Package ‘Branching’

January 20, 2025

Version 0.9.7

Date 2024-01-11

Title Simulation and Estimation for Branching Processes

Author Camilo Jose Torres-Jimenez <`cjtorresj@unal.edu.co`>

Maintainer Camilo Jose Torres-Jimenez <`cjtorresj@unal.edu.co`>

Description Simulation and parameter estimation of multitype Bienayme - Galton - Watson processes.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-01-12 00:20:02 UTC

Contents

`BGWM.covar` *Variances and covariances of a multi-type Bienayme - Galton - Watson process*

Description

Calculates the covariance matrices of a multi-type Bienayme - Galton - Watson process from its offspring distributions, additionally, it could be obtained the covariance matrices in a specific time n and the covariance matrix of the population in the nth generation, if it is providesd the initial population vector.

Usage

```
BGWM.covar(dists, type=c("general","multinomial","independents"),
           d, n=1, z0=NULL, maxiter = 1e5)
```

Arguments

`dists`	offspring distributions. Its structure depends on the class of the Bienayme - Galton - Watson process (See details and examples).
`type`	Class or family of the Bienayme - Galton - Watson process (See details and examples).
`d`	positive integer, number of types.
`n`	positive integer, nth generation.
`z0`	nonnegative integer vector of size d; initial population by type.
`maxiter`	positive integer, size of the simulated sample used to estimate the parameters of univariate distributions that do not have an analytical formula for their exact calculation.

Details

This function calculates the covariance matrices of a multi-type Bienayme - Galton - Watson (BGWM) process from its offspring distributions.

From particular offspring distributions and taking into account a differentiated algorithmic approach, we propose the following classes or types for these processes:

`general` This option is for BGWM processes without conditions over the offspring distributions, in this case, it is required as input data for each distribution, all d-dimensional vectors with their respective, greater than zero, probability.

`multinomial` This option is for BGMW processes where each offspring distribution is a multinomial distribution with a random number of trials, in this case, it is required as input data, d univariate distributions related to the random number of trials for each multinomial distribution and a $d \times d$ matrix where each row contains probabilities of the d possible outcomes for each multinomial distribution.

`independents` This option is for BGMW processes where each offspring distribution is a joint distribution of d combined independent discrete random variables, one for each type of individuals, in this case, it is required as input data d^2 univariate distributions.

The structure need it for each classification is illustrated in the examples.

These are the univariate distributions available:

unif Discrete uniform distribution, parameters min and max. All the non-negative integers between min y max have the same probability.

binom Binomial distribution, parameters n and p.

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

for x = 0, . . . , n.

hyper Hypergeometric distribution, parameters m (the number of white balls in the urn), n (the number of white balls in the urn), k (the number of balls drawn from the urn).

$$p(x) = \binom{m}{x}\binom{n}{k-x} \bigg/ \binom{m+n}{k}$$

for x = 0, ..., k.

geom Geometric distribution, parameter p.

$$p(x) = p(1-p)^x$$

for x = 0, 1, 2, . . .

nbinom Negative binomial distribution, parameters n and p.

$$p(x) = \frac{\Gamma(x+n)}{\Gamma(n)x!} p^n (1-p)^x$$

for x = 0, 1, 2, . . .

pois Poisson distribution, parameter λ.

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

for x = 0, 1, 2, . . .

norm Normal distribution rounded to integer values and negative values become 0, parameters μ and σ.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$

for x = 1, 2, . . .

$$p(x) = \int_{-\infty}^{0.5} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$

for x = 0

lnorm Lognormal distribution rounded to integer values, parameters `logmean` $= \mu$ y `logsd` $= \sigma$.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sqrt{2\pi}\sigma t} e^{-(\log(t)-\mu)^2/2\sigma^2} dt$$

for x = 1, 2, . . .

$$p(x) = \int_{0}^{0.5} \frac{1}{\sqrt{2\pi}\sigma t} e^{-(\log(t)-\mu)^2/2\sigma^2} dt$$

for x = 0

gamma Gamma distribution rounded to integer values, parameters `shape` $= \alpha$ y `scale` $= \sigma$.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sigma^\alpha \Gamma(\alpha)} t^{\alpha-1} e^{-t/\sigma} dt$$

para x = 1, 2, . . .

$$p(x) = \int_{0}^{0.5} \frac{1}{\sigma^\alpha \Gamma(\alpha)} t^{\alpha-1} e^{-t/\sigma} dt$$

for x = 0

When the offspring distributions used `norm`, `lnorm` or `gamma`, mean and variance related to these univariate distributions is estimated by calculating sample mean and sample variance of `maxiter` random values generated from the corresponding distribution.

Value

A `matrix` object with the covariance matrices of the process in the nth generation, combined by rows, or, a `matrix` object with the covariace matrix of the population in the nth generation, in case of provide the initial population vector (z0).

Author(s)

Camilo Jose Torres-Jimenez <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. (2010), Relative frequencies and parameter estimation in multi-type Bienayme - Galton - Watson processes, Master's Thesis, Master of Science in Statistics. Universidad Nacional de Colombia. Bogota, Colombia.

Stefanescu, C. (1998), 'Simulation of a multitype Galton-Watson chain', Simulation Practice and Theory 6(7), 657-663.

Athreya, K. & Ney, P. (1972), Branching Processes, Springer-Verlag.

Harris, T. E. (1963), The Theory of Branching Processes, Courier Dover Publications.

See Also

`BGWM.mean`, `rBGWM`, `BGWM.mean.estim`, `BGWM.covar.estim`

Examples

```
## Not run:
## Variances and covariances of a BGWM process based on a model analyzed
## in Stefanescu (1998)

# Variables and parameters
d <- 2
n <- 30
N <- c(90, 10)
a <- c(0.2, 0.3)

# with independent distributions
Dists.i <- data.frame( name=rep( "pois", d*d ),
                       param1=rep( a, rep(d,d) ),
                       stringsAsFactors=FALSE )

# covariance matrices of the process
I.matriz.V <- BGWM.covar(Dists.i, "independents", d)

# covariance matrix of the population in the nth generation
# from vector N representing the initial population
I.matrix.V.n_N <- BGWM.covar(Dists.i, "independents", d, n, N)

# with multinomial distributions
dist <- data.frame( name=rep( "pois", d ),
                    param1=a*d,
```

```
                    stringsAsFactors=FALSE )
matrix.b <- matrix( rep(0.5, 4), nrow=2 )
Dists.m <- list( dists.eta=dist, matrix.B=matrix.b )

# covariance matrices of the process
M.matrix.V <- BGWM.covar(Dists.m, "multinomial", d)

# covariance matrix of the population in the nth generation
# from vector N representing the initial population
M.matrix.V.n_N <- BGWM.covar(Dists.m, "multinomial", d, n, N)

# with general distributions (approximation)
max <- 30
A <- t(expand.grid(c(0:max),c(0:max)))
aux1 <- factorial(A)
aux1 <- apply(aux1,2,prod)
aux2 <- apply(A,2,sum)
distp <- function(x,y,z){ exp(-d*x)*(x^y)/z }
p <- sapply( a, distp, aux2, aux1 )
prob <- list( dist1=p[,1], dist2=p[,2] )
size <- list( dist1=ncol(A), dist2=ncol(A) )
vect <- list( dist1=t(A), dist2=t(A) )
Dists.g <- list( sizes=size, probs=prob, vects=vect )

# covariance matrices of the process
G.matrix.V <- BGWM.covar(Dists.g, "general", d)

# covariance matrix of the population in the nth generation
# from vector N representing the initial population
G.matrix.V.n_N <- BGWM.covar(Dists.g, "general", d, n, N)

# Comparison of results
I.matrix.V.n_N
I.matrix.V.n_N - M.matrix.V.n_N
M.matrix.V.n_N - G.matrix.V.n_N
G.matrix.V.n_N - I.matrix.V.n_N

## End(Not run)
```

`BGWM.covar.estim` *Estimation of the covariance matrices of a multi-type Bienayme - Galton - Watson process*

Description

Calculates a estimation of the covariance matrices of a multi-type Bienayme - Galton - Watson process from experimental observed data that can be modeled by this kind of process.

Usage

```
BGWM.covar.estim(sample, method=c("EE-m","MLE-m"), d, n, z0)
```

Arguments

`sample`	nonnegative integer matrix with d columns and dn rows, trajectory of the process with the number of individuals for every combination parent type - descendent type (observed data).
`method`	methods of estimation (EE-m with empirical estimation of the mean matrix, MLE-m with maximum likelihood estimation of the mean matrix).
`d`	positive integer, number of types.
`n`	positive integer, nth generation.
`z0`	nonnegative integer vector of size d, initial population by type.

Details

This function estimates the covariance matrices of a BGWM process using two possible estimators from asymptotic results related with empirical estimator and maximum likelihood estimator of the mean matrix, they both require the so-called full sample associated with the process, ie, it is required to have the trajectory of the process with the number of individuals for every combination parent type - descendent type. For more details see Torres-Jimenez (2010) or Maaouia & Touati (2005).

Value

A `list` object with:

`method`	method of estimation selected.
`V`	A `matrix` object, estimation of the d covariance matrices of the process, combined by rows.

Author(s)

Camilo Jose Torres-Jimenez <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. (2010), Relative frequencies and parameter estimation in multi-type Bienayme - Galton - Watson processes, Master's Thesis, Master of Science in Statistics. Universidad Nacional de Colombia. Bogota, Colombia.

Maaouia, F. & Touati, A. (2005), 'Identification of Multitype Branching Processes', The Annals of Statistics 33(6), 2655-2694.

See Also

`BGWM.mean`, `BGWM.covar`, `BGWM.mean.estim`, `rBGWM`

Examples

```
## Not run:
## Estimation of covariace matrices from simulated data

# Variables and parameters
```

```
d <- 3
n <- 30
N <- c(10,10,10)
LeslieMatrix <- matrix( c(0.08, 1.06, 0.07,
                          0.99, 0, 0,
                          0, 0.98, 0), 3, 3 )

# offspring distributions from the Leslie matrix
# (with independent distributions)
Dists.pois <- data.frame( name=rep( "pois", d ),
                          param1=LeslieMatrix[,1],
                          param2=NA,
                          stringsAsFactors=FALSE )
Dists.binom <- data.frame( name=rep( "binom", 2*d ),
                           param1=rep( 1, 2*d ),
                           param2=c(t(LeslieMatrix[,-1])),
                           stringsAsFactors=FALSE )
Dists.i <- rbind(Dists.pois,Dists.binom)
Dists.i <- Dists.i[c(1,4,5,2,6,7,3,8,9),]
Dists.i

# covariance matrices of the process from its offspring distributions
V <- BGWM.covar(Dists.i,"independents",d)

# generated trajectories of the process from its offspring distributions
simulated.data <- rBGWM(Dists.i, "independents", d, n, N,
                        TRUE, FALSE, FALSE)$o.c.s

# estimation of covariance matrices using mean matrix empiric estimate
# from generated trajectories of the process
V.EE.m <- BGWM.covar.estim( simulated.data, "EE-m", d, n, N )$V

# estimation of covariance matrices using mean matrix maximum likelihood
# estimate from generated trajectories of the process
V.MLE.m <- BGWM.covar.estim( simulated.data, "MLE-m", d, n, N )$V

# Comparison of exact and estimated covariance matrices
V
V - V.EE.m
V - V.MLE.m

## End(Not run)
```

`BGWM.mean` *Means of a multi-type Bienayme - Galton - Watson process*

Description

Calculates the mean matrix of a multi-type Bienayme - Galton - Watson process from its offspring distributions, additionally, it could be obtained the mean matrix in a specific time n and the mean vector of the population in the nth generation, if it is provided the initial population vector.

Usage

```
BGWM.mean(dists, type=c("general","multinomial","independents"),
          d, n=1, z0=NULL, maxiter = 1e5)
```

Arguments

`dists`	offspring distributions. Its structure depends on the class of the Bienayme - Galton - Watson process (See details and examples).
`type`	Class or family of the Bienayme - Galton - Watson process (See details and examples).
`d`	positive integer, number of types.
`n`	positive integer, nth generation.
`z0`	nonnegative integer vector of size d, initial population by type.
`maxiter`	positive integer, size of the simulated sample used to estimate the parameters of univariate distributions that do not have an analytical formula for their exact calculation.

Details

This function calculates the mean matrix of a multi-type Bienayme - Galton - Watson (BGWM) process from its offspring distributions.

From particular offspring distributions and taking into account a differentiated algorithmic approach, we propose the following classes or types for these processes:

`general` This option is for BGWM processes without conditions over the offspring distributions, in this case, it is required as input data for each distribution, all d-dimensional vectors with their respective, greater than zero, probability.

`multinomial` This option is for BGMW processes where each offspring distribution is a multinomial distribution with a random number of trials, in this case, it is required as input data, d univariate distributions related to the random number of trials for each multinomial distribution and a $d \times d$ matrix where each row contains probabilities of the d possible outcomes for each multinomial distribution.

`independents` This option is for BGMW processes where each offspring distribution is a joint distribution of d combined independent discrete random variables, one for each type of individuals, in this case, it is required as input data d^2 univariate distributions.

The structure need it for each classification is illustrated in the examples.

These are the univariate distributions available:

unif Discrete uniform distribution, parameters min and max. All the non-negative integers between min y max have the same probability.

binom Binomial distribution, parameters n and p.

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

for x = 0, . . . , n.

hyper Hypergeometric distribution, parameters m (the number of white balls in the urn), n (the number of white balls in the urn), k (the number of balls drawn from the urn).

$$p(x) = \binom{m}{x}\binom{n}{k-x}\Big/\binom{m+n}{k}$$

for x = 0, ..., k.

geom Geometric distribution, parameter p.

$$p(x) = p(1-p)^x$$

for x = 0, 1, 2, . . .

nbinom Negative binomial distribution, parameters n and p.

$$p(x) = \frac{\Gamma(x+n)}{\Gamma(n)x!}p^n(1-p)^x$$

for x = 0, 1, 2, . . .

pois Poisson distribution, parameter λ.

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

for x = 0, 1, 2, . . .

norm Normal distribution rounded to integer values and negative values become 0, parameters μ and σ.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$

for x = 1, 2, . . .

$$p(x) = \int_{-\infty}^{0.5} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$

for x = 0

lnorm Lognormal distribution rounded to integer values, parameters `logmean` $= \mu$ y `logsd` $= \sigma$.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sqrt{2\pi}\sigma t} e^{-(\log(t)-\mu)^2/2\sigma^2} dt$$

for x = 1, 2, . . .

$$p(x) = \int_{0}^{0.5} \frac{1}{\sqrt{2\pi}\sigma t} e^{-(\log(t)-\mu)^2/2\sigma^2} dt$$

for x = 0

gamma Gamma distribution rounded to integer values, parameters `shape` $= \alpha$ y `scale` $= \sigma$.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sigma^\alpha \Gamma(\alpha)} t^{\alpha-1} e^{-t/\sigma} dt$$

para x = 1, 2, . . .

$$p(x) = \int_{0}^{0.5} \frac{1}{\sigma^\alpha \Gamma(\alpha)} t^{\alpha-1} e^{-t/\sigma} dt$$

for x = 0

When the offspring distributions used `norm`, `lnorm` or `gamma`, mean related to these univariate distributions is estimated by calculating sample mean of `maxiter` random values generated from the corresponding distribution.

Value

A `matrix` object with the mean matrix of the process in the nth generation, or, a `vector` object with the mean vector of the population in the nth generation, in case of provide the initial population vector (z0).

Author(s)

Camilo Jose Torres-Jimenez <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. (2010), Relative frequencies and parameter estimation in multi-type Bienayme - Galton - Watson processes, Master's Thesis, Master of Science in Statistics. Universidad Nacional de Colombia. Bogota, Colombia.

Stefanescu, C. (1998), 'Simulation of a multitype Galton-Watson chain', Simulation Practice and Theory 6(7), 657-663.

Athreya, K. & Ney, P. (1972), Branching Processes, Springer-Verlag.

Harris, T. E. (1963), The Theory of Branching Processes, Courier Dover Publications.

See Also

`rBGWM`, `BGWM.covar`, `BGWM.mean.estim`, `BGWM.covar.estim`

Examples

```
## Not run:
## Means of a BGWM process based on a model analyzed in Stefanescu (1998)

# Variables and parameters
d <- 2
n <- 30
N <- c(90, 10)
a <- c(0.2, 0.3)

# with independent distributions
Dists.i <- data.frame( name=rep( "pois", d*d ),
                       param1=rep( a, rep(d,d) ),
                       stringsAsFactors=FALSE )

# mean matrix of the process
I.matriz.m <- BGWM.mean(Dists.i, "independents", d)

# mean vector of the population in the nth generation
# from vector N representing the initial population
I.vector.m.n_N <- BGWM.mean(Dists.i, "independents", d, n, N)

# with multinomial distributions
dist <- data.frame( name=rep( "pois", d ),
                    param1=a*d,
                    stringsAsFactors=FALSE )
```

```
matrix.b <- matrix( rep(0.5, 4), nrow=2 )
Dists.m <- list( dists.eta=dist, matrix.B=matrix.b )

# mean matrix of the process
M.matrix.m <- BGWM.mean(Dists.m, "multinomial", d)

# mean vector of the population in the nth generation
# from vector N representing the initial population
M.vector.m.n_N <- BGWM.mean(Dists.m, "multinomial", d, n, N)

# with general distributions (approximation)
max <- 30
A <- t(expand.grid(c(0:max),c(0:max)))
aux1 <- factorial(A)
aux1 <- apply(aux1,2,prod)
aux2 <- apply(A,2,sum)
distp <- function(x,y,z){ exp(-d*x)*(x^y)/z }
p <- sapply( a, distp, aux2, aux1 )
prob <- list( dist1=p[,1], dist2=p[,2] )
size <- list( dist1=ncol(A), dist2=ncol(A) )
vect <- list( dist1=t(A), dist2=t(A) )
Dists.g <- list( sizes=size, probs=prob, vects=vect )

# mean matrix of the process
G.matrix.m <- BGWM.mean(Dists.g, "general", d)

# mean vector of the population in the nth generation
# from vector N representing the initial population
G.vector.m.n_N <- BGWM.mean(Dists.g, "general", d, n, N)

# Comparison of results
I.vector.m.n_N
I.vector.m.n_N - M.vector.m.n_N
M.vector.m.n_N - G.vector.m.n_N
G.vector.m.n_N - I.vector.m.n_N

## End(Not run)
```

`BGWM.mean.estim` *Estimation of the mean matrix of a multi-type Bienayme - Galton - Watson process*

Description

Calculates a estimation of the mean matrix of a multi-type Bienayme - Galton - Watson process from experimental observed data that can be modeled by this kind of process.

Usage

```
BGWM.mean.estim(sample, method=c("EE","MLE"), d, n, z0)
```

Arguments

`sample`	nonnegative integer matrix with d columns and dn rows, trajectory of the process with the number of individuals for every combination parent type - descendent type (observed data).
`method`	methods of estimation (EE Empirical estimacion, MLE Maximum likelihood estimation).
`d`	positive integer, number of types.
`n`	positive integer, nth generation.
`z0`	nonnegative integer vector of size d, initial population by type.

Details

This function estimates the mean matrix of a BGWM process using two possible estimators, empirical estimator and maximum likelihood estimator, they both require the so-called full sample associated with the process, ie, it is required to have the trajectory of the process with the number of individuals for every combination parent type - descendent type. For more details see Torres-Jimenez (2010) or Maaouia & Touati (2005).

Value

A `list` object with:

`method`	method of estimation selected.
`m`	A `matrix` object, estimation of the $d \times d$ mean matrix of the process.

Author(s)

Camilo Jose Torres-Jimenez <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. (2010), Relative frequencies and parameter estimation in multi-type Bienayme - Galton - Watson processes, Master's Thesis, Master of Science in Statistics. Universidad Nacional de Colombia. Bogota, Colombia.

Maaouia, F. & Touati, A. (2005), 'Identification of Multitype Branching Processes', The Annals of Statistics 33(6), 2655-2694.

See Also

`BGWM.mean`, `BGWM.covar`, `rBGWM`, `BGWM.covar.estim`

Examples

```
## Not run:
## Estimation of mean matrix from simulated data

# Variables and parameters
d <- 3
```

```
n <- 30
N <- c(10,10,10)
LeslieMatrix <- matrix( c(0.08, 1.06, 0.07,
                          0.99, 0, 0,
                          0, 0.98, 0), 3, 3 )

# offspring distributions from the Leslie matrix
# (with independent distributions)
Dists.pois <- data.frame( name=rep( "pois", d ),
                          param1=LeslieMatrix[,1],
                          param2=NA,
                          stringsAsFactors=FALSE )
Dists.binom <- data.frame( name=rep( "binom", 2*d ),
                           param1=rep( 1, 2*d ),
                           param2=c(t(LeslieMatrix[,-1])),
                           stringsAsFactors=FALSE )
Dists.i <- rbind(Dists.pois,Dists.binom)
Dists.i <- Dists.i[c(1,4,5,2,6,7,3,8,9),]
Dists.i

# mean matrix of the process from its offspring distributions
m <- BGWM.mean(Dists.i,"independents",d)

# generated trajectories of the process from its offspring distributions
simulated.data <- rBGWM(Dists.i, "independents", d, n, N,
                        TRUE, FALSE, FALSE)$o.c.s

# mean matrix empiric estimate from generated trajectories of the process
m.EE <- BGWM.mean.estim( simulated.data, "EE", d, n, N )$m

# mean matrix maximum likelihood estimate from generated trajectories
# of the process
m.MLE <- BGWM.mean.estim( simulated.data, "MLE", d, n, N )$m

# Comparison of exact and estimated mean matrices
m
m - m.EE
m - m.MLE

## End(Not run)
```

`rBGWM` *Simulating a multi-type Bienayme - Galton - Watson process*

Description

Generate the trajectories of a multi-type Bienayme - Galton - Watson process from its offspring distributions, using three different algorithms based on three different classes or families of these processes.

Usage

```
rBGWM(dists, type=c("general","multinomial","independents"), d,
      n, z0=rep(1,d), c.s=TRUE, tt.s=TRUE, rf.s=TRUE, file=NULL)
```

Arguments

`dists`	offspring distributions. Its structure depends on the class of the Bienayme - Galton - Watson process (See details and examples).
`type`	Class or family of the Bienayme - Galton - Watson process (See details).
`d`	positive integer, number of types.
`n`	positive integer, maximum lenght of the wanted trajectory.
`z0`	nonnegative integer vector of size d; initial population by type.
`c.s`	logical value, if TRUE, the output object will include the generated trajectory of the process with the number of individuals for every combination parent type - descendent type.
`tt.s`	logical value, if TRUE, the output object will include the generated trajectory of the process with the number of descendents by type.
`rf.s`	logical value, if TRUE, the output object will include the generated trajectory of the process with the relative frequencies by type.
`file`	the name of the output file where the generated trajectory of the process with the number of individuals for every combination parent type - descendent type could be stored.

Details

This function performs a simulation of a multi-type Bienayme - Galton - Watson process (BGWM) from its offspring distributions.

From particular offspring distributions and taking into account a differentiated algorithmic approach, we propose the following classes or types for these processes:

`general` This option is for BGWM processes without conditions over the offspring distributions, in this case, it is required as input data for each distribution, all d-dimensional vectors with their respective, greater than zero, probability.

`multinomial` This option is for BGMW processes where each offspring distribution is a multinomial distribution with a random number of trials, in this case, it is required as input data, d univariate distributions related to the random number of trials for each multinomial distribution and a $d \times d$ matrix where each row contains probabilities of the d possible outcomes for each multinomial distribution.

`independents` This option is for BGMW processes where each offspring distribution is a joint distribution of d combined independent discrete random variables, one for each type of individuals, in this case, it is required as input data d^2 univariate distributions.

The structure need it for each classification is illustrated in the examples.

These are the univariate distributions available:

unif Discrete uniform distribution, parameters min and max. All the non-negative integers between min y max have the same probability.

binom Binomial distribution, parameters n and p.

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}$$

for x = 0, . . . , n.

hyper Hypergeometric distribution, parameters m (the number of white balls in the urn), n (the number of white balls in the urn), k (the number of balls drawn from the urn).

$$p(x) = \binom{m}{x} \binom{n}{k-x} \Big/ \binom{m+n}{k}$$

for x = 0, ..., k.

geom Geometric distribution, parameter p.

$$p(x) = p(1-p)^x$$

for x = 0, 1, 2, . . .

nbinom Negative binomial distribution, parameters n and p.

$$p(x) = \frac{\Gamma(x+n)}{\Gamma(n)x!} p^n (1-p)^x$$

for x = 0, 1, 2, . . .

pois Poisson distribution, parameter λ.

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

for x = 0, 1, 2, . . .

norm Normal distribution rounded to integer values and negative values become 0, parameters μ and σ.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$

for x = 1, 2, . . .

$$p(x) = \int_{-\infty}^{0.5} \frac{1}{\sqrt{2\pi}\sigma} e^{-(t-\mu)^2/2\sigma^2} dt$$

for x = 0

lnorm Lognormal distribution rounded to integer values, parameters `logmean` $= \mu$ y `logsd` $= \sigma$.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sqrt{2\pi}\sigma t} e^{-(\log(t)-\mu)^2/2\sigma^2} dt$$

for x = 1, 2, . . .

$$p(x) = \int_{0}^{0.5} \frac{1}{\sqrt{2\pi}\sigma t} e^{-(\log(t)-\mu)^2/2\sigma^2} dt$$

for x = 0

gamma Gamma distribution rounded to integer values, parameters `shape` $= \alpha$ y `scale` $= \sigma$.

$$p(x) = \int_{x-0.5}^{x+0.5} \frac{1}{\sigma^{\alpha}\Gamma(\alpha)} t^{\alpha-1} e^{-t/\sigma} dt$$

para x = 1, 2, . . .

$$p(x) = \int_{0}^{0.5} \frac{1}{\sigma^{\alpha}\Gamma(\alpha)} t^{\alpha-1} e^{-t/\sigma} dt$$

for x = 0

Value

An object of class `list` with these components:

`i.d`	input. number of types.
`i.dists`	input. offspring distributions.
`i.n`	input. maximum lenght of the generated trajectory.
`i.z0`	input. initial population by type.
`o.c.s`	output. A matrix indicating the generated trajectory of the process with the number of individuals for every combination parent type - descendent type.
`o.tt.s`	output. A matrix indicating the generated trajectory of the process with the number of descendents by type.
`o.rf.s`	output. A matrix indicating the generated trajectory of the process with the relative frequencies by type.

Author(s)

Camilo Jose Torres-Jimenez `<cjtorresj@unal.edu.co>`

References

Torres-Jimenez, C. J. (2010), Relative frequencies and parameter estimation in multi-type Bienayme - Galton - Watson processes, Master's Thesis, Master of Science in Statistics. Universidad Nacional de Colombia. Bogota, Colombia.

Stefanescu, C. (1998), 'Simulation of a multitype Galton-Watson chain', Simulation Practice and Theory 6(7), 657-663.

Athreya, K. & Ney, P. (1972), Branching Processes, Springer-Verlag.

See Also

`BGWM.mean`, `BGWM.covar`, `BGWM.mean.estim`, `BGWM.covar.estim`

Examples

```
## Not run:
## Simulation based on a model analyzed in Stefanescu(1998)

# Variables and parameters
d <- 2
n <- 30
N <- c(90, 10)
a <- c(0.2, 0.3)

# with independent distributions
Dists.i <- data.frame( name=rep( "pois", d*d ),
                       param1=rep( a, rep(d,d) ),
                       stringsAsFactors=FALSE )

rA <- rBGWM(Dists.i, "independents", d, n, N)

# with multinomial distributions
dist <- data.frame( name=rep( "pois", d ),
                    param1=a*d,
                    stringsAsFactors=FALSE )
matrix.b <- matrix( rep(0.5, 4), nrow=2 )
Dists.m <- list( dists.eta=dist, matrix.B=matrix.b )

rB <- rBGWM(Dists.m, "multinomial", d, n, N)

# with general distributions (approximation)
max <- 30
A <- t(expand.grid(c(0:max),c(0:max)))
aux1 <- factorial(A)
aux1 <- apply(aux1,2,prod)
aux2 <- apply(A,2,sum)
distp <- function(x,y,z){ exp(-d*x)*(x^y)/z }
p <- sapply( a, distp, aux2, aux1 )
prob <- list( dist1=p[,1], dist2=p[,2] )
size <- list( dist1=ncol(A), dist2=ncol(A) )
vect <- list( dist1=t(A), dist2=t(A) )
Dists.g <- list( sizes=size, probs=prob, vects=vect )

rC <- rBGWM(Dists.g, "general", d, n, N)

# Comparison chart
dev.new()
plot.ts(rA$o.tt.s,main="with independents")
dev.new()
plot.ts(rB$o.tt.s,main="with multinomial")
dev.new()
plot.ts(rC$o.tt.s,main="with general (aprox.)")

## End(Not run)
```

Index