

Package ‘modelSSE’

September 7, 2023

Title Modelling Infectious Disease Superspreading from Contact Tracing Data

Version 0.1-3

Description Comprehensive analytical tools are provided to characterize infectious disease superspreading from contact tracing surveillance data. The underlying theoretical frameworks of this toolkit include branching process with transmission heterogeneity (Lloyd-Smith et al. (2005) <[doi:10.1038/nature04153](https://doi.org/10.1038/nature04153)>), case cluster size distribution (Nishiura et al. (2012) <[doi:10.1016/j.jtbi.2011.10.039](https://doi.org/10.1016/j.jtbi.2011.10.039)>), Blumberg et al. (2014) <[doi:10.1371/journal.ppat.1004452](https://doi.org/10.1371/journal.ppat.1004452)>, and Kucharski and Althaus (2015) <[doi:10.2807/1560-7917.ES2015.20.25.21167](https://doi.org/10.2807/1560-7917.ES2015.20.25.21167)>), and decomposition of reproduction number (Zhao et al. (2022) <[doi:10.1371/journal.pcbi.1010281](https://doi.org/10.1371/journal.pcbi.1010281)>).

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 3.0.0)

LazyData true

Imports Delaporte

NeedsCompilation no

Author Shi Zhao [aut, cre] (<<https://orcid.org/0000-0001-8722-6149>>)

Maintainer Shi Zhao <zhaoshi.cmsa@gmail.com>

Repository CRAN

Date/Publication 2023-09-07 10:30:02 UTC

R topics documented:

COVID19_JanApr2020_HongKong	2
d_nextgenclusterdistn	3
d_offspringdistn	5
d_outbreakdistn	8
d_reproductiondistn	12
MERS_2013_MEregion	14

mpox_19801984_DRC	15
overalllikelihood	16
paraest.MCMC	19
paraest.ML	24
smallpox_19581973_Europe	28
tailoffspringQ	29

Index	32
--------------	-----------

COVID19_JanApr2020_HongKong

A dataset of COVID-19 outbreak in Hong Kong

Description

This dataset (i.e., COVID19_JanApr2020_HongKong) contains 290 observations of **offspring case number** generated by one seed case, which were collected during the coronavirus disease 2019 (COVID-19) outbreak in Hong Kong, China from January to April 2020.

Usage

COVID19_JanApr2020_HongKong

Format

A data frame (`data.frame`) with 290 rows of records, and 2 columns of variables:

`obs` Observations of offspring (or secondary) case numbers generated by each seed case.

`type` A categorical variable takes the value "secondary" indicating the type of observations.

Source

Summary of data and outbreak situation of the Severe Respiratory Disease associated with a Novel Infectious Agent, released by the Centre for Health Protection, Department of Health, the Government of the Hong Kong Special Administrative Region. See for example, <https://www.coronavirus.gov.hk/eng/index.html>

References

Adam DC, Wu P, Wong JY, Lau EH, Tsang TK, Cauchemez S, Leung GM, Cowling BJ. Clustering and superspreading potential of SARS-CoV-2 infections in Hong Kong. *Nature Medicine*. 2020;26(11):1714-1719. doi:10.1038/s4159102010920

Examples

```
data(COVID19_JanApr2020_HongKong)
summary(COVID19_JanApr2020_HongKong)
table(COVID19_JanApr2020_HongKong$obs)
```

d_nextgenclusterdistn *The next-generation cluster size distribution*

Description

Density, cumulative distribution, quantile, and random variable generating functions for the **next-generation cluster size** distribution with pre-defined epidemiological parameters.

Usage

```
d_nextgenclusterdistn(  
  x = 5,  
  seed.size = 1,  
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),  
  offspring.type = "D",  
  is.log = FALSE  
)
```

```
p_nextgenclusterdistn(  
  q = 10.5,  
  seed.size = 1,  
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),  
  offspring.type = "D",  
  lower.tail = TRUE,  
  is.log = FALSE  
)
```

```
q_nextgenclusterdistn(  
  p = 0.8,  
  seed.size = 1,  
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),  
  offspring.type = "D",  
  lower.tail = TRUE  
)
```

```
r_nextgenclusterdistn(  
  n = 10,  
  seed.size = 1,  
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),  
  offspring.type = "D"  
)
```

Arguments

x A scalar, or a vector of positive integer, for the next-generation cluster size. The value of x must be not less than seed.size.

seed.size	A scalar, or a vector of positive integer. For vector type of seed.size, it only applies to d_nextgenclusterdistn(), p_nextgenclusterdistn(), and q_nextgenclusterdistn(). If seed.size and x, q or p are vectors, seed.size should be of the same length as x, q or p.
epi.para	A list (list) of pre-defined epidemiological parameters for offspring distribution, in the format of list(mean = ?, disp = ?, shift = ?), where the three parameters accept non-negative values. Each parameter can be either a scalar, or a vector. For the parameters being assigned with values of vectors, the vectors should be of the same length. For Delaporte distribution, the value of mean should be larger than the value of shift.
offspring.type	A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values: <ul style="list-style-type: none"> • "D" indicates the <i>Delaporte</i> distribution, • "NB" indicates the <i>negative binomial</i> distribution, • "G" indicates the <i>geometric</i> distribution, or • "P" indicates the <i>Poisson</i> distribution. By default, offspring.type = 'D'.
is.log	A logical variable, under which probability would be taken natural logarithm, if is.log = TRUE. By default, is.log = FALSE.
q	A scalar, or a vector of positive number (not necessarily integer), for the next-generation cluster size. The value of q must be not less than seed.size.
lower.tail	A logical variable, under which the probability is cumulative distribution function (CDF, i.e., $P(X \leq x)$), if lower.tail = TRUE, and otherwise, 1 - CDF (i.e., $P(X > x)$). By default, lower.tail = TRUE.
p	A scalar, or a vector of probability (i.e., ranging from 0 to 1).
n	A scalar of positive integer.

Details

Function d_nextgenclusterdistn() returns the probability of having a next-generation case cluster with size x generated by seed.size index cases, where (seed.size) is given.

Function p_nextgenclusterdistn() returns the probability of having a next-generation case cluster with size less than or equal to, or larger than q (depending on the value of lower.tail), generated by seed.size index cases, where (seed.size) is given.

Function q_nextgenclusterdistn() returns a value such that there is a probability of p for having a next-generation case cluster with size less than or equal to, or larger than this value (depending on the value of lower.tail) generated by seed.size index cases, where (seed.size) is given.

Function r_nextgenclusterdistn() returns a set of random variables of n next-generation cluster size, given (seed.size).

Value

For the values returned from the four functions,

- d_nextgenclusterdistn() is the *probability mass function* (PMF), and it returns value of probability (i.e., ranging from 0 to 1);

- `p_nextgenclusterdistn()` is the *cumulative distribution function* (CDF), and it returns value of probability (i.e., ranging from 0 to 1);
- `q_nextgenclusterdistn()` is the *quantile function*, and it returns value of quantile (positive integer); and
- `r_nextgenclusterdistn()` is the *random variable generating function*, and it generates a set of `n` random variables (positive integers).

Note

Depending on the values of parameters, the functions could take hours to complete, given the double-summation nature for the Delaporte distribution.

References

Blumberg S, Lloyd-Smith JO. Inference of R_0 and transmission heterogeneity from the size distribution of stuttering chains. *PLoS Computational Biology*. 2013 May 2;9(5):e1002993. doi:10.1371/journal.pcbi.1002993

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

See Also

[Delaporte](#) for the parameterization of Delaporte distribution.

Examples

```
## Please see the "Usage" section.
```

d_offspringdistn	<i>The offspring distribution</i>
------------------	-----------------------------------

Description

Density, cumulative distribution, quantile, and random variable generating functions for the **offspring** distribution with pre-defined epidemiological parameters.

Usage

```
d_offspringdistn(
  x = 1,
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  is.log = FALSE
)
```

```

p_offspringdistn(
  q = 1.5,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  is.log = FALSE,
  lower.tail = TRUE
)

q_offspringdistn(
  p = 0.8,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  lower.tail = TRUE
)

r_offspringdistn(
  n = 10,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D"
)

```

Arguments

x	A scalar, or a vector of non-negative integer.
epi.para	A list (list) of pre-defined epidemiological parameters for offspring distribution, in the format of <code>list(mean = ?, disp = ?, shift = ?)</code> , where the three parameters accept non-negative values. Each parameter can be either a scalar, or a vector. For the parameters being assigned with values of vectors, the vectors should be of the same length. For Delaporte distribution, the value of mean should be larger than the value of shift.
offspring.type	A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values: <ul style="list-style-type: none"> • "D" indicates the <i>Delaporte</i> distribution, • "NB" indicates the <i>negative binomial</i> distribution, • "G" indicates the <i>geometric</i> distribution, or • "P" indicates the <i>Poisson</i> distribution. By default, <code>offspring.type = 'D'</code> .
is.log	A logical variable, under which probability would be taken natural logarithm, if <code>is.log = TRUE</code> . By default, <code>is.log = FALSE</code> .
q	A scalar, or a vector of non-negative number (not necessarily integer).
lower.tail	A logical variable, under which the probability is cumulative distribution function (CDF, i.e., $P(X \leq x)$), if <code>lower.tail = TRUE</code> , and otherwise, $1 - \text{CDF}$ (i.e., $P(X > x)$). By default, <code>lower.tail = TRUE</code> .
p	A scalar, or a vector of probability (i.e., ranging from 0 to 1).
n	A scalar of positive integer.

Details

For different values of `offspring.type`,

- When `offspring.type = "D"`, no action to any parameter;
- When `offspring.type = "NB"`, we set parameter `shift = 0`;
- When `offspring.type = "G"`, we set parameters `disp = 1` and `shift = 0`; and
- When `offspring.type = "P"`, we set parameters `disp = +Inf` and `shift = mean`.

Value

For the values returned from the four functions,

- `d_offspringdistn()` is the *probability mass function* (PMF), and it returns value of probability (i.e., ranging from 0 to 1);
- `p_offspringdistn()` is the *cumulative distribution function* (CDF), and it returns value of probability (i.e., ranging from 0 to 1);
- `q_offspringdistn()` is the *quantile function*, and it returns value of quantile (non-negative integer); and
- `r_offspringdistn()` is the *random variable generating function*, and it generates a set of `n` random variables (non-negative integers).

Note

Depending on the values of parameters, the functions could take hours to complete, given the double-summation nature for the Delaporte distribution.

References

Vose D. Risk analysis: a quantitative guide. John Wiley & Sons. 2008; pp. 618-619. ISBN: 978-0-470-51284-5

Lloyd-Smith JO, Schreiber SJ, Kopp PE, Getz WM. Superspreading and the effect of individual variation on disease emergence. *Nature*. 2005;438(7066):355-359. doi:10.1038/nature04153

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

See Also

[Delaporte](#) for the parameterization of Delaporte distribution.

Examples

```
## Please see the "Usage" section.
```

```
## the following returns the proportion of index cases that generated at least 1 offspring cases.
p_offspringdistn(
  q = 0,
```

```

    epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
    offspring.type = 'D', lower.tail = FALSE
  )

  ## reproducing the results in Adam, et al. (2020)
  ## paper doi link: https://doi.org/10.1038/s41591-020-1092-0 (see Fig 3b),
  ## where the number of offspring cases were fitted
  ## with parameter R of 0.58 and k of 0.43 under NB distribution.
  data(COVID19_JanApr2020_HongKong)
  hist(
    COVID19_JanApr2020_HongKong$sobs, breaks = c(0:100) -0.5, xlim = c(0,12),
    freq = FALSE, xlab = 'secondary cases', ylab = 'rel. freq.', main = ''
  )
  lines(0:12, d_offspringdistn(
    x = 0:12,
    epi.para = list(mean = 0.58, disp = 0.43, shift = 0.2),
    offspring.type = "NB"
  ), pch = 20, type = 'o', lty = 2)

  ## an example to generate 100 rv of offspring case number
  table(r_offspringdistn(
    n = 100,
    epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
    offspring.type = 'D'
  ))

```

d_outbreakdistn

The final outbreak size distribution

Description

Density, cumulative distribution, quantile, and random variable generating functions for the **final outbreak size** distribution with pre-defined epidemiological parameters.

Usage

```

d_outbreakdistn(
  x = 10,
  seed.size = 1,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  is.log = FALSE
)

p_outbreakdistn(
  q = 30.5,
  seed.size = 1,

```



```

    epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
    offspring.type = "D",
    lower.tail = TRUE,
    is.log = FALSE
  )

q_outbreakdistn(
  p = 0.8,
  seed.size = 1,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  lower.tail = TRUE,
  upr.limit = 1000
)

r_outbreakdistn(
  n = 10,
  seed.size = 1,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  upr.limit = 1000
)

```

Arguments

x	A scalar, or a vector of final outbreak size, which is positive integer. The value of x must be not less than seed.size.
seed.size	A scalar, or a vector of positive integer. For vector type of seed.size, it only applies to d_outbreakdistn(), and p_outbreakdistn(). If seed.size and x or q are vectors, seed.size should be of the same length as x or q.
epi.para	A list (list) of pre-defined epidemiological parameters for offspring distribution, in the format of list(mean = ?, disp = ?, shift = ?), where the three parameters accept non-negative values. Each parameter must be a scalar. For Delaporte distribution, the value of mean should be larger than the value of shift.
offspring.type	A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values: <ul style="list-style-type: none"> • "D" indicates the <i>Delaporte</i> distribution, • "NB" indicates the <i>negative binomial</i> distribution, • "G" indicates the <i>geometric</i> distribution, or • "P" indicates the <i>Poisson</i> distribution. By default, offspring.type = 'D'.
is.log	A logical variable, under which probability would be taken natural logarithm, if is.log = TRUE. By default, is.log = FALSE.
q	A scalar, or a vector of positive number (not necessarily integer). The value of q must be not less than seed.size.

lower.tail	A logical variable, under which the probability is cumulative distribution function (CDF, i.e., $P(X \leq x)$), if lower.tail = TRUE, and otherwise, 1 - CDF (i.e., $P(X > x)$). By default, lower.tail = TRUE.
p	A scalar, or a vector of probability (i.e., ranging from 0 to 1).
upr.limit	A positive integer. If the result final outbreak size is larger than upr.limit, the returned value will be set as "> upr.limit". The value of upr.limit must be larger than seed.size. By default, upr.limit = 1000, and no need to change the default setting here unless for special reasons.
n	A scalar of positive integer.

Details

Function d_outbreakdistn() returns the probability of having an outbreak with final size x generated by seed.size index cases, where (seed.size) is given.

Function p_outbreakdistn() returns the probability of having an outbreak with final size less than or equal to, or larger than q (depending on the value of lower.tail), generated by seed.size index cases, where (seed.size) is given.

Function q_outbreakdistn() returns a value such that there is a probability of p for having a final outbreak size less than or equal to, or larger than this value (depending on the value of lower.tail) generated by seed.size index cases, where (seed.size) is given.

Function r_outbreakdistn() returns a set of n random variables of final outbreak size, given (seed.size).

Value

For the values returned from the four functions,

- d_outbreakdistn() is the *probability mass function* (PMF), and it returns value of probability (i.e., ranging from 0 to 1);
- p_outbreakdistn() is the *cumulative distribution function* (CDF), and it returns value of probability (i.e., ranging from 0 to 1);
- q_outbreakdistn() is the *quantile function*, and it returns value of quantile (positive integer); and
- r_outbreakdistn() is the *random variable generating function*, and it generates a set of n random variables (positive integers).

Specially, due to the computational consumption, an upper limit was set for the functions q_outbreakdistn() and r_outbreakdistn(), i.e., upr.limit, and thus, both functions here return value in string form (i.e., character).

Note

Each parameter in epi.para = list(mean = ?, disp = ?, shift = ?) should be a scalar, which means vector is not allowed here.

When q is large, e.g., $q > 10000$, the function p_outbreakdistn() could take few seconds, or even minutes to complete.

When `upr.limit` is large, e.g., `upr.limit > 10000`, the functions `q_outbreakdistn()` and `r_outbreakdistn()` could take few seconds, or even minutes to complete. Thus, we do not recommend the users to change the default setting of `upr.limit` unless for special reasons.

References

Farrington CP, Kanaan MN, Gay NJ. Branching process models for surveillance of infectious diseases controlled by mass vaccination. *Biostatistics*. 2003;4(2):279-95. doi:10.1093/biostatistics/4.2.279

Nishiura H, Yan P, Sleeman CK, Mode CJ. Estimating the transmission potential of supercritical processes based on the final size distribution of minor outbreaks. *Journal of Theoretical Biology*. 2012;294:48-55. doi:10.1016/j.jtbi.2011.10.039

Blumberg S, Funk S, Pulliam JR. Detecting differential transmissibilities that affect the size of self-limited outbreaks. *PLoS Pathogens*. 2014;10(10):e1004452. doi:10.1371/journal.ppat.1004452

Kucharski AJ, Althaus CL. The role of superspreading in Middle East respiratory syndrome coronavirus (MERS-CoV) transmission. *Eurosurveillance*. 2015;20(25):21167. doi:10.2807/1560-7917.ES2015.20.25.21167

Endo A, Abbott S, Kucharski AJ, Funk S. Estimating the overdispersion in COVID-19 transmission using outbreak sizes outside China. *Wellcome Open Research*. 2020;5:67. doi:10.12688/wellcomeopenres.15842.3

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

See Also

[d_offspringdistn](#)

Examples

```
## an example to generate 1000 rv of final outbreak size
table(r_outbreakdistn(
  n = 1000,
  seed.size = 1,
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  upr.limit = 10
))
```

```
## an attempt to reproduce the results in Guo, et al. (2022)
## paper doi link: https://doi.org/10.1016/j.jinf.2022.05.041 (see Fig 1B),
## where the probability of one seed case generating an outbreak with final size >= a given number,
## with parameter R of 0.78 and k of 0.10 under NB distribution.
plot(1:100, 1 - c(0,cumsum(d_outbreakdistn(
  x = 1:99,
```

```

seed.size = 1,
epi.para = list(mean = 0.78, disp = 0.10, shift = 0.2),
offspring.type = "NB",
)), log = 'y', type = 'l', xlab = 'outbreak size', ylab = 'probability')
plot(1:100, c(1,p_outbreakdistn(
  q = 1:99,
  seed.size = 1,
  epi.para = list(mean = 0.78, disp = 0.10, shift = 0.2),
  offspring.type = "NB",
  lower.tail = FALSE
)), log = 'y', type = 'l', xlab = 'outbreak size', ylab = 'probability')

```

d_reproductiondistn *The distribution of individual reproduction number*

Description

This function (i.e., `d_reproductiondistn()`) is the probability density function (PDF) of **individual reproduction number** that was modelled as a shifted gamma distribution.

Usage

```

d_reproductiondistn(
  x = 1,
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  is.log = FALSE
)

```

Arguments

- | | |
|-----------------------------|--|
| <code>x</code> | A scalar, or a vector of non-negative integer. |
| <code>epi.para</code> | A list (list) of pre-defined epidemiological parameters for offspring distribution, in the format of <code>list(mean = ?, disp = ?, shift = ?)</code> , where the three parameters accept non-negative values. Each parameter can be either a scalar, or a vector. For the parameters being assigned with values of vectors, the vectors should be of the same length. For Delaporte distribution, the value of mean should be larger than the value of shift. |
| <code>offspring.type</code> | A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values: <ul style="list-style-type: none"> • "D" indicates the Delaporte distribution for offspring cases, where reproduction number follows a <i>shifted gamma</i> distribution; • "NB" indicates the negative binomial distribution for offspring cases, where reproduction number follows a (non-shifted, or standard) <i>gamma</i> distribution; |

- "G" indicates the geometric distribution for offspring cases, where reproduction number follows an *exponential* distribution; or
- "P" indicates the Poisson distribution for offspring cases, where reproduction number follows a *Dirac delta* distribution.

By default, `offspring.type = 'D'`.

`is.log` A logical variable, under which probability would be taken natural logarithm, if `is.log = TRUE`. By default, `is.log = FALSE`.

Value

`d_reproductiondistn()` is the probability density function (PDF), and it returns value of probability density (non-negative value).

Note

Only the PDF of individual reproduction number (i.e., `d_reproductiondistn()`) was created here (without cumulative distribution, quantile, or random variable generating functions). The function `d_reproductiondistn()` was used mainly for data visualization purpose (rather than using for analysis), because the distribution of individual reproduction number was not explicitly used in model fitting to the disease contact tracing data.

When `offspring.type = "P"`, individual reproduction number follows a *Dirac delta* distribution, which is difficult to return any value, or visualize the PDF, because of the nature of this pulse function.

References

Lloyd-Smith JO, Schreiber SJ, Kopp PE, Getz WM. Superspreading and the effect of individual variation on disease emergence. *Nature*. 2005;438(7066):355-359. doi:10.1038/nature04153

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

Examples

```
## an example to visualize individual reproduction number is as follows.
plot(seq(0.01,9.99, length.out = 1001), d_reproductiondistn(
  x = seq(0,10, length.out = 1001),
  epi.param = list(mean = 2, disp = 1.5, shift = 0.5),
  offspring.type = "D",
  is.log = FALSE
), type = 'l', xlab = 'individual reproduction number', ylab = 'density')
```

MERS_2013_MEregion *A dataset of MERS outbreaks in the Middle East region*

Description

This dataset (i.e., MERS_2013_MEregion) contains 55 observations of **final outbreak size** generated by given numbers of seed cases, which were collected in Middle East respiratory syndrome (MERS) outbreaks in the Middle East (ME) regions in 2013.

Usage

```
MERS_2013_MEregion
```

Format

A data frame (`data.frame`) with 55 rows of records, and 3 columns of variables:

`obs.seed` Observations of the number of seed cases that generated the outbreak.

`obs.finalsize` Observations of final outbreak size generated by given numbers (given in `obs.seed`) of seed cases.

`type` A categorical variable takes the value "outbreaksize" indicating the type of observations.

References

Poletto C, Pelat C, Levy-Bruhl D, Yazdanpanah Y, Boelle PY, Colizza V. Assessment of the Middle East respiratory syndrome coronavirus (MERS-CoV) epidemic in the Middle East and risk of international spread using a novel maximum likelihood analysis approach. *Eurosurveillance*. 2014;19(23):20824. doi:10.2807/15607917.ES2014.19.23.20824, see the "baseline" column in Table 1 for the raw data.

Kucharski AJ, Althaus CL. The role of superspreading in Middle East respiratory syndrome coronavirus (MERS-CoV) transmission. *Eurosurveillance*. 2015;20(25):21167. doi:10.2807/1560-7917.ES2015.20.25.21167

Examples

```
data(MERS_2013_MEregion)
summary(MERS_2013_MEregion)
```

mpox_19801984_DRC *A dataset of mpox outbreaks in DRC*

Description

This dataset (i.e., mpox_19801984_DRC) contains 125 observations of one of the following types:

- **offspring case number**,
- **next-generation cluster size**, and
- **final outbreak size**,

generated by given numbers of seed cases, which were collected in mpox (i.e., monkeypox) outbreaks in Democratic Republic of the Congo (DRC, or previous named "Zaire" before 1997) from 1980 to 1984.

Usage

mpox_19801984_DRC

Format

A data frame (`data.frame`) with 125 rows of records, and 3 columns of variables:

`obs.seed` Observations of the number of seed cases that generated the offspring cases in the next transmission generation.

`obs.size` Observations of cases numbers generated by given numbers (given in `obs.seed`) of seed cases, see type variable for the detailed meaning of the value recorded under this variable here.

`type` A categorical variable takes one of the 3 values indicating the type of observations for `obs.size` as follows:

- "offpring" indicated the **offspring case number**, generated by 1 seed case for each observation.
- "nextgen" indicated the **next-generation cluster size** (including seed cases), generated by a group of seed cases with size given in `obs.seed`.
- "outbreak" indicated the **final outbreak size** (including seed cases), generated by a group of seed cases with size given in `obs.seed`.

Details

Note one difference between mpox_19801984_DRC and the original dataset (i.e., the dataset presented in reference) was that for simplicity, observations in the original dataset that involved transmission chains of more than 1 generation were aggregated as **final outbreak size** observations in mpox_19801984_DRC.

References

Fine PE, Jezek Z, Grab B, Dixon H. The transmission potential of monkeypox virus in human populations. *International Journal of Epidemiology*. 1988;17(3):643-650. doi:10.1093/ije/17.3.643, see Table 1 for the raw data.

Examples

```
data(mpox_19801984_DRC)
summary(mpox_19801984_DRC)
table(mpox_19801984_DRC$type)
```

overalllikelihood *The likelihood function*

Description

This function (i.e., `overalllikelihood()`) calculates the likelihood value with a list of pre-defined epidemiological parameters and a given structured contact tracing data.

Usage

```
overalllikelihood(
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  is.log = TRUE,
  data = NULL,
  var.name = list(obssize = NULL, seedsize = NULL, typelab = NULL),
  obs.type.lab = list(offspring = NULL, nextgen = NULL, outbreak = NULL)
)
```

Arguments

- | | |
|-----------------------------|---|
| <code>epi.param</code> | A list (<code>list</code>) of pre-defined epidemiological parameters for offspring distribution, in the format of <code>list(mean = ?, disp = ?, shift = ?)</code> , where the three parameters accept non-negative values. Each parameter must be a scalar. For Delaporte distribution, the value of mean should be larger than the value of shift. |
| <code>offspring.type</code> | A character label (<code>character</code>) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values: <ul style="list-style-type: none"> • "D" indicates the <i>Delaporte</i> distribution, • "NB" indicates the <i>negative binomial</i> distribution, • "G" indicates the <i>geometric</i> distribution, or • "P" indicates the <i>Poisson</i> distribution. By default, <code>offspring.type = 'D'</code> . |
| <code>is.log</code> | A logical variable, under which the likelihood would be taken natural logarithm, i.e., log-likelihood, if <code>is.log = TRUE</code> . By default, <code>is.log = TRUE</code> . |
| <code>data</code> | A data frame (<code>data.frame</code>), or a vector (only when <code>obs.type.lab = "offspring"</code>) that contains the structured contact tracing data. |

var.name	A list (list), or a character of variable name for the column names of dataset given in data. For a list of variable names, it should be in the format of <code>list(obssize = ?, seedsize = ?, typelab = ?)</code> . Please see the details section for more information. By default, <code>var.name = list(obssize = NULL, seedsize = NULL, typelab = NULL)</code> .
obs.type.lab	A list (list), or a character of labels (i.e., "offspring", "nextgen", or "outbreak") for the type of observations. For a list of labels, it should be in the format of <code>list(offspring = ?, nextgen = ?, outbreak = ?)</code> . Please see the details section for more information. By default, <code>obs.type.lab = list(offspring = NULL, nextgen = NULL, outbreak = NULL)</code> .

Details

When `obs.type.lab` is a character, it should be either "offspring", "nextgen", or "outbreak" for type of observations.

When `obs.type.lab` is a list, this occurs when the contact tracing data has more than one types of observations. See example 4 in the Examples section.

When the contact tracing dataset is offspring case observations, the function arguments `data` could be either a vector, or a data frame. If `data` is a vector, it is not necessary to assign any value to `var.name`. If `data` is a data frame, it is necessary to identify the variable name of offspring observations in `var.name`. See example 1 in the Examples section.

When the contact tracing dataset is next-generation cluster size, or final outbreak size observations, the variable names of both observations and seed case size should be identified in `var.name` with the format of `list(obssize = ?, seedsize = ?)`. See example 2 and example 3 in the Examples section.

When the contact tracing dataset has more than one types of observations, the variable names of observations, seed case size, and observation type should be identified in `var.name` with the format of `list(obssize = ?, seedsize = ?, typelab = ?)`. See example 4 in the Examples section.

Value

The log-likelihood (by default), or likelihood value from contact tracing data, with pre-defined epidemiological parameters.

Note

Each parameter in `epi.para = list(mean = ?, disp = ?, shift = ?)` should be a scalar, which means vector is not allowed here.

For the contact tracing data in `data`, unknown observations (i.e., NA) is not allowed.

References

Lloyd-Smith JO, Schreiber SJ, Kopp PE, Getz WM. Superspreading and the effect of individual variation on disease emergence. *Nature*. 2005;438(7066):355-359. doi:10.1038/nature04153

Nishiura H, Yan P, Sleeman CK, Mode CJ. Estimating the transmission potential of supercritical processes based on the final size distribution of minor outbreaks. *Journal of Theoretical Biology*. 2012;294:48-55. doi:10.1016/j.jtbi.2011.10.039

Blumberg S, Funk S, Pulliam JR. Detecting differential transmissibilities that affect the size of self-limited outbreaks. *PLoS Pathogens*. 2014;10(10):e1004452. doi:10.1371/journal.ppat.1004452

Kucharski AJ, Althaus CL. The role of superspreading in Middle East respiratory syndrome coronavirus (MERS-CoV) transmission. *Eurosurveillance*. 2015;20(25):21167. doi:10.2807/1560-7917.ES2015.20.25.21167

Endo A, Abbott S, Kucharski AJ, Funk S. Estimating the overdispersion in COVID-19 transmission using outbreak sizes outside China. *Wellcome Open Research*. 2020;5:67. doi:10.12688/wellcomeopenres.15842.3

Adam DC, Wu P, Wong JY, Lau EH, Tsang TK, Cauchemez S, Leung GM, Cowling BJ. Clustering and superspreading potential of SARS-CoV-2 infections in Hong Kong. *Nature Medicine*. 2020;26(11):1714-1719. doi:10.1038/s4159102010920

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

Examples

```
# example 1 #
## likelihood for the offspring observations
data(COVID19_JanApr2020_HongKong)
overalllikelihood(
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  data = COVID19_JanApr2020_HongKong,
  var.name = list(obssize = 'obs'),
  obs.type.lab = 'offspring'
)
overalllikelihood(
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
)

# example 2 #
## likelihood for the next-generation cluster size observations
data(smallpox_19581973_Europe)
overalllikelihood(
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = 'D',
  data = smallpox_19581973_Europe,
  var.name = list(obssize = 'obs.clustersize', seedsize = 'obs.seed'),
  obs.type.lab = 'nextgen'
)

# example 3 #
## likelihood for the final outbreak size observations
```

```

data(MERS_2013_MEregion)
overalllikelihood(
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = 'D',
  data = MERS_2013_MEregion,
  var.name = list(obssize = 'obs.finalsized', seedsize = 'obs.seed'),
  obs.type.lab = 'outbreak'
)

# example 4 #
## likelihood for more than one types of observations
data(mpox_19801984_DRC)
overalllikelihood(
  epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = 'D',
  data = mpox_19801984_DRC,
  var.name = list(obssize = 'obs.size', seedsize = 'obs.seed', typelab = 'type'),
  obs.type.lab = list(offspring = 'offspring', nextgen = 'nextgen', outbreak = 'outbreak')
)

# example 5 #
## reproducing the AIC results in Adam, et al. (2020) https://doi.org/10.1038/s41591-020-1092-0,
## (see Supplementary Table 4),
## where the AIC scores were calculated for NB, Geometric, and Poisson models from top to bottom.
## Here, the AIC is defined as:  $AIC = -2 * \log\text{-likelihood} + 2 * \text{number of unknown model parameters}$ .
data(COVID19_JanApr2020_HongKong)
overalllikelihood(
  epi.para = list(mean = 0.58, disp = 0.43, shift = 0.2),
  offspring.type = "NB",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
) * (-2) + 2*2
overalllikelihood(
  epi.para = list(mean = 0.63, disp = 0.43, shift = 0.2),
  offspring.type = "G",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
) * (-2) + 1*2
overalllikelihood(
  epi.para = list(mean = 0.58, disp = 0.43, shift = 0.2),
  offspring.type = "P",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
) * (-2) + 1*2

```

Description

This function (i.e., `paraest.MCMC()`) performs model parameter estimation using random walk **Markov chain Monte Carlo** (MCMC) approach with given structured contact tracing data.

Usage

```
paraest.MCMC(
  can.epi.para.start = list(mean = 1, disp = 0.5, shift = 0.2),
  isfix.epi.para = list(mean = FALSE, disp = FALSE, shift = FALSE),
  offspring.type = "D",
  para.comb.num = 10000,
  burnin.frac = 0.33,
  data = NULL,
  var.name = list(obssize = NULL, seedsize = NULL, typelab = NULL),
  obs.type.lab = list(offspring = NULL, nextgen = NULL, outbreak = NULL)
)
```

Arguments

- `can.epi.para.start` A list (list) of the starting values for unknown epidemiological parameters for offspring distribution. The list should be in the format of `list(mean = ?, disp = ?, shift = ?)`. Each parameter must be a scalar, and only accept non-negative values. The default setting is given in the code Usage section. For Delaporte distribution, the value of mean should be larger than the value of shift.
- `isfix.epi.para` A list (list) of logistic values that indicate whether each parameter is fixed. By default, `isfix.epi.para = list(mean = FALSE, disp = FALSE, shift = FALSE)`, where all three parameters are not fixed and to be estimated.
- `offspring.type` A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values:
- "D" indicates the *Delaporte* distribution,
 - "NB" indicates the *negative binomial* distribution,
 - "G" indicates the *geometric* distribution, or
 - "P" indicates the *Poisson* distribution.
- By default, `offspring.type = 'D'`.
- `para.comb.num` A positive integer for the number of iterations used for MCMC runs. By default, `para.comb.num = 10000`, and no need to change the default setting here unless for special reasons.
- `burnin.frac` A number with range strictly between 0 and 1 for the fraction of MCMC chain that will be discarded as the "burn-in" stage. By default, `burnin.frac = 0.33`, and no need to change the default setting here unless for special reasons.
- `data` A data frame (data.frame), or a vector (only when `obs.type.lab = "offspring"`) that contains the structured contact tracing data.
- `var.name` A list (list), or a character of variable name for the column names of dataset given in data. For a list of variable names, it should be in the format of

`list(obssize = ?, seedsize = ?, typelab = ?)`. Please see the details section for more information. By default, `var.name = list(obssize = NULL, seedsize = NULL, typelab = NULL)`.

`obs.type.lab` A list (`list`), or a character of labels (i.e., "offspring", "nextgen", or "outbreak") for the type of observations. For a list of labels, it should be in the format of `list(offspring = ?, nextgen = ?, outbreak = ?)`. Please see the details section for more information. By default, `obs.type.lab = list(offspring = NULL, nextgen = NULL, outbreak = NULL)`.

Details

For the values of parameters given in `can.epi.para.start`, they are some rough starting points for the MCMC to run, which are not necessarily to be precise (but have to be within a reasonable range), and they will be used as a "start" status to find the posterior MCMC samples.

When `obs.type.lab` is a character, it should be either "offspring", "nextgen", or "outbreak" for type of observations. When `obs.type.lab` is a list, this occurs when the contact tracing data has more than one type of observations.

When the contact tracing dataset is offspring case observations, the function arguments `data` could be either a vector, or a data frame. If `data` is a vector, it is not necessary to assign any value to `var.name`. If `data` is a data frame, it is necessary to identify the variable name of offspring observations in `var.name`.

When the contact tracing dataset is next-generation cluster size, or final outbreak size observations, the variable names of both observations and seed case size should be identified in `var.name` with the format of `list(obssize = ?, seedsize = ?)`.

When the contact tracing dataset has more than one type of observations, the variable names of observations, seed case size, and observation type should be identified in `var.name` with the format of `list(obssize = ?, seedsize = ?, typelab = ?)`.

Value

A list (i.e., `list`) contains the following three items:

- a data frame (`data.frame`) summarizing the median, and 2.5% and 97.5% percentiles (95% credible interval, CrI) of the posterior MCMC samples for each unknown parameter,
- the maximum log-likelihood value, and
- a data frame (`data.frame`) of the posterior MCMC samples and their corresponding log-likelihood values.

Note

Each parameter in `can.epi.para.start = list(mean = ?, disp = ?, shift = ?)` should be a scalar, which means vector is not allowed here.

For the contact tracing data in `data`, unknown observations (i.e., NA) is not allowed.

As `para.comb.num` is the number of iterations for the MCMC runs, when `para.comb.num` is large, e.g., `para.comb.num > 100000`, the function `paraest.MCMC()` could take few seconds, or even minutes to complete, depending on the sample size, etc. Thus, we do not recommend the users to change the default setting of `para.comb.num` unless for special reasons.

References

- Blumberg S, Funk S, Pulliam JR. Detecting differential transmissibilities that affect the size of self-limited outbreaks. *PLoS Pathogens*. 2014;10(10):e1004452. doi:10.1371/journal.ppat.1004452
- Kucharski AJ, Althaus CL. The role of superspreading in Middle East respiratory syndrome coronavirus (MERS-CoV) transmission. *Eurosurveillance*. 2015;20(25):21167. doi:10.2807/1560-7917.ES2015.20.25.21167
- Adam DC, Wu P, Wong JY, Lau EH, Tsang TK, Cauchemez S, Leung GM, Cowling BJ. Clustering and superspreading potential of SARS-CoV-2 infections in Hong Kong. *Nature Medicine*. 2020;26(11):1714-1719. doi:10.1038/s4159102010920
- Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

See Also

`modelSSE::overalllikelihood()`

Examples

```
# example 1: for offspring observations #
## reproducing the parameter estimation results in Adam, et al. (2020)
## paper doi link: https://doi.org/10.1038/s41591-020-1092-0,
## (see the first row in Supplementary Table 4),
## where R of 0.58 (95% CI: 0.45, 0.72), and k of 0.43 (95% CI: 0.29, 0.67).
data(COVID19_JanApr2020_HongKong)
set.seed(2023)
MCMC.output = paraest.MCMC(
  #can.epi.para.start = list(mean = 0.60, disp = 0.50, shift = 0.20),
  offspring.type = "NB", para.comb.num = 10000,
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
)
print(MCMC.output$epi.para.est.output)
## Then, plot the posterior fitting results using 100 randomly-selected MCMC samples.
hist(
  COVID19_JanApr2020_HongKong$obs, breaks = c(0:100) -0.5, xlim = c(0,12),
  freq = FALSE, xlab = 'secondary cases', ylab = 'rel. freq.', main = ''
)
for(jj in 1:100){
  temp.random.col.index = sample.int(n = nrow(MCMC.output$est.record.mat), size = 1)
  est.record.array = MCMC.output$est.record.mat[temp.random.col.index,]
  lines(0:12, d_offspringdistn(
    x = 0:12,
    epi.para = list(
      mean = est.record.array$epi.para.mean,
      disp = est.record.array$epi.para.disp, shift = 0.1
    ),
    offspring.type = "NB"
  )
}
```

```

    ), col = '#0066FF33', type = 'l', lty = 3)
}

# example 2: for offspring observations #
## reproducing the parameter estimation results in Zhao, et al. (2020)
## paper doi link: https://doi.org/10.1371/journal.pcbi.1010281,
## (see the results of dataset #3 using Delaporte distribution in Table 1), where
## R of 0.59 (95% CI: 0.46, 0.78),
## k of 0.16 (95% CI: 0.06, 0.40), and
## shift of 0.17 (95% CI: 0.04, 0.30).
data(COVID19_JanApr2020_HongKong)
set.seed(2023)
paraest.MCMC(
  can.epi.para.start = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
)$epi.para.est.output

# example 3: for next-generation cluster size observations #
## reproducing the parameter estimation results in Blumberg, et al, (2014)
## paper doi link: https://doi.org/10.1371/journal.ppat.1004452,
## (see the last row in Table 3, and Fig 4A),
## where R of 3.14 (95% CI: 2, >6), and k of 0.37 (95% CI: not reported).
data(smallpox_19581973_Europe)
set.seed(2023)
paraest.MCMC(
  can.epi.para.start = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "NB",
  data = smallpox_19581973_Europe,
  var.name = list(obssize = 'obs.clustersize', seedsize = 'obs.seed'),
  obs.type.lab = 'nextgen'
)$epi.para.est.output

# example 4: final outbreak size observations #
## reproducing the parameter estimation results in Kucharski, Althaus. (2015)
## paper doi link: https://doi.org/10.2807/1560-7917.ES2015.20.25.21167,
## (see Fig 1, and Finding section),
## where R of 0.47 (95% CI: 0.29, 0.80), and k of 0.26 (95% CI: 0.09, 1.24).
data(MERS_2013_MEregion)
set.seed(2023)
paraest.MCMC(
  can.epi.para.start = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "NB",
  data = MERS_2013_MEregion,
  var.name = list(obssize = 'obs.finalsized', seedsize = 'obs.seed'),
  obs.type.lab = 'outbreak'
)$epi.para.est.output

```

```
# example 5: for more than one types of observations #
## reproducing the parameter estimation results in Blumberg, et al, (2014)
## paper doi link: https://doi.org/10.1371/journal.ppat.1004452,
## (see the last row in Table 5, and Fig 6A),
## where R of 0.3 (95% CI: 0.2, 0.5), and k of 0.4 (95% CI: not reported).
data(mpox_19801984_DRC)
set.seed(2023)
paraest.MCMC(
  can.epi.para.start = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "NB", para.comb.num = 30000,
  data = mpox_19801984_DRC,
  var.name = list(obssize = 'obs.size', seedsize = 'obs.seed', typelab = 'type'),
  obs.type.lab = list(offspring = 'offspring', nextgen = 'nextgen', outbreak = 'outbreak')
)$epi.para.est.output
```

paraest.ML

To estimate model parameters using maximum likelihood approach

Description

This function (i.e., `paraest.ML()`) performs model parameter estimation using **maximum likelihood (ML)** approach with given structured contact tracing data.

Usage

```
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 2), disp = c(0.01, 2.5), shift = c(0.01, 0.5)),
  offspring.type = "D",
  para.comb.num = 1000,
  can.epi.para.set = NULL,
  data = NULL,
  var.name = list(obssize = NULL, seedsize = NULL, typelab = NULL),
  obs.type.lab = list(offspring = NULL, nextgen = NULL, outbreak = NULL)
)
```

Arguments

`can.epi.para.range`

A list (`list`) of ranges, or fixed values for unknown epidemiological parameters for offspring distribution. For the ranges of unknown epidemiological parameters, the list should be in the format of `list(mean = c(?, ?), disp = c(?, ?), shift = c(?, ?))`. For the fixed values of unknown epidemiological parameters, the list should be in the format of `list(mean = ?, disp = ?, shift = ?)`. Each parameter must be a scalar, and only accept non-negative values. The default setting is given in the code Usage section. For Delaporte distribution, the value of mean should be larger than the value of shift.

offspring.type	<p>A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values:</p> <ul style="list-style-type: none"> • "D" indicates the <i>Delaporte</i> distribution, • "NB" indicates the <i>negative binomial</i> distribution, • "G" indicates the <i>geometric</i> distribution, or • "P" indicates the <i>Poisson</i> distribution. <p>By default, offspring.type = 'D'.</p>
para.comb.num	<p>A positive integer for the number of parameter combinations used to construct log-likelihood profile. By default, para.comb.num = 1000, and no need to change the default setting here unless for special reasons.</p>
can.epi.para.set	<p>A data frame (data.frame) of different parameter combinations. The data frame must have three variables with names "epi.para.mean", "epi.para.disp", and "epi.para.shift" for the three parameters. By default, can.epi.para.set = NULL. Note that the function argument can.epi.para.set is usually used internally, and thus no need to change the default setting here unless for special reasons</p>
data	<p>A data frame (data.frame), or a vector (only when obs.type.lab = "offspring") that contains the structured contact tracing data.</p>
var.name	<p>A list (list), or a character of variable name for the column names of dataset given in data. For a list of variable names, it should be in the format of list(obssize = ?, seedsize = ?, typelab = ?). Please see the details section for more information. By default, var.name = list(obssize = NULL, seedsize = NULL, typelab = NULL).</p>
obs.type.lab	<p>A list (list), or a character of labels (i.e., "offspring", "nextgen", or "outbreak") for the type of observations. For a list of labels, it should be in the format of list(offspring = ?, nextgen = ?, outbreak = ?). Please see the details section for more information. By default, obs.type.lab = list(offspring = NULL, nextgen = NULL, outbreak = NULL).</p>

Details

For the ranges of parameters given in can.epi.para.range, they are some rough ranges, which are not necessarily to be precise (but have to be within a reasonable range), and they will be used as a "start" status to find the maximum likelihood estimate.

When obs.type.lab is a character, it should be either "offspring", "nextgen", or "outbreak" for type of observations. When obs.type.lab is a list, this occurs when the contact tracing data has more than one types of observations.

When the contact tracing dataset is offspring case observations, the function arguments data could be either a vector, or a data frame. If data is a vector, it is not necessary to assign any value to var.name. If data is a data frame, it is necessary to identify the variable name of offspring observations in var.name.

When the contact tracing dataset is next-generation cluster size, or final outbreak size observations, the variable names of both observations and seed case size should be identified in var.name with the format of list(obssize = ?, seedsize = ?).

When the contact tracing dataset has more than one types of observations, the variable names of observations, seed case size, and observation type should be identified in `var.name` with the format of `list(obssize = ?, seedsize = ?, typelab = ?)`.

Value

A list (i.e., `list`) contains the following three items:

- a data frame (`data.frame`) of the maximum likelihood estimate and 95% confidence interval (CI) of each unknown parameters,
- the maximum log-likelihood value, and
- a data frame (`data.frame`) of different parameter combinations and their corresponding log-likelihood values.

Note

For the contact tracing data in `data`, unknown observations (i.e., NA) is not allowed.

When `para.comb.num` is large, e.g., `para.comb.num > 10000`, the function `paraest.ML()` could take few seconds, or even minutes to complete, depending on the sample size, and model settings, etc. Thus, we do not recommend the users to change the default setting of `para.comb.num` unless for special reasons.

References

Blumberg S, Funk S, Pulliam JR. Detecting differential transmissibilities that affect the size of self-limited outbreaks. *PLoS Pathogens*. 2014;10(10):e1004452. doi:10.1371/journal.ppat.1004452

Kucharski AJ, Althaus CL. The role of superspreading in Middle East respiratory syndrome coronavirus (MERS-CoV) transmission. *Eurosurveillance*. 2015;20(25):21167. doi:10.2807/1560-7917.ES2015.20.25.21167

Adam DC, Wu P, Wong JY, Lau EH, Tsang TK, Cauchemez S, Leung GM, Cowling BJ. Clustering and superspreading potential of SARS-CoV-2 infections in Hong Kong. *Nature Medicine*. 2020;26(11):1714-1719. doi:10.1038/s4159102010920

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

See Also

[modelSSE::overalllikelihood\(\)](#)

Examples

```
## try to estimate the parameter (which is already known),
## using random samples generated from a geometric distribution with mean of 1.
set.seed(2020)
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 2.0), disp = c(0.01, 2.5), shift = c(0.01, 0.5)),
  offspring.type = "NB", para.comb.num = 100,
```

```

data = r_offspringdistn(
  n = 99, epi.para = list(mean = 1, disp = 0.5, shift = 0.2), offspring.type = "G"
),
obs.type.lab = 'offspring'
)$epi.para.est.output

# example 1: for offspring observations #
## reproducing the parameter estimation results in Adam, et al. (2020)
## paper doi link: https://doi.org/10.1038/s41591-020-1092-0,
## (see the first row in Supplementary Table 4),
## where R of 0.58 (95% CI: 0.45, 0.72), and k of 0.43 (95% CI: 0.29, 0.67).
data(COVID19_JanApr2020_HongKong)
set.seed(2020)
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 2.0), disp = c(0.01, 2.5), shift = c(0.01,0.5)),
  offspring.type = "NB",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
)$epi.para.est.output

# example 2: for offspring observations #
## reproducing the parameter estimation results in Zhao, et al. (2020)
## paper doi link: https://doi.org/10.1371/journal.pcbi.1010281,
## (see the results of dataset #3 using Delaporte distribution in Table 1), where
## R of 0.59 (95% CI: 0.46, 0.78),
## k of 0.16 (95% CI: 0.06, 0.40), and
## shift of 0.17 (95% CI: 0.04, 0.30).
data(COVID19_JanApr2020_HongKong)
set.seed(2020)
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 2.0), disp = c(0.01, 2.5), shift = c(0.01,0.5)),
  offspring.type = "D",
  data = COVID19_JanApr2020_HongKong$obs,
  obs.type.lab = 'offspring'
)$epi.para.est.output

# example 3: for next-generation cluster size observations #
## reproducing the parameter estimation results in Blumberg, et al, (2014)
## paper doi link: https://doi.org/10.1371/journal.ppat.1004452,
## (see the last row in Table 3, and Fig 4A),
## where R of 3.14 (95% CI: 2, >6), and k of 0.37 (95% CI: not reported).
data(smallpox_19581973_Europe)
set.seed(2020)
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 10.0), disp = c(0.01, 2.5), shift = c(0.01,0.5)),
  offspring.type = "NB",
  data = smallpox_19581973_Europe,
  var.name = list(obs.size = 'obs.clustersize', seed.size = 'obs.seed'),

```

```

  obs.type.lab = 'nextgen'
)$epi.para.est.output

# example 4: final outbreak size observations #
## reproducing the parameter estimation results in Kucharski, Althaus. (2015)
## paper doi link: https://doi.org/10.2807/1560-7917.ES2015.20.25.21167,
## (see Fig 1, and Finding section),
## where R of 0.47 (95% CI: 0.29, 0.80), and k of 0.26 (95% CI: 0.09, 1.24).
data(MERS_2013_MERegion)
set.seed(2020)
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 2.0), disp = c(0.01, 2.5), shift = c(0.01,0.5)),
  offspring.type = "NB",
  data = MERS_2013_MERegion,
  var.name = list(obssize = 'obs.finalsized', seedsize = 'obs.seed'),
  obs.type.lab = 'outbreak'
)$epi.para.est.output

# example 5: for more than one types of observations #
## reproducing the parameter estimation results in Blumberg, et al, (2014)
## paper doi link: https://doi.org/10.1371/journal.ppat.1004452,
## (see the last row in Table 5, and Fig 6A),
## where R of 0.3 (95% CI: 0.2, 0.5), and k of 0.4 (95% CI: not reported).
data(mpx_19801984_DRC)
set.seed(2020)
paraest.ML(
  can.epi.para.range = list(mean = c(0.1, 2.0), disp = c(0.01, 2.5), shift = c(0.01,0.5)),
  offspring.type = "NB",
  data = mpx_19801984_DRC,
  var.name = list(obssize = 'obs.size', seedsize = 'obs.seed', typelab = 'type'),
  obs.type.lab = list(offspring = 'offspring', nextgen = 'nextgen', outbreak = 'outbreak')
)$epi.para.est.output

```

smallpox_19581973_Europe

A dataset of smallpox outbreaks in Europe

Description

This dataset (i.e., smallpox_19581973_Europe) contains 34 observations of **next-generation cluster size** generated by given numbers of seed cases, which were collected in smallpox outbreaks in Europe from 1958 to 1973. Here, in this dataset, only the smallpox case observations involved in the *first* indigenous transmission generation were recorded.

Usage

```
smallpox_19581973_Europe
```

Format

A data frame (`data.frame`) with 34 of records, and 3 columns of variables:

`obs.seed` Observations of the number of seed cases that generated the offspring cases in the next transmission generation.

`obs.clustersize` Observations of next-generation cluster size generated by given numbers (given in `obs.seed`) of seed cases.

`type.obs` A categorical variable takes the value "nextgen_clustersize" indicating the type of observations.

References

Fenner F, Henderson DA, Arita I, Jezek Z, Ladnyi ID. Smallpox and its eradication. *Geneva: World Health Organization*. 1988. <https://apps.who.int/iris/handle/10665/39485>, see Table 23.4 for the raw data.

Blumberg S, Funk S, Pulliam JR. Detecting differential transmissibilities that affect the size of self-limited outbreaks. *PLoS Pathogens*. 2014;10(10):e1004452. doi:10.1371/journal.ppat.1004452, see Table 3 in the data supplementary file for the processed data.

Examples

```
data(smallpox_19581973_Europe)
summary(smallpox_19581973_Europe)
```

tailoffspringQ	<i>The "20/80" rule</i>
----------------	-------------------------

Description

To calculate proportion of (Q) offspring cases generated from proportion of (P) the most infectious index cases with pre-defined epidemiological parameters for the offspring distribution.

Usage

```
tailoffspringQ(
  P = 0.2,
  epi.param = list(mean = 1, disp = 0.5, shift = 0.2),
  offspring.type = "D",
  n.seed = 1000
)

mostinfectiousP(
```

```

Q = 0.8,
epi.para = list(mean = 1, disp = 0.5, shift = 0.2),
offspring.type = "D",
n.seed = 1000
)

```

Arguments

P, Q	A scalar, or a vector of probability (i.e., ranging from 0 to 1).
epi.para	A list (list) of pre-defined epidemiological parameters for offspring distribution, in the format of <code>list(mean = ?, disp = ?, shift = ?)</code> , where the three parameters accept non-negative values. Each parameter must be a scalar. For Delaporte distribution, the value of mean should be larger than the value of shift.
offspring.type	A character label (character) indicating the type of distribution used to describe the offspring distribution. It only accepts one of the following values: <ul style="list-style-type: none"> • "D" indicates the <i>Delaporte</i> distribution, • "NB" indicates the <i>negative binomial</i> distribution, • "G" indicates the <i>geometric</i> distribution, or • "P" indicates the <i>Poisson</i> distribution. By default, <code>offspring.type = 'D'</code> .
n.seed	A positive integer, for the number of seeds used to solve P or Q numerically. By default, <code>n.seed = 1000</code> , and no need to change the default setting here unless for special reasons.

Value

Function `tailoffspringQ()` returns the proportion of (Q) offspring cases generated from proportion of (P) index cases, where (P) is given.

Function `mostinfectiousP()` returns the proportion of (P) index cases that generated proportion of (Q) offspring cases, where (Q) is given.

Note

When `n.seed` is large, e.g., `n.seed > 100000`, the functions could take minutes to complete. As such, we do not recommend the users to change the default setting of `n.seed` unless for special reasons.

Each parameter in `epi.para = list(mean = ?, disp = ?, shift = ?)` should be a scalar, which means vector is not allowed here.

References

Lloyd-Smith JO, Schreiber SJ, Kopp PE, Getz WM. Superspreading and the effect of individual variation on disease emergence. *Nature*. 2005;438(7066):355-359. doi:10.1038/nature04153

Endo A, Abbott S, Kucharski AJ, Funk S. Estimating the overdispersion in COVID-19 transmission using outbreak sizes outside China. *Wellcome Open Research*. 2020;5:67. doi:10.12688/wellcomeopenres.15842.3

Adam DC, Wu P, Wong JY, Lau EH, Tsang TK, Cauchemez S, Leung GM, Cowling BJ. Clustering and superspreading potential of SARS-CoV-2 infections in Hong Kong. *Nature Medicine*. 2020;26(11):1714-9. doi:10.1038/s4159102010920

Zhao S, Chong MK, Ryu S, Guo Z, He M, Chen B, Musa SS, Wang J, Wu Y, He D, Wang MH. Characterizing superspreading potential of infectious disease: Decomposition of individual transmissibility. *PLoS Computational Biology*. 2022;18(6):e1010281. doi:10.1371/journal.pcbi.1010281

See Also

[d_offspringdistn](#)

Examples

```
## reproducing the results in Endo, et al. (2020) https://doi.org/10.12688/wellcomeopenres.15842.3,
## where ~80% offspring cases were generated from ~10% index cases
## with parameters R of ~2.5 (ranging from 2 to 3) and
## k of ~0.1 (ranging from 0.05 to 0.20) under NB distribution.
tailoffspringQ(
  P = 0.10,
  epi.param = list(mean = 2.5, disp = 0.10, shift = 0.2),
  offspring.type = "NB"
)
mostinfectiousP(
  Q = 0.80,
  epi.param = list(mean = 2.5, disp = 0.10, shift = 0.2),
  offspring.type = "NB"
)

## reproducing the results in Adam, et al. (2020) https://doi.org/10.1038/s41591-020-1092-0,
## where ~80% offspring cases were generated from ~19% index cases
## with parameters R of 0.58 and k of 0.43 under NB distribution.
tailoffspringQ(
  P = 0.19,
  epi.param = list(mean = 0.58, disp = 0.43, shift = 0.2),
  offspring.type = "NB"
)
mostinfectiousP(
  Q = 0.80,
  epi.param = list(mean = 0.58, disp = 0.43, shift = 0.2),
  offspring.type = "NB"
)
```

Index

* datasets

COVID19_JanApr2020_HongKong, [2](#)
MERS_2013_MEregion, [14](#)
mpox_19801984_DRC, [15](#)
smallpox_19581973_Europe, [28](#)

COVID19_JanApr2020_HongKong, [2](#)

d_nextgenclusterdistn, [3](#)
d_offspringdistn, [5](#), [11](#), [31](#)
d_outbreakdistn, [8](#)
d_reproductiondistn, [12](#)
Delaporte, [5](#), [7](#)

MERS_2013_MEregion, [14](#)
modelSSE::overalllikelihood(), [22](#), [26](#)
mostinfectiousP (tailoffspringQ), [29](#)
mpox_19801984_DRC, [15](#)

overalllikelihood, [16](#)

p_nextgenclusterdistn
 (d_nextgenclusterdistn), [3](#)
p_offspringdistn (d_offspringdistn), [5](#)
p_outbreakdistn (d_outbreakdistn), [8](#)
paraest.MCMC, [19](#)
paraest.ML, [24](#)

q_nextgenclusterdistn
 (d_nextgenclusterdistn), [3](#)
q_offspringdistn (d_offspringdistn), [5](#)
q_outbreakdistn (d_outbreakdistn), [8](#)

r_nextgenclusterdistn
 (d_nextgenclusterdistn), [3](#)
r_offspringdistn (d_offspringdistn), [5](#)
r_outbreakdistn (d_outbreakdistn), [8](#)

smallpox_19581973_Europe, [28](#)

tailoffspringQ, [29](#)