

Package: SEIRfancy (via r-universe)

October 12, 2024

Type Package

Title Extended Susceptible-Exposed-Infected-Recovery Model

Version 1.1.1

Description Extended Susceptible-Exposed-Infected-Recovery Model for handling high false negative rate and symptom based administration of diagnostic tests.
<doi:10.1101/2020.09.24.20200238>.

Maintainer Michael Kleinsasser <biostat-cran-manager@umich.edu>

License GPL-2

Encoding UTF-8

LazyData true

Biarch true

Depends R (>= 3.5.0)

Imports dplyr, arm, pbapply, DescTools, patchwork, ggpubr, ggplot2, knitr, rlang, magrittr, scales

URL <https://github.com/umich-biostatistics/SEIRfancy>

BugReports <https://github.com/umich-biostatistics/SEIRfancy/issues>

RoxygenNote 7.1.1

Roxygen list(markdown = TRUE)

Repository <https://umich-biostatistics.r-universe.dev>

RemoteUrl <https://github.com/umich-biostatistics/seirfancy>

RemoteRef HEAD

RemoteSha 97f08b28553f116567fc26eece15d7bf72fb034f

Contents

covid19	2
plot.SEIRfancy	2
plot.SEIRfancyPredict	3
SEIRfancy	3
SEIRfancy.predict	7

Index**11**

covid19

*COVID-19 Cases Time Series in India***Description**

Contains dailies and totals of cases, recoveries, and deaths from the COVID-19 outbreak in India from January 30 to September 21 of 2020.

Usage

```
covid19
```

Format

An object of class `data.frame` with 236 rows and 7 columns.

Date Date as a character string

Daily.Confirmed Daily confirmed cases as an integer vector

Total.Confirmed Total confirmed cases upto current date as an integer vector

Daily.Recovered Daily recovered cases as an integer vector

Total.Recovered Total confirmed cases upto current date as an integer vector

Daily.Deceased Daily deceased cases as an integer vector

Total.Deceased Total deceased cases upto current date as an integer vector

Examples

```
covid19
head(covid19)
tail(covid19)
```

plot.SEIRfancy

*Plot an SEIRfancy Fit Object***Description**

This is a convenient wrapper for output that is already included in the SEIRfancy output in the plots element. Options are trace or boxplot.

Usage

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

Arguments

x	SEIRfancy object to plot
type	type of plot to render. Options are "trace" or "boxplot".
...	not currently used

plot.SEIRfancyPredict *Plot an SEIRfancyPredict Fit Object*

Description

This is a convenient wrapper for output that is already included in the predict output in the plots element. Options are panel and cases.

Usage

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

Arguments

x	SEIRfancyPredict object to plot
type	type of plot to render. Options are "trace", "boxplot", "panel", or "cases".
...	not currently used

SEIRfancy *Estimate SEIRfancy Model Parameters*

Description

This function is used to estimate the different parameters of interest like the time varying transmission rates, proportion of reported cases, and the basic reproduction rates.

Usage

```
SEIRfancy(
  data,
  data_init,
  N,
  init_pars = NULL,
  niter = 1e+05,
  BurnIn = 1e+05,
  model = "Multinomial",
  plot = TRUE,
  period_start,
  auto.initialize = TRUE,
  ...
)
```

Arguments

<code>data</code>	(mandatory): If the model is Multinomial, then the data matrix should contain the 3 columns Confirmed, Recovered, and Death. If the model is Poisson or Binomial, then the data should contain only the column Confirmed. Please ensure that the names of the columns are exactly as stated above.
<code>data_init</code>	(mandatory): These are the initial data values provided by the user as a numeric vector of length six. The entries should be the Total Confirmed, Total Recovered, Total Death, Daily Confirmed, Daily Recovered, and Daily Death for the Starting Date. Note: If the starting total confirmed is 0, please replace it by 1.
<code>N</code>	(mandatory): The population size.
<code>init_pars</code>	NULL(default): If not equal to NULL, then the user can give a user input initial parameters which should consist of the initial values of the time varying beta, proportion of testing for the different periods.
<code>niter</code>	1e5 (default): Number of iterations for the MCMC Metropolis Hastings algorithm.
<code>BurnIn</code>	5e4 (default): This is the number of burn-in iterations for the MCMC algorithm
<code>model</code>	"Multinomial" (default): This is the likelihood function that will be used. There are three options available: "Multinomial", "Poisson", or "Binomial".
<code>plot</code>	TRUE (default): This will give the box plot for the basic reproduction number for the different periods.
<code>period_start</code>	The total time period is divided into small periods depending on the lock down measures imposed by the government. So this is a numeric vector consisting of the start dates for the different time periods.
<code>auto.initialize</code>	TRUE (default): This is the option for using a mle based initial parameter.
<code>...</code>	arguments passed to the function <code>model_initializeR</code> and <code>model_plotR</code> which is used for initializing the parameters. The parameters are described below: <ul style="list-style-type: none"> • <code>step_pars</code>: <code>init_pars/500</code> (default): It is the variance of the proposal distribution for the Metropolis Hastings Algorithm which is assumed to be a Random Walk. • <code>alpha_p</code>: 0.5 (default): It is defined as the ratio of rate of spread of infection by tested positive patients to that by false negatives. We have taken $\alpha_p < 1$ as patients who are tested positive are subjected to quarantine where the chance of spreading the disease is less than that of false negative patients who are mostly unaware of their infectious nature. So, false negative individuals continue to spread the disease freely at a higher rate than tested positive individuals. • <code>alpha_u</code>: 0.7 (default): It is defined as the scaling factor for the rate of spread of infection by untested individuals. α_u is assumed to be < 1 as U mostly consists of asymptomatic or mildly symptomatic cases who are known to spread the disease at a much lower rate than those with higher levels of symptoms. • <code>beta_1</code>: 0.6 (default): It is the scaling factor for rate of recovery for untested individuals. β_1 is assumed to be less than 1. The condition of

Untested individuals is not as severe as they consist of mostly asymptomatic people. So, they are assumed to recover faster than the Current Positive ones.

- `beta_2`: 0.7 (default): It is the inverse of the scaling factor for rate of recovery for false negative individuals. `$beta_2$` is assumed to be less than 1. It is assumed that the recovery rate is slower than the detected ones for the False Negative ones because they are not getting any hospital treatments.
- `delta_1`: 0.3 (default): It is the scaling factor for death rate for undetected individuals. `$delta_1$` is assumed to be less than 1. Similarly for the Untested ones, the death rate is taken to be lesser because they are mostly asymptomatic. So, their probability of dying is much lower.
- `delta_2`: 0.7 (default): It is the inverse of the scaling factor for death rate for false negative individuals. `$delta_2$` is assumed to be less than 1. Same as before, the death rate for False Negative ones are assumed to be higher than the Current detected Positive as they are not receiving proper treatment.
- `lambda`: $1 / (66.26 * 365)$ (default): Natural birth rate. The value given here as the default value is the world's common birth rate.
- `mu`: $1 / (66.26 * 365)$ (default): Natural death rates. This is assumed to be equal with natural birth rate for the sake of simplicity.
- `D_d`: 17.8 (default): Mean days until death for positive individuals.
- `D_e`: 5.2 (default): Incubation period.
- `Dr`: 17.8 (default): Mean days until recovery for positive individuals.
- `f`: 0.15 (default): False negative probability of RT-PCR test.
- `mCFR`: NULL (It is calculated from the data by default) It is defined as the ratio of the total reported deceased cases and the total removed cases until that day.
- `init.exposed.ratio`: 3 (default): This is the scaling factor for the calculation of the initial number of exposed people from the sum of the initial number of unreported, reported people.
- `init.confirmed.ratio`: 0.15 (default): This is the initial value of the probability of being tested.
- `opt_num`: 100 (default): The number of times an user wants to run the mle optimization before deciding on the best initial parameter.
- `trace_plot.common_axis`: FALSE (default): This will give the trace plot for the convergence of the MCMC estimated time varying parameters.
- `save plot`: TRUE (default): It is the option for saving the plots in the directory folder.

Value

A list with class "SEIRfancy", which contains the items described below:

- `mcmc_pars`: a matrix of the mcmc draws for the parameters
- `plots`: a list of ggplot objects

Examples

```

library(dplyr)
train = covid19[which(covid19$Date == "01 April "):which(covid19$Date == "30 June "),]
test = covid19[which(covid19$Date == "01 July "):which(covid19$Date == "31 July "),]

train_multinom =
  train %>%
    rename(Confirmed = Daily.Confirmed,
           Recovered = Daily.Recovered,
           Deceased = Daily.Deceased) %>%
    dplyr::select(Confirmed, Recovered, Deceased)

test_multinom =
  test %>%
    rename(Confirmed = Daily.Confirmed,
           Recovered = Daily.Recovered,
           Deceased = Daily.Deceased) %>%
    dplyr::select(Confirmed, Recovered, Deceased)

N = 1341e6 # population size of India
data_initial = c(2059, 169, 58, 424, 9, 11)
pars_start = c(c(1,0.8,0.6,0.4,0.2), c(0.2,0.2,0.2,0.25,0.2))
phases = c(1,15,34,48,62)

cov19est = SEIRfansy(data = train_multinom, init_pars = pars_start,
                    data_init = data_initial, niter = 1e3, BurnIn = 1e2,
                    model = "Multinomial", N = N, lambda = 1/(69.416 * 365),
                    mu = 1/(69.416 * 365), period_start = phases, opt_num = 1,
                    auto.initialize = TRUE, f = 0.15)

names(cov19est)
class(cov19est$mcmc_pars)
names(cov19est$plots)

plot(cov19est, type = "trace")
plot(cov19est, type = "boxplot")

# quick test for package check
# not for use outside CRAN check()
cov19est = SEIRfansy(data = train_multinom, init_pars = pars_start,
                    data_init = data_initial, niter = 33, BurnIn = 18,
                    model = "Multinomial", N = N, lambda = 1/(69.416 * 365),
                    mu = 1/(69.416 * 365), period_start = phases, opt_num = 1,
                    auto.initialize = TRUE, f = 0.15, plot = FALSE, system_test = NULL)

```

SEIRfancy.predict *Prediction for SEIRfancy Model*

Description

This function is used to predict the total reported as well as unreported case counts, total recovered, and total deaths.

Usage

```
SEIRfancy.predict(
  data = NULL,
  data_init,
  init_pars = NULL,
  N,
  plot = TRUE,
  T_predict,
  period_start,
  estimate = TRUE,
  pars = NULL,
  data_test = NULL,
  auto.initialize = TRUE,
  ...
)
```

Arguments

<code>data</code>	(mandatory): input the training data set. If the model is Multinomial then the data matrix should contain the 3 columns Confirmed, Recovered, and Death. If the model is Poisson or Binomial, then the data should contain only the column Confirmed. Note that the names of the columns must be the same as stated above.
<code>data_init</code>	(mandatory): These are the initial data values provided by the user as a numeric vector of length six. The entries should be the Total Confirmed, Total Recovered, Total Death, Daily Confirmed, Daily Recovered, and Daily Death for the Starting Date. Note: If the starting total confirmed is 0, please replace it by 1.
<code>init_pars</code>	NULL (default): If not equal to NULL, then the user can give a user input initial parameters which should consist of the initial values of the time varying beta, proportion of testing for the different periods.
<code>N</code>	(mandatory): The population size.
<code>plot</code>	TRUE (default): If <code>estimate = FALSE</code> , this will give two plots. One is the panel plot for total cases, total recovered, total death, and total confirmed if the model is Multinomial. Otherwise it will give only a plot for total confirmed when

the model is binomial or Poisson, and the second plot is the plot of untested, false negative, and reported cases. And when `estimate = TRUE`, it will give two other plots along with the previous two plots. One is the box plot for basic reproduction number and the other one is the trace plot for the convergence of the MCMC parameters.

<code>T_predict</code>	It is the number of days that we want to predict after the train period. The value of <code>T_predict</code> should be greater than or equal to the number of rows of <code>data_test</code> .
<code>period_start</code>	The total time period is divided into small periods depending on the lock down measures imposed by the government. So this is a numeric vector consisting of the start dates for the different time periods.
<code>estimate</code>	<code>TRUE</code> (default): If it is <code>TRUE</code> then it will run the MCMC algorithm to estimate the parameters. If it is <code>FALSE</code> , then the user needs to give input the parameter values in the <code>pars</code> argument.
<code>pars</code>	<code>= NULL</code> (default): If <code>estimate = FALSE</code> , then the user needs to input the parameter estimates.
<code>data_test</code>	<code>NULL</code> (default): Otherwise need to give the test data for comparing with the model estimates.
<code>auto.initialize</code>	<code>TRUE</code> (default): This is the option for using a mle based initial parameter.
<code>...</code>	arguments passed to the function <code>SEIRfansy</code> , <code>model_initializeR</code> and <code>model_plotR</code> which are used for initializing the parameters. The parameters are described below: <ul style="list-style-type: none"> • <code>niter</code>: <code>1e5</code> (default): Number of iteration for the MCMC metropolis hastig algorithm. • <code>BurnIn</code>: <code>5e4</code> (default) This is the Burn-In Period for the MCMC algorithm. • <code>model</code>: "Multinomial" (default): This is the likelihood function that will be used. There are three options available including "Multinomial", "Poisson", and "Binomial". • <code>alpha_p</code>: <code>0.5</code> (default): It is defined as the ratio of rate of spread of infection by tested positive patients to that by false negatives. We have taken $\alpha_p < 1$ as patients who are tested positive are subjected to quarantine where the chance of spreading the disease is less than that of false negative patients who are mostly unaware of their infectious nature. So, false negative individuals continue to spread the disease freely at a higher rate than tested positive individuals. • <code>alpha_u</code>: <code>0.7</code> (default): It is defined as the scaling factor for the rate of spread of infection by untested individuals. α_u is assumed to be < 1 as <code>U</code> mostly consists of asymptomatic or mildly symptomatic cases who are known to spread the disease at a much lower rate than those with higher levels of symptoms. • <code>beta_1</code>: <code>0.6</code> (default): It is the scaling factor for rate of recovery for untested individuals. β_1 is assumed to be less than 1. The condition of Untested individuals is not so severe as they consist of mostly asymptomatic people. So, they are assumed to recover faster than the Current Positive ones.

- `beta_2`: 0.7 (default): It is the inverse of the scaling factor for rate of recovery for false negative individuals. `$beta_2$` is assumed to be less than 1. It is assumed that the recovery rate is slower than the detected ones for the False Negative ones because they are not getting any hospital treatment.
- `delta_1`: 0.3 (default): It is the scaling factor for death rate for undetected individuals. `$delta_1$` is assumed to be less than 1. Similarly, for the Untested ones, the death rate is taken to be lesser because they are mostly asymptomatic. So, their probability of dying is much less.
- `delta_2`: 0.7 (default): It is the inverse of the scaling factor for death rate for false negative individuals. `$delta_2$` is assumed to be less than 1. Same as before, the death rate for False Negative ones are assumed to be higher than the Current detected Positive as they are not receiving proper treatment.
- `lambda`: $1 / (66.26 * 365)$ (default): Natural birth rate. The value given here as the default value is the world's common birth rate.
- `mu`: $1 / (66.26 * 365)$ (default): Natural death rates. This is assumed to be equal with natural birth rate for the sake of simplicity.
- `D_e`: 5.2 (default): Incubation period.
- `Dr`: 17.8 (default): Mean days until recovery for positive individuals.
- `f`: 0.15 (default): False negative probability of RT-PCR test.
- `mCFR`: NULL (default): (It is calculated from the data by default) It is defined as the ratio of the total reported deceased cases and the total removed cases until that day.
- `init.exposed.ratio`: 3 (default): This is the scaling factor for the calculation of the initial number of exposed people from the sum of the initial number of unreported, reported people.
- `init.confirmed.ratio`: 0.15 (default): This is the initial value of the probability of being tested.
- `opt_num`: 100 (default): The number of times an user wants to run the mle optimization before deciding on the best initial parameter.
- `trace_plot.common_axis`: FALSE (default): This will give the trace plot for the convergence of the MCMC estimated time varying parameters.
- `save plot`: TRUE (default): It is the option for saving the plots in the directory folder.

Value

A list with class "SEIRfansyPredict", which contains the items described below:

- `mcmc_pars`: a matrix of the mcmc draws for the parameters
- `plots`: a list of ggplot objects

Examples

```
library(dplyr)
train = covid19[which(covid19$Date == "01 April "):which(covid19$Date == "30 June "),]
test = covid19[which(covid19$Date == "01 July "):which(covid19$Date == "31 July "),]
```

```

train_multinom =
  train %>%
    rename(Confirmed = Daily.Confirmed,
           Recovered = Daily.Recovered,
           Deceased = Daily.Deceased) %>%
    dplyr::select(Confirmed, Recovered, Deceased)

test_multinom =
  test %>%
    rename(Confirmed = Daily.Confirmed,
           Recovered = Daily.Recovered,
           Deceased = Daily.Deceased) %>%
    dplyr::select(Confirmed, Recovered, Deceased)

N = 1341e6 # population size of India
data_initial = c(2059, 169, 58, 424, 9, 11)
pars_start = c(c(1,0.8,0.6,0.4,0.2), c(0.2,0.2,0.2,0.25,0.2))
phases = c(1,15,34,48,62)

cov19pred = SEIRfansy.predict(data = train_multinom, init_pars = pars_start,
                             data_init = data_initial, T_predict = 60, niter = 1e3,
                             BurnIn = 1e2, data_test = test_multinom, model = "Multinomial",
                             N = N, lambda = 1/(69.416 * 365), mu = 1/(69.416 * 365),
                             period_start = phases, opt_num = 1,
                             auto.initialize = TRUE, f = 0.15)

names(cov19pred)
class(cov19pred$prediction)
class(cov19pred$mcmc_pars)
names(cov19pred$plots)

plot(cov19pred, type = "trace")
plot(cov19pred, type = "boxplot")
plot(cov19pred, type = "panel")
plot(cov19pred, type = "cases")

# quick test for package check
# not for use outside CRAN check()
cov19est = SEIRfansy(data = train_multinom, init_pars = pars_start,
                    data_init = data_initial, niter = 33, BurnIn = 18,
                    model = "Multinomial", N = N, lambda = 1/(69.416 * 365),
                    mu = 1/(69.416 * 365), period_start = phases, opt_num = 1,
                    auto.initialize = TRUE, f = 0.15, plot = FALSE, system_test = NULL)

```

Index

* datasets

covid19, [2](#)

covid19, [2](#)

plot.SEIRfancy, [2](#)

plot.SEIRfancyPredict, [3](#)

SEIRfancy, [3](#)

SEIRfancy.predict, [7](#)