Package 'EpiILM'

June 29, 2020

Title Spatial and Network Based Individual Level Models for Epidemics

Version 1.5.1

Date 2020-06-28

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

Imports methods

Author Vineetha Warriyar. K. V., Waleed Almutiry, and Rob Deardon

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

Description Provides tools for simulating from discrete-time individual level models for infectious disease data analysis. This epidemic model class contains spatial and contact-network based models with two disease types: Susceptible-Infectious (SI) and Susceptible-Infectious-Removed (SIR).

License GPL (>= 2)

URL https://github.com/vineetha-warriyar/EpiILM

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-06-29 12:42:52 UTC

R topics documented:

`EpiILM-package` **EpiILM**: *Spatial and Network Based Individual Level Models for Epidemics*

Description

The R package **EpiILM** is provided for simulating from, and carrying out Bayesian MCMC-based statistical inference for spatial and/or network-based individual-level modelling framework. The package allows for the incorporation of individual-level susceptibility and transmissibility covariates in models, and provides various methods of summarizing epidemic data sets.

Details

The R package **EpiILM** can be used to carry out simulation of epidemics, estimate the basic reproduction number, plot various epidemic summary graphics, calculate the log-likelihood, carry out Bayesian inference using Metropolis-Hastings MCMC, and implement posterior predictive checks and model selection for a given data set and model. The key functions for this package are detailed in the value section. One of the important functions `epimcmc` depends heavily on the `MCMC` from the **adaptMCMC** package for performing the MCMC analysis. This function implements the robust adaptive Metropolis sampler of Vihola (2012) for tuning the covariance matrix of the (normal) jump distribution adaptively to achieve the desired acceptance rate. The package has other features for making predictions or forecasting for a specific model via the `pred.epi` function. The main functions, including for epidemic simulation (`epidata`) and likelihood calculation (`epilike`) are coded in Fortran in order to achieve the goal of agile implementation.

Value

Key functions for this package:

`epidata`	Simulates epidemics for the specified model type and parameters.
`epilike`	Calculates the log-likelihood for the specified model and data set.
`epimcmc`	Runs an MCMC algorithm for the estimation of specified model parameters.
`pred.epi`	Computes posterior predictions for a specified epidemic model.

Author(s)

Vineetha Warriyar. K. V., Waleed Almutiry, and Rob Deardon
Maintainer: Waleed Almutiry <wkmtierie@qu.edu.sa>

References

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, 239-261.

Vihola, M. (2012) Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statistics and Computing*, 22(5), 997-1008. doi:10.1007/s11222-011-9269-5.

Examples

```
## Not run:
demo(EpiILM.spatial)
demo(EpiILM.network)

## End(Not run)
```

`as.epidata` *Discrete Time level information of an Epidemic*

Description

This function allows the user to generate objects of class `"epidata"`. The output of this function provides information required to be used in the other functions in the package.

Usage

```
as.epidata (type, n, x = NULL, y = NULL, inftime, infperiod = NULL, contact = NULL)
```

Arguments

`type`	Type of compartment framework, with the choice of "SI" for Susceptible-Infectious diseases and "SIR" for Susceptible-Infectious-Removed.
`n`	Population size
`x`	X coordinates of individuals.
`y`	Y coordinates of individuals.
`inftime`	Times at which individuals are infected to initialize epidemic simulation.
`infperiod`	Length of infectious period for each individual.
`contact`	A contact network matrix or an array of contact network matrices.

Value

An object of class `epidata` is returned containing the following:

type Type of compartment framework, with the choice of "SI" for Susceptible-Infectious diseases and "SIR" for Susceptible-Infectious-Removed.

XYcoordinates The XY-coordinates of individuals if distance-based ILM is used, otherwise is NULL.

contact Contact network matrix if network-based ILM is used, otherwise is NULL.

inftime The infection times of individuals.

remtime The removal times of individuals when `type` = “SIR”.

See Also

`epidata`, `plot.epidata`.

Examples

```
# generate 100 X-Y coordinates for a distance-based ILM

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

# suppose we know the infection times for a spatial SI model based on above X-Y coordinates

inftime  <- rpois(100, 8)

# Now we can convert above information to an epidata object with

out <- as.epidata(type = "SI", n = 100, x = x, y = y, inftime = inftime )
out
```

---

epiBR0 *Basic reproduction number (R0)*

Description

Gives a Monte Carlo estimate of the basic reproduction number for a specified SIR model and data set

Usage

```
epiBR0 (x = NULL, y = NULL, contact = NULL, sus.par, trans.par = NULL, beta,

        spark = NULL, infperiod, Sformula = NULL, Tformula = NULL, tmax,

        niter)
```

Arguments

x	X coordinates of individuals
y	Y coordinates of individuals
contact	Contact network(s)
sus.par	Susceptibility parameter(>0)
trans.par	Transmissibility parameter(>0)
beta	Spatial parameter(s) (>0) or network parameter (s) (>0) if contact is used
spark	Sparks parameter (>=0), representing infections unexplained by other parts of the model or infections coming in from outside the observed population, default value is zero
infperiod	Length of infectious period for each individual

`Sformula`	An object of class formula. See formula Individual-level covariate information associated with susceptibility can be passed through this argument. An expression of the form `~ model` is interpreted as a specification that the susceptibility function, $\Omega_S(i)$ is modelled by a linear predictor specified symbolically by the model term. Such a model consists of a series of terms separated by + and - operators. If there is no susceptibility covariate information, `Sformula` is null.
`Tformula`	An object of class formula. See formula Individual-level covariate information associated with transmissibility can be passed through this argument. An expression of the form `~ -1+model` is interpreted as a specification that the transmissibility function, $\Omega_T(j)$ is modelled by a linear predictor specified symbolically by the model terms without the incorporation of the intercept term. Such a model consists of a series of terms separated by + and - operators. If there is no transmissibility covariate information, `Tformula` is null.
`tmax`	The last time point of simulation
`niter`	Number of epidemic simulations to calculate basic reproduction number

Value

A list is returned with the following components:

`BasicR0`	The basic reproduction number value
`simulated_BR0`	Number of infections per simulation

Examples

```
# generate 100 X-Y coordinates for a distance-based ILM

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

# Suppose we know the length of infectious period for each individual. Also, assume
# susceptibility parameter = 1.5 and spatial parameter = 5 for this SIR model

infperiod <- rep(3, 100)

# For a 1000 iteration with a last observed time point 15, we can estimate the basic
# reproduction number using Monte Carlo simulation

out <- epiBR0(x = x, y = y,  sus.par = 1.5,  beta = 5, infperiod= infperiod,

               tmax = 15, niter = 1000)

out$BasicR0
```

epidata *Simulates epidemic for the specified model type and parameters*

Description

This function allows the user to simulate epidemics under different models and scenarios

Usage

```
epidata (type, n, tmin = NULL, tmax, sus.par, trans.par = NULL, beta = NULL, spark = NULL,

          Sformula = NULL, Tformula = NULL, x = NULL, y = NULL,

          inftime = NULL, infperiod = NULL, contact = NULL)
```

Arguments

type	Type of compartment framework, with the choice of "SI" for Susceptible-Infectious diseases and "SIR" for Susceptible-Infectious-Removed.
n	Population size
tmin	The time point at which simulation begins, default value is one.
tmax	The last time point of simulation.
sus.par	Susceptibility parameter (>0).
trans.par	Transmissibility parameter (>0).
beta	Spatial parameter(s) (>0) or network parameter (s) (>0) if contact network is used.
spark	Sparks parameter (>=0), representing infections unexplained by other parts of the model (eg. infections coming in from outside the observed population), default value is zero.
Sformula	An object of class formula. See formula. Individual-level covariate information associated with susceptibility can be passed through this argument. An expression of the form `~ model` is interpreted as a specification that the susceptibility function, $\Omega_S(i)$ is modelled by a linear predictor specified symbolically by the model term. Such a model consists of a series of terms separated by + and - operators. If there is no susceptibility covariate information, `Sformula` is null.
Tformula	An object of class formula. See formula. Individual-level covariate information associated with transmissibility can be passed through this argument. An expression of the form `~ -1+model` is interpreted as a specification that the transmissibility function, $\Omega_T(j)$ is modelled by a linear predictor specified symbolically by the model terms without the incorporation of the intercept term. Such a model consists of a series of terms

	separated by + and - operators. If there is no transmissibility covariate information, `Tformula` is null.
`x`	X coordinates of individuals.
`y`	Y coordinates of individuals.
`inftime`	Times at which individuals are infected to initialize epidemic simulation.
`infperiod`	Length of infectious period for each individual.
`contact`	A contact network matrix or an array of contact network matrices.

Details

We consider following two individual level models:

Spatial model:

$$P(i,t) = 1 - \exp\{-\Omega_S(i) \sum_{j \in I(t)} \Omega_T(j) d_{ij}^{-\beta} - \varepsilon\}$$

Network model:

$$P(i,t) = 1 - \exp\{-\Omega_S(i) \sum_{j \in I(t)} \Omega_T(j)(\beta_1 C_{ij}^{(1)} + \ldots + \beta_n C_{ij}^{(n)}) - \varepsilon\}$$

where $P(i,t)$ is the probability that susceptible individual i is infected at time point t, becoming infectious at time t+1; $\Omega_S(i)$ is a susceptibility function which accommodates potential risk factors associated with susceptible individual i contracting the disease; $\Omega_T(j)$ is a transmissibility function which accommodates potential risk factors associated with infectious individual j; ε is a sparks term which represents infections originating from outside the population being observed or some other unobserved infection mechanism.

The susceptibility function can incorporate any individual-level covariates of interest and $\Omega_S(i)$ is treated as a linear function of the covariates, i.e., $\Omega_S(i) = \alpha_0 + \alpha_1 X_1(i) + \alpha_2 X_2(i) + \ldots + \alpha_{n_s} X_{n_s}(i)$, where $X_1(i), \ldots, X_{n_s}(i)$ denote $n_s$covariates associated with susceptible individual i, along with susceptibility parameters $\alpha_0, \ldots, \alpha_{n_s} > 0$. If the model does not contain any susceptibility covariates then $\Omega_S(i) = \alpha_0$ is used. In a similar way, the transmissibility function can incorporate any individual-level covariates of interest associated with infectious individual. $\Omega_T(j)$ is also treated as a linear function of the covariates, but without the intercept term, i.e., $\Omega_T(j) = \phi_1 X_1(j) + \phi_2 X_2(j) + \ldots + \phi_{n_t} X_{n_t}(j)$, where $X_1(j), \ldots, X_{n_t}(j)$ denote the n_t covariates associated with infectious individual j, along with transmissibility parameters $\phi_1, \ldots, \phi_{n_t} > 0$. If the model does not contain any transmissibility covariates then $\Omega_T(j) = 1$ is used.

Value

An object of class `epidata` is returned containing the following:

type Type of compartment framework, with the choice of "SI" for Susceptible-Infectious diseases and "SIR" for Susceptible-Infectious-Removed

XYcoordinates The XY-coordinates of individuals.

contact Contact network matrix.

inftime The infection times of individuals.

remtime The removal times of individuals when `type` = "SIR".

References

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, 239-261.

Deardon, R., Fang, X., and Kwong, G.P.S. (2014). Statistical modelling of spatio-temporal infectious disease transmission in analyzing and modeling Spatial and temporal dynamics of infectious diseases, *(Ed: D. Chen, B. Moulin, J. Wu), John Wiley & Sons.* Chapter 11.

See Also

plot.epidata, epimcmc, epilike, pred.epi.

Examples

```
## Example 1:  spatial SI model
# generate 100 individuals

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)

out1 <- epidata(type = "SI",n = 100, Sformula = ~covariate, tmax = 15,
               sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)

# Plots of epidemic progression (optional)

plot(out1, plottype = "spatial")
plot(out1, plottype = "curve", curvetype = "newinfect")

## Example 2: spatial SIR model
# generate infectious period(=3) for 100 individuals

lambda <- rep(3, 100)

out2 <- epidata(type = "SIR", n = 100, tmax = 15, sus.par = 0.3, beta = 5.0, infperiod = lambda,
        x = x, y = y)

plot(out2, plottype = "spatial")
plot(out2, plottype = "curve", curvetype = "newinfect")

## Example 3:   SI network model

contact1 <- matrix(rbinom(10000, 1, 0.1), nrow = 100, ncol = 100)

contact2 <- matrix(rbinom(10000, 1, 0.1), nrow = 100, ncol = 100)

diag(contact1[,] ) <- 0
```

```
diag(contact2[,] ) <- 0

contact <- array(c(contact1, contact2), dim = c(100, 100, 2))

out3 <- epidata(type = "SI", n = 100, tmax = 15, sus.par = 0.3, beta = c(3.0, 5.0),
        contact = contact)
plot(out3, plottype = "curve", curvetype = "complete")
plot(out3, plottype = "curve", curvetype = "susceptible")
plot(out3, plottype = "curve", curvetype = "newinfect")
plot(out3, plottype = "curve", curvetype = "totalinfect")
```

---

epidic *Deviance Information Criterion (DIC)*

Description

Computes the Deviance Information Criterion for individual level models

Usage

```
epidic (burnin, niter, LLchain, LLpostmean)
```

Arguments

burnin	Burnin period for MCMC
niter	Number of MCMC iterations
LLchain	Loglikelihood values from the MCMC output
LLpostmean	Loglikelihood value of the model with posterior mean of estimates

References

Spiegelhalter, D., Best, N., Carlin, B., Van der Linde, A. (2002). Bayesian Measures of Model Complexity and Fit. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 64(4), 583-639.

Examples

```
## Example 1:  spatial SI model
# generate 100 individuals

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)
```

```
out1 <- epidata(type = "SI", n = 100, Sformula = ~covariate, tmax = 15,
              sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)

unif_range <- matrix(c(0, 0, 10000, 10000), nrow = 2, ncol = 2)

# estimate parameters
mcmcout <- epimcmc(out1, tmax = 15, niter = 1500,
                    Sformula = ~covariate,
                    sus.par.ini = c(0.003, 0.01), beta.ini =0.01,
                    pro.sus.var = c(0.1, 0.1),pro.beta.var = 0.5,
                    prior.sus.par = unif_range,
                   prior.sus.dist = c("uniform","uniform"), prior.beta.dist = "uniform",
                    prior.beta.par = c(0, 10000), adapt = TRUE, acc.rate = 0.5 )
# store the estimates
sus.parameters = c(mean(unlist(mcmcout$Estimates[1])), mean(unlist(mcmcout$Estimates[2])))
beta.par = mean(unlist(mcmcout$Estimates[3]))

# likelihood value
loglike <- epilike(out1, tmax = 15, Sformula = ~covariate, sus.par = sus.parameters,
                   beta = beta.par)

# deviance information criterion calculation for the above epidemic
dic <- epidic(burnin = 500, niter = 1500, LLchain = mcmcout$Loglikelihood,
              LLpostmean = loglike)
dic
```

`epilike` *Calculates the log likelihood*

Description

Calculates the log likelihood for the specified individual level model and data set

Usage

```
epilike (object, tmin = NULL, tmax, sus.par, trans.par = NULL,

         beta = NULL, spark = NULL, Sformula = NULL, Tformula = NULL)
```

Arguments

`object`	An object of class `epidata` that can be the output of `epidata` or `as.epidata`.
`tmin`	The first time point at which data is observed, default value is one.
`tmax`	The last time point at which data is observed.
`sus.par`	Susceptibility parameter(>0).
`trans.par`	Transmissibility parameter(>0).

`beta` Spatial parameter(s) (>0) or network parameter (s) (>0) if contact network is used.

`spark` Sparks parameter(>=0), representing infections unexplained by other parts of the model or infections coming in from outside the observed population, default value is zero.

`Sformula` An object of class formula. See formula.

Individual-level covariate information associated with susceptibility can be passed through this argument. An expression of the form `~ model` is interpreted as a specification that the susceptibility function, $\Omega_S(i)$ is modelled by a linear predictor specified symbolically by the model term. Such a model consists of a series of terms separated by + and - operators. If there is no susceptibility covariate information, `Sformula` is null.

`Tformula` An object of class formula. See formula.

Individual-level covariate information associated with transmissibility can be passed through this argument. An expression of the form `~ -1+model` is interpreted as a specification that the transmissibility function, $\Omega_T(j)$ is modelled by a linear predictor specified symbolically by the model terms without the incorporation of the intercept term. Such a model consists of a series of terms separated by + and - operators. If there is no transmissibility covariate information, `Tformula` is null.

Value

Returns the value of the log-likelihood function.

References

Deardon R, Brooks, S. P., Grenfell, B. T., Keeling, M. J., Tildesley, M. J., Savill, N. J., Shaw, D. J., Woolhouse, M. E. (2010). Inference for individual level models of infectious diseases in large populations. *Statistica Sinica*, 20, 239-261.

See Also

epimcmc.

Examples

```
## Example 1:  spatial SI model
# generate 100 individuals

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)

out1 <- epidata(type = "SI", n = 100, Sformula = ~covariate, tmax = 15,
              sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)
```

```
epilike(out1, tmax = 15,
         sus.par = c(0.1, 0.3), beta = 5, Sformula = ~covariate)

## Example 2: spatial SIR model
# generate infectious period (=3) for 100 individuals

lambda <- rep(3, 100)

out2 <- epidata(type = "SIR", n = 100, tmax = 15, sus.par  =0.3, beta = 5.0,
              infperiod = lambda, x = x, y = y)

epilike(out2,
        tmax = 15, sus.par = 0.3, beta = 5.0)
```

---

epimcmc *Monte Carlo Simulation*

Description

Runs an MCMC algorithm for the estimation of specified model parameters

Usage

```
epimcmc (object, tmin = NULL, tmax,

      niter, sus.par.ini, trans.par.ini = NULL, beta.ini = NULL, spark.ini = NULL,

         Sformula = NULL, Tformula = NULL,

      pro.sus.var, pro.trans.var = NULL, pro.beta.var = NULL, pro.spark.var = NULL,

         prior.sus.dist, prior.trans.dist = NULL, prior.beta.dist = NULL,

      prior.spark.dist = NULL, prior.sus.par, prior.trans.par, prior.beta.par = NULL,

         prior.spark.par = NULL, adapt = FALSE, acc.rate = NULL)
```

Arguments

object	An object of class epidata that can be the output of epidata or as.epidata.
tmin	The first time point at which the infection occurs, default value is one.
tmax	The last time point at which data is observed.
niter	Number of MCMC iterations.
sus.par.ini	Initial value(s) of the susceptibility parameter(s) (>0).
trans.par.ini	Initial value(s) of the transmissibility parameter(s) (>0).

`beta.ini` Initial value(s) of the spatial parameter(s) (>0) or the network parameter(s) (>0) if contact network is used.

`spark.ini` Initial value of the spark parameter (>=0).

`Sformula` An object of class formula. See formula

Individual-level covariate information associated with susceptibility can be passed through this argument. An expression of the form `~ model` is interpreted as a specification that the susceptibility function, $\Omega_S(i)$ is modelled by a linear predictor specified symbolically by the model term. Such a model consists of a series of terms separated by + and - operators. If there is no susceptibility covariate information, `Sformula` is null.

`Tformula` An object of class formula. See formula

Individual-level covariate information associated with transmissibility can be passed through this argument. An expression of the form `~ -1+model` is interpreted as a specification that the transmissibility function, $\Omega_T(j)$ is modelled by a linear predictor specified symbolically by the model terms without the incorporation of the intercept term. Such a model consists of a series of terms separated by + and - operators. If there is no transmissibility covariate information, `Tformula` is null.

`pro.sus.var` Proposal density variance(s) for susceptibility parameter(s). If a zero value is assigned to the proposal variance of any parameter, the parameter is considered fixed to its `sus.par.ini` value.

`pro.trans.var` Proposal density variance(s) for transmissibility parameter(s). If a zero value is assigned to the proposal variance of any parameter, the parameter is considered fixed to its `sus.par.ini` value.

`pro.beta.var` Proposal density variance(s) for beta parameter(s). If a zero value is assigned to the proposal variance of any parameter, the parameter is considered fixed to its `sus.par.ini` value.

`pro.spark.var` Proposal density variance for the spark parameter.

`prior.sus.dist` Select the prior distribution(s) for the susceptibility parameter(s) with the choice of "halfnormal" for positive half normal distribution, "gamma" for gamma distribution and "uniform" for uniform distribution

`prior.trans.dist`

Select the prior distribution(s) for the transmissibility parameter(s) with the choice of "halfnormal" for positive half normal distribution, "gamma" for gamma distribution and "uniform" for uniform distribution

`prior.beta.dist`

Select the prior distribution(s) for the beta parameter(s) with the choice of "halfnormal" for half normal distribution, "gamma" for gamma distribution and "uniform" for uniform distribution

`prior.spark.dist`

Select the prior distribution for the spark parameter with the choice of "halfnormal" for half normal distribution, "gamma" for gamma distribution and "uniform" for uniform distribution

`prior.sus.par` A vector (matrix) of the prior distribution parameters for updating the susceptibility parameter(s).

`prior.trans.par`
: A vector (matrix) of the prior distribution parameters for updating the transmissibility parameter(s).

`prior.beta.par`
: A vector (matrix) of the prior distribution parameters for updating the kernel parameter(s).

`prior.spark.par`
: A vector of the prior distribution parameters for updating the spark parameter.

`adapt`
: To enable the adaptive MCMC method in the `MCMC` function, default is FALSE.

`acc.rate`
: To set an acceptance rate. This option will be ignored if `adapt` = FALSE. See `MCMC` for more details.

Details

Independent Gaussian random walks are used as the Metropolis-Hastings MCMC proposal for all parameters. The `epimcmc` function depends on the `MCMC` function from the **adaptMCMC** package.

Value

Returns an object of class `epimcmc` that contains:

type: the compartmental framework model used in the analysis.

kernel.type: the used `kernel.type` in the function (distance-based or network-based).

Estimates: the MCMC output of the updated model parameters.

Loglikelihood: the loglikelihood of the updated model parameters.

Fullsamples: the MCMC output of all the model parameters (including fixed parameters).

n.sus.par: the number of parameters in the susceptibility function.

n.trans.par: the number of parameters in the transmissibility function.

n.ker.par: the number of parameters in the kernel function.

References

Rob Deardon, Xuan Fang, and Grace P. S. Kwong (2015). Statistical modelling of spatio-temporal infectious disease tranmission in Analyzing and Modeling Spatial and Temporal Dynamics of Infectious Diseases, *(Ed: D. Chen, B. Moulin, J. Wu), John Wiley & Sons.*. Chapter 11.

See Also

`summary.epimcmc`, `plot.epimcmc`, `epidata`, `epilike`, `pred.epi`.

Examples

```
## Example 1:  spatial SI model
# generate 100 individuals

x <- runif(100, 0, 10)
```

```
y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)

out1 <- epidata(type = "SI", n = 100, Sformula = ~covariate, tmax = 15,
               sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)

alphapar1 <- matrix(c(1, 1, 1, 1), ncol = 2, nrow = 2)

betapar1 <- c(10, 2)

epi <- epimcmc(object = out1, tmin = 1, tmax = 15,
               niter = 1000, sus.par.ini = c(1, 1), beta.ini = 1,
               Sformula = ~covariate, pro.sus.var = c(0.5, 0.3), pro.beta.var = 0.1,
               prior.sus.dist = c("gamma", "gamma"), prior.beta.dist = "gamma",
               prior.sus.par = alphapar1, prior.beta.par = betapar1,
               adapt = TRUE, acc.rate = 0.5)
epi

## Example 2:  spatial SIR model

lambda <- rep(3, 100)

out2 <- epidata(type = "SIR", n = 100, tmax = 15, sus.par = 0.3, beta = 5.0, infperiod = lambda,
        x = x, y = y)

alphapar2 <- c(1, 1)
betapar2 <- c(1, 1)

epi2 <- epimcmc(object = out2, tmin = 1, tmax = 15,
               niter = 1000, sus.par.ini = 1, beta.ini = 1,
               Sformula = NULL, pro.sus.var = 0.3, pro.beta.var = 0.1,
               prior.sus.dist = "gamma", prior.beta.dist = "gamma",
               prior.sus.par = alphapar2, prior.beta.par = betapar2,
               adapt = FALSE, acc.rate = NULL)
epi2
```

`plot.epidata` *S3 method to provide some graphics of epidemic.*

Description

Produces various graphs summarizing epidemic of class `epidata`.

Usage

```
## S3 method for class 'epidata'
plot(x, plottype, curvetype = NULL, time_id = NULL, tmin = NULL, timepoints = NULL, ...)
```

Arguments

x	An object of class epidata that can be the output of epidata or as.epidata
plottype	Provide two types of plots. When plottype = "curve", graphs of various epidemic curves can be produced, and when plottype = "spatial", spatial plots of epidemic progression over time is produced.
curvetype	It has four options: "complete", "susceptible", "newinfect", and "totalinfect". See details for more information.
time_id	Specify time points at which the spatial square is plotted.
tmin	Initial time point at which infection occurs, default value is one.
timepoints	Specify time points at which the curve is plotted
...	.....

Details

The argument plottype has two options. When plottype="spatial" spatial plots are produced for the epidemic progression over time, and when plottype="curve", the argument curvetype has to be specified to one of the four available options: "complete" for plotting the number of susceptible, infected and removed individuals at each time point, "susceptible" for plotting the number of susceptible individuals at each time point, "newinfect" for plotting the number of newly infected individuals at each time point, and "totalinfect" for plotting the cumulative number of infected individuals at each time point.

Value

plot

See Also

epidata, plot.epimcmc, plot.pred.epi.

Examples

```
## Example :  spatial SI model
# generate 100 individuals

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)

out1 <- epidata(type = "SI",n = 100, Sformula = ~covariate, tmax = 15,
               sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)

# Plots of epidemic progression
```

```
plot(out1, plottype = "spatial")
plot(out1, plottype = "curve", curvetype = "newinfect")
```

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

Description

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

Usage

```
## S3 method for class 'epimcmc'
plot(x, partype, start = 1, end = NULL, thin = 1, ...)
```

Arguments

`x`	An S3 object of class `epimcmc` (i.e. the output of the `epimcmc` function).
`partype`	Determines which of two options to plot the output of the `epimcmc` function are used: "parameter" produces trace plots for each of the model parameters, and "loglik" produces trace plot of the log-likelihood of the MCMC samples.
`start, end, thin`	options for creating `mcmc` object.
`...`	additional arguments that are passed to the generic `plot` function.

Value

plot.

See Also

`epimcmc`, `summary.epimcmc`, `mcmc`, `plot.mcmc`.

Examples

```
## Example :  spatial SI model
# generate 100 individuals

set.seed(59991)

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)
```

```
out1 <- epidata(type = "SI", n = 100, Sformula = ~covariate, tmax = 15,
                sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)

alphapar1 <- matrix(c(1, 1, 1, 1), ncol = 2, nrow = 2)

betapar1 <- c(10, 2)

epi <- epimcmc(object = out1, tmin = 1, tmax = 15,
                niter = 1000, sus.par.ini = c(0.1, 0.1), beta.ini = 5,
                Sformula = ~covariate, pro.sus.var = c(0.2, 0.3), pro.beta.var = 0.8,
                prior.sus.dist = c("gamma", "gamma"), prior.beta.dist = "gamma",
                prior.sus.par = alphapar1, prior.beta.par = betapar1,
                adapt = TRUE, acc.rate = 0.5)

# plot estimates
plot(epi, partype = "parameter", start = 100)
```

`plot.pred.epi` *S3 method to provide plots of posterior predictive check.*

Description

Produces various graphs for the output of the posterior predictive check of class `pred.epi`.

Usage

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

Arguments

`x`	An object of class `pred.epi` which is the output of the `pred.epi` function.
`...`	additional arguments to be passed to `plot` generic function.

Value

plot

See Also

`pred.epi`, `plot.epidata`, `plot.epimcmc`.

pred.epi *Posterior predictive check.*

Description

Computing the posterior predictive check based on different summary statistics.

Usage

```
pred.epi (object, xx, criterion , n.samples, burnin = NULL, tmin = NULL,

  Sformula = NULL, Tformula = NULL, showProgressBar = interactive())
```

Arguments

object	An object of class epidata that can be the output of epidata or as.epidata
xx	An object of class epimcmc that is the output of epimcmc.
criterion	The (multivariate) statistical criteria used in the posterior predictive check. It has three options: "newly infectious" which is a multivariate statistics represented by the number of newly infectious individuals over time, "epidemic length" represents the length of epidemic, and "peak time" represents the time of the peak of epidemic.
n.samples	The number of epidemics that needs to be simulated in the posterior predictive check procedure.
burnin	A scalar value which represents the number of samples needs to be discarded from the MCMC output.
tmin	The first time point at which the infection occurs, default value is one.
Sformula	An object of class formula. See formula. Individual-level covariate information associated with susceptibility can be passed through this argument. An expression of the form ~ model is interpreted as a specification that the susceptibility function, $\Omega_S(i)$ is modelled by a linear predictor specified symbolically by the model term. Such a model consists of a series of terms separated by + and - operators. If there is no susceptibility covariate information, Sformula is null.
Tformula	An object of class formula. See formula. Individual-level covariate information associated with transmissibility can be passed through this argument. An expression of the form ~ -1+model is interpreted as a specification that the transmissibility function, $\Omega_T(j)$ is modelled by a linear predictor specified symbolically by the model terms without the incorporation of the intercept term. Such a model consists of a series of terms separated by + and - operators. If there is no transmissibility covariate information, Tformula is null.
showProgressBar	logical. If TRUE a progress bar is shown.

Value

An object of class `pred.epi` that contains the following:

type: The compartmental framework model used in the analysis.

criterion: The (multivariate) statistical criteria used in the posterior predictive check.

crit.sim: The output of the evaluated `criterion` on the simulated epidemics.

crit.obs: The output of the evaluated `criterion` on the observed epidemics.

tmax: The last time point at which data is observed.

n.samples: The number of simulated epidemics used in the posterior predictive check procedure.

References

Deardon R, Brooks, S. P., Grenfell, B. T., Keeling, M. J., Tildesley, M. J., Savill, N. J., Shaw, D. J., Woolhouse, M. E. (2010). Inference for individual level models of infectious diseases in large populations. *Statistica Sinica*, 20, 239-261.

See Also

`epimcmc`, `epidata`, `epilike`, `plot.pred.epi`.

Examples

```
## Example 1:  spatial SI model
# generate 100 individuals

set.seed(59991)

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- cbind(runif(100, 0, 2), rbinom(100, 1, 0.5))

out <- epidata(type = "SI",n = 100, Sformula = ~covariate, tmax = 15,
               sus.par = c(0.1, 0.3, 0.01), beta = 5.0, x = x, y = y)

alphapar2 <- matrix(c(1, 1, 1, 1, 1, 1), ncol = 2, nrow = 3)

betapar2 <- c(1, 1)

epi<-epimcmc(object = out, tmin = 1, tmax = 15,
       niter = 500, sus.par.ini = c(1, 1, 1), beta.ini = 1,
       Sformula = ~covariate,
 pro.sus.var = c(0.5, 0.3, 0.2), pro.beta.var = 0.1,
 prior.sus.dist = c("gamma", "gamma", "gamma"),
 prior.beta.dist = "gamma",
 prior.sus.par = alphapar2, prior.beta.par = betapar2,
       adapt = TRUE, acc.rate = 0.5)

epipred1 <- pred.epi (object = out, xx = epi,
```

```
criterion = "newly infectious",
  n.samples = 100, burnin = 200, tmin = 1,
  Sformula = ~covariate)

plot(epipred1, col = "red", type = "b", lwd = 2)

epipred2 <- pred.epi (object = out, xx = epi,
criterion = "peak time",
  n.samples = 100, burnin = 200, tmin = 1,
  Sformula = ~covariate)

plot(epipred2, col = "dark gray")
```

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

Description

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

Usage

```
## S3 method for class 'epimcmc'
summary(object, ...)
```

Arguments

`object`	an S3 object of class `epimcmc` (i.e. the output of the `epimcmc` function).
`...`	potential further arguments (require by generic).

See Also

`epimcmc`, `plot.epimcmc`.

Examples

```
## Example:  spatial SI model
# generate 100 individuals

x <- runif(100, 0, 10)

y <- runif(100, 0, 10)

covariate <- runif(100, 0, 2)

out1 <- epidata(type = "SI", n = 100, Sformula = ~covariate, tmax = 15,
```

```
                sus.par = c(0.1, 0.3), beta = 5.0, x = x, y = y)

alphapar1 <- matrix(c(1, 1, 1, 1), ncol = 2, nrow = 2)

betapar1 <- c(1, 1)

epi <- epimcmc(object = out1, tmin = 1, tmax = 15,
                niter = 1000, sus.par.ini = c(1, 1), beta.ini = 1,
                Sformula = ~covariate, pro.sus.var = c(0.5, 0.3), pro.beta.var = 0.1,
                prior.sus.dist = c("gamma", "gamma"), prior.beta.dist = "gamma",
                prior.sus.par = alphapar1, prior.beta.par = betapar1,
                adapt = TRUE, acc.rate = 0.5)

# summary of mcmc output
summary(epi)
```

`tswv` *Tomato Spotted Wilt Virus (TSWV) data*

Description

Data extracted from Hughes et al. (1997). Data obtained from a field experiment as the spatial dynamics of tomato spotted wilt virus (tswv).

Usage

```
data(tswv)
```

Format

A data frame with following variables

x X coordinate

y Y coordinate

inftime Infection times

removaltime Times at which individuals are removed

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.

Examples

```
data("tswv")

x <- tswv$x
y <- tswv$y
inftime <- tswv$inftime
removaltime <- tswv$removaltime
infperiod <- rep(3, length(x))

# change to epilate object
 epidat.tswv <- as.epidata(type = "SIR", n = 520, x = x, y = y,
                           inftime = inftime, infperiod = infperiod)

# plot
plot(epidat.tswv, plottype = "spatial", tmin = 2)
```

Index