic dispersion over time. It shows the changes in the individual status that related to the chosen compartmental framework in `type`.

Value

plot

See Also

`contactnet`, `datagen`.

`plot.epictmcmc` *Plot the output of* `epictmcmc` *object*

Description

`plot.epictmcmc` is an S3 method that plots the output of an S3 object of class `epictmcmc`.

Usage

```
## S3 method for class 'epictmcmc'
plot(x, epi = NULL, plottype = NULL, start = NULL, end = NULL,
thin = NULL, ...)
```

Arguments

`x` an S3 object of class `epictmcmc` (i.e. the output of the `epictmcmc` function).

`epi` epidemic data of class `datagen`.

`plottype` determines which of three options to plot the output of the `epictmcmc` function are used: "parameter" produces trace plots for each of the model parameters, and "inf.times" ("rem.times") produces plots of the average posterior and 95% CI of the unobserved infection (removal) times when `datatype` set to "known removal" ("unknown removal").

`start, end, thin` options for creating `mcmc` object.

`...` additional arguments that are passed to the generic `plot` function.

Value

The argument plottype is used to specify the kind of plot produced. Trace and density plots of the posterior distribution of the model parameters can be obtained by setting plottype = “parameter”. Additionally, in the case of unobserved infection (removal) times, a plot of the average posterior and 95% percentile intervals of the inferred infection (removal) times are produced by setting plottype = “inf.times” (“rem.times”).

See Also

epictmcmc, print.epictmcmc, summary.epictmcmc, mcmc, plot.mcmc.

print.epictmcmc *Prints the contents of an* epictmcmc *object to the console*

Description

print.epictmcmc is an S3 method that prints the content of an S3 object of class epictmcmc to the console.

Usage

```
## S3 method for class 'epictmcmc'
print(x, digits = 6, ...)
```

Arguments

x	an S3 object of class epictmcmc (i.e. the output of the epictmcmc function).
digits	the number of printed digits of the estimates. Default value is 6.
...	additional arguments that are passed to the generic print function

Value

Formatted output of a epictmcmc object.

See Also

epictmcmc, summary.epictmcmc, plot.epictmcmc.

SpatialData *Simulated epidemic data set from a distance-based SIR ILM*

Description

This is a simulated epidemic data set of population size = 50 individuals that was generated using a distance-based SIR individual-level model (ILM) with power-law distance kernel. The model has no susceptible covariates and the infectivity rate is given by:

$$\lambda_{jt} = \alpha_0 \sum_{i \in I_t} d_{ij}^{-\beta}$$

The infectious period is assumed to follow an exponential distribution with rate λ. The epidemic was simulated with the following parameter values: $\alpha_0 = 2.0$, $\beta = 3$ and $\delta = 2$.

The data set file is an object of class "datagen" that has a list of `type`, `kerneltype`, `epidat` and `location`.

Usage

```
data(SpatialData)
```

Format

An object of class “datagen” that contains the following:

type: The “SIR” compartmental framework.

kerneltype: The “distance” kernel function.

epidat: A matrix of the simulated epidemic with four columns as: the id numbers of individuals, removal times, infectious periods, and infection times.

location: A matrix of the XY coordinates of individuals.

network: A zero matrix of contact network.

SpatialNetData *Simulated epidemic data set from a distance and network-based SIR ILM*

Description

This is a simulated epidemic data set of population size = 50 individuals that was generated using a combined distance and network-based SIR individual-level model (ILM) with power-law distance kernel. The model has one binary susceptible covariate and the infectivity rate is given by:

$$\lambda_{jt} = (\alpha_0 + \alpha_1 z_j) \sum_{i \in I_t} d_{ij}^{-\beta_1} + \beta_2 c_{ij}$$

The infectious period is assumed to follow an exponential distribution with rate δ. The epidemic was simulated with the following parameter values: $\alpha_0 = 0.008$, $\alpha_1 = 0.2$, $\beta_1 = 2$, $\beta_2 = 0.5$ and $\delta = 2$.

The data set file is a list of an object of class "datagen" that contains of `type`, `kerneltype`, `epidat`, `location` and `network`, and the covariate matrix.

Usage

```
data(SpatialNetData)
```

Format

It is a list of an object of class “datagen” that contains the following:

type: The “SIR” compartmental framework.

kerneltype: The “both” distance and network kernel functions.

epidat: A matrix of the simulated epidemic with four columns as: the id numbers of individuals, removal times, infectious periods, and infection times.

location: A matrix of the XY coordinates of individuals.

network: The undirected binary contact network matrix.

and a 50×2 matrix of the covariates represents the unity intercept and the binary covariate z.

`summary.epictmcmc` *Summary method for* `epictmcmc` *objects*

Description

Summarize a `epictmcmc` object and return an object of class `summary.epictmcmc`.

Usage

```
## S3 method for class 'epictmcmc'
summary(object, digits = NULL, start = NULL, end = NULL, thin = NULL, ...)
## S3 method for class 'summary.epictmcmc'
print(x, digits, start, end, thin, ...)
```

Arguments

`x`, `object`	an S3 object of class `epictmcmc` (i.e. the output of the `epictmcmc` function).
`digits`	the number of printed digits of the estimates. Default value is 6.
`start`, `end`, `thin`	options for creating `mcmc` object.
`...`	potential further arguments (require by generic).

See Also

`epictmcmc`, `print.epictmcmc`, `plot.epictmcmc`.

`tswv` *Tomato Spotted Wilt Virus data*

Description

Data extracted from plots in Hughes et al. (1997)

Usage

```
data(tswv)
```

Format

A list of two objects of class “datagen” as follows:

tswvsir: an object of class “datagen” for the “SIR” compartmental framework.

tswvsinr: an object of class “datagen” for the “SINR” compartmental framework.

References

Hughes, G., McRoberts, N., Madden, L.V., Nelson, S. C. (1997). Validating mathematical models of plant disease progress in space and time. *IMA Journal of Mathematics Applied in Medicine and Biology*, 14, 85-112.

Index