Package ‘branchCorr’

October 18, 2016

Type Package

Title Correlation Matching for Branching Processes

Maintainer Jason Xu <jqxu@ucla.edu>

Author Jason Xu

Description Z-estimation via correlation matching for partially observed branching processes and stochastic compartmental models.

Version 0.1.0

License What license is it under?

LazyData TRUE

Depends R (>= 3.0.1), BiasedUrn

Suggests plyr

Collate branchCorr_source.r

RoxygenNote 5.0.1

R topics documented:

`Cor_xyFrom1` *Model-based correlation from HSC*

Description

This function computes the model-based correlation between two mature cell types given that the process begins with one HSC. The function is basically a wrapper around those used to compute variance and covariances.

Usage

```
Cor_xyFrom1(t, rates, progType1, progType2, type1, type2, progStructure)
```

Arguments

`t`	The length of time
`rates`	A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`progType1`	An index indicating the progenitor type that mature cell 1 can descend from
`progType2`	An index indicating the progenitor type that mature cell 2 can descend from
`type1`	An index indicating the type of mature cell 1
`type2`	An index indicating the type of mature cell 2
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Correlation between specified mature type compartments at time t

`Cor_xyFrom2` *Model-based correlation from progenitor*

Description

This function computes the model-based correlation between two mature cell types given that the process begins with one progenitor. The function is basically a wrapper around those used to compute variance and covariances.

Usage

```
Cor_xyFrom2(t, rates, progType1, progType2, type1, type2, progStructure)
```

Arguments

`t`	The length of time
`rates`	A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`progType1`	An index indicating the progenitor type that mature cell 1 can descend from
`progType2`	An index indicating the progenitor type that mature cell 2 can descend from
`type1`	An index indicating the type of mature cell 1
`type2`	An index indicating the type of mature cell 2
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Correlation between specified mature type compartments at time t

`Cov_xyFrom1` *Model-based covariance from HSC*

Description

This function computes the model-based covariance between two mature cell types given that the process begins with one HSC. The function simply converts the raw moments U, M into the corresponding covariance expression.

Usage

```
Cov_xyFrom1(t, rates, progType1, progType2, type1, type2, progStructure)
```

Arguments

`t` The length of time

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progType1` An index indicating the progenitor type that mature cell 1 can descend from

`progType2` An index indicating the progenitor type that mature cell 2 can descend from

`type1` An index indicating the type of mature cell 1

`type2` An index indicating the type of mature cell 2

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Covariance between specified mature type compartments at time t

`Cov_xyFrom2` *Model-based covariance from progenitor*

Description

This function computes the model-based covariance between two mature cell types given that the process begins with one progenitor. The function simply converts the raw moments U, M into the corresponding covariance expression.

Usage

```
Cov_xyFrom2(t, rates, progType1, progType2, type1, type2, progStructure)
```

Arguments

`t` The length of time

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progType1` An index indicating the progenitor type that mature cell 1 can descend from

`progType2` An index indicating the progenitor type that mature cell 2 can descend from

`type1` An index indicating the type of mature cell 1

`type2` An index indicating the type of mature cell 2

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Covariance between specified mature type compartments at time t

expit *Expit function*

Description

Evaluates the expit function

Usage

```
expit(x)
```

Arguments

`x`	A number or vector

Value

The value of expit(x)

Examples

```
x = .25; expit(x)
x = 1:5; expit(x)
```

getObsCorr *Compute observed pairwise correlations*

Description

Returns the empirical pairwise correlations between each pair of types, given a dataset of observed counts and a vector containing the corresponding observation times. The data matrix should be in the same format as returned by `simCompObserved` with vecOutput=TRUE

Usage

```
getObsCorr(data, obsTimes)
```

Arguments

`data`	A matrix of mature cell type counts, in format produced by `simCompObserved` with vecOutput=TRUE
`obsTimes`	A vector of corresponding observation times

Value

Matrix containing pairwise correlations at each observation time

hyperGeoSample *Hypergeometric sampling*

Description

Samples hypergeometrically from a matrix containing discretely observed data from a stochastic compartmental model.

Usage

```
hyperGeoSample(data, sampSize, constantPopulation = FALSE, N = 10000)
```

Arguments

`data`	A matrix containing discretely observed data, in the format returned by `simCompObserved`
`sampSize`	The sample size or number of draws n in a hypergeometric distribution
`constantPopulation`	Logical whether to sample from equal total population, N, for all compartments
`N`	The fixed total population if constantPopulation=TRUE

inferNLMINBSamplingCorr *Optimize correlation loss function*

Description

This function uses the generic optimization package `nlminb` to optimize the objective function given by `samplingCorrObjective`. Takes additional initial guess parameter and allows specification of true death rates, which are often known and can be fixed throughout optimization.

Usage

```
inferNLMINBSamplingCorr(initGuess, obsTimes, obsCorr, nSample, nTotal,
  trueDeaths, progStructure, max = 5000)
```

Arguments

`initGuess`	Vector containing initial guess for par
`obsTimes`	A vector of observation times
`obsCorr`	Matrix of empirical correlations returned by `getObsCorr`
`nSample`	A vector containing the sample sizes of each mature type as arguments to hypergeometric sampling; this is the number of draws n
`nTotal`	A vector containing the total sizes of each mature population; this is the total population N in hypergeometric sampling
`trueDeaths`	Vector containing the true death rates to be fixed
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends
`max`	Max iterations, default to 5000

Value

An `nlminb` object containing optimal parameters

`marginalizedCor_xy` *Marginalized model-based correlations*

Description

Computes the marginalized correlation between two mature cell types, given the initial probability distribution.

Usage

```
marginalizedCor_xy(t, rates, initProb, type1, type2, progStructure)
```

Arguments

`t`	The time length
`rates`	A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`initProb`	vector containing initial probabilities of beginning in HSC comparmtent and each progenitor compartment
`type1`	Index of mature type 1
`type2`	Index of mature type 2
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

The model-based marginalized correlation between types at time t

---

`M_1x` *Mean population starting from HSC*

Description

This function computes the model-based mean of a mature cell type compartment, given rates and that the process begins with one initial hematopoietic stem cell (compartment 1).

Usage

```
M_1x(t, rates, progType, type, progStructure)
```

Arguments

- `t` The length of time
- `rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
- `progType` An index indicating the progenitor type that the mature cell type can descend from
- `type` An index indicating the mature cell type
- `progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

The mean population of compartment ‘type’ after time t

M_2x *This function computes the model-based mean of a mature cell type compartment, given rates and that the process begins with one initial progenitor.*

Description

This function computes the model-based mean of a mature cell type compartment, given rates and that the process begins with one initial progenitor.

Usage

```
M_2x(t, rates, progType, type, progStructure)
```

Arguments

- `t` The length of time
- `rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
- `progType` An index indicating the initial progenitor type
- `type` An index indicating the mature cell type
- `progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

The mean population of compartment ‘type’ after time t

`NLMINBtrycatch` *Infer optimal rates via correlation loss function with error handling*

Description

This function is a wrapper for `optimizeSamplingCorrNLMINB` that checks for any error codes using `tryCatch`

Usage

```
NLMINBtrycatch(data, numInits, initMean, obsTimes, nSample, nTotal, trueDeaths,
  progStructure)
```

Arguments

`data`	A matrix of observed counts in the same format as produced by `simCompObserved`
`numInits`	Number of random restarts
`initMean`	The mean for the random initial parameters
`obsTimes`	A vector of observation times
`nSample`	A vector containing the sample sizes of each mature type as arguments to hypergeometric sampling; this is the number of draws n
`nTotal`	A vector containing the total sizes of each mature population; this is the total population N in hypergeometric sampling
`trueDeaths`	Vector containing the true death rates to be fixed
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

`optimizeSamplingCorrNLMINB` *Infer optimal rates via correlation loss function*

Description

This function is a wrapper for `inferNLMINBSamplingCorr` that takes a dataset in the format produced by `simCompObserved` and infers its most likely parameters. Computes the empirical correlations and optimizes the correlation loss function over many random initializations, and returns the best estimate in terms of lowest objective function value.

Usage

```
optimizeSamplingCorrNLMINB(data, numInits, initMean, obsTimes, nSample, nTotal,
  trueDeaths, progStructure)
```

Arguments

`data` A matrix of observed counts in the same format as produced by `simCompObserved`

`numInits` Number of random restarts

`initMean` The mean for the random initial parameters

`obsTimes` A vector of observation times

`nSample` A vector containing the sample sizes of each mature type as arguments to hypergeometric sampling; this is the number of draws n

`nTotal` A vector containing the total sizes of each mature population; this is the total population N in hypergeometric sampling

`trueDeaths` Vector containing the true death rates to be fixed

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Vector containing the best fitting parameters

`randInit` *Generate random initial population vector*

Description

This function converts an initial distribution to an initial population indicator vector. It samples from the initial distribution, and returns an initial population vector with an indicator of 1 in the compartment from which the process will begin, with zeros elsewhere.

Usage

```
randInit(initProbs, totalTypes)
```

Arguments

`initProbs` The initial distribution vector

`totalTypes` The total number of compartments or types

Value

A vector of zeros in all but one entry, indicating the initial type

samplingCorrObjective *Correlation loss function objective*

Description

Computes the objective function of the correlation-matching loss function estimator. Takes in model parameters and a matrix of empirical correlations in the format produced by `getObsCorr`. This can then be plugged into a generic optimization routine such as `nlminb`. Natural constraints such as positivity of parameters are enforced via a log-barrier penalty.

Usage

```
samplingCorrObjective(par, obsTimes, obsCorr, nSample, nTotal, progStructure)
```

Arguments

`par`	A vector containing the process rates followed by all but the first component of the initial distribution vector. The rates should be ordered as before, beginning with self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`obsTimes`	A vector of observation times
`obsCorr`	Matrix of empirical correlations returned by `getObsCorr`
`nSample`	A vector containing the sample sizes of each mature type as arguments to hypergeometric sampling; this is the number of draws n
`nTotal`	A vector containing the total sizes of each mature population; this is the total population N in hypergeometric sampling
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Objective value

samplingCor_xy *Marginalized model-based correlations with sampling*

Description

Computes the marginalized correlation between two mature cell types, given the initial probability distribution, incorporating the effect of hypergeometric sampling.

Usage

```
samplingCor_xy(t, rates, initProb, type1, type2, pop1, pop2, total1, total2,
  progStructure)
```

Arguments

`t` The time length

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`initProb` vector containing initial probabilities of beginning in HSC comparmtent and each progenitor compartment

`type1` Index of mature type 1

`type2` Index of mature type 2

`pop1` Sample size of mature type 1

`pop2` Sample size of mature type 2

`total1` Total population of mature type 1 cells

`total2` Total population of mature type 2 cells

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Correlation between types at time t after marginalizing over sampling distribution and initial distribution

sim.once *Simulate discretely observed data from a stochastic compartmental model with error check*

Description

Wrapper for `simCompObserved` that checks for no error code

Usage

```
sim.once(t.end, initPopulation, rates, progStructure, obsTimes,
  vecOutput = FALSE, maxEvents = 999999999999)
```

Arguments

`t.end` The total time until end of simulation

`initPopulation` A vector of length equal to total number of compartments/types; entries contain the initial population of each type

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

`obsTimes` A vector containing the observation times

`vecOutput` Logical, whether to return a vectorized representation of the matrix of observation times by population counts

`maxEvents` The maximum number of events to simulate

`simCompartments` *Forward simulate from a stochastic compartmental model*

Description

Uses Gillespie forward simulation from a stochastic compartmental model, i.e. representing a hematopoietic tree.

Usage

```
simCompartments(t.end, initPopulation, rates, progStructure,
  maxEvents = 999999999999)
```

Arguments

`t.end` The total time until end of simulation

`initPopulation` A vector of length equal to total number of compartments/types; entries contain the initial population of each type

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

`maxEvents` The maximum number of events to simulate

Value

A vector containing the mature type populations at end of simulation period (time t=t.end).

Examples

```
progStructure <- c(1,1,1,2,2)
initPopulation <- c(10,0,0,0,0,0,0,0)
rates <- c(.3,.2,.5,.06, .03, 2, 4, 5, 3, 1, .15,.5,.8,.1,.05)
simCompartments(5, initPopulation, rates,progStructure)
```

simCompObserved *Simulate discretely observed data from a stochastic compartmental model*

Description

This code simulates data analogously to `simCompartments`, but records populations at a specified list of observation times.

Usage

```
simCompObserved(t.end, initPopulation, rates, progStructure, obsTimes,
  vecOutput = FALSE, maxEvents = 999999999999)
```

Arguments

`t.end` The total time until end of simulation

`initPopulation` A vector of length equal to total number of compartments/types; entries contain the initial population of each type

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progStructure` A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

`obsTimes` A vector containing the observation times

`vecOutput` Logical, whether to return a vectorized representation of the matrix of observation times by population counts

`maxEvents` The maximum number of events to simulate

Value

A matrix of population sizes of each compartment at each observaiton time. Rows index cell type; columns index observation times. Returns the vectorized form of thie matrix if vecOutput=TRUE.

Examples

```
progStructure <- c(1,1,1,2,2)
initPopulation <- c(10,0,0,0,0,0,0,0)
obsTimes <- 1:5
rates <- c(.3,.2,.5,.06, .03, 2, 4, 5, 3, 1, .15,.5,.8,.1,.05)
simCompObserved(5, initPopulation, rates,progStructure,obsTimes)
```

U_xx *Second moments starting from HSC*

Description

This function computes the model-based second moments, denoted U_mm|0 in the manuscript, of a mature cell type, given rates and that the process begins with one initial hematopoietic stem cell (compartment 1).

Usage

```
U_xx(t, rates, progType, type, progStructure)
```

Arguments

`t`	The length of time
`rates`	A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`progType`	An index indicating the progenitor type that the mature cell type can descend from
`type`	An index indicating the mature cell type
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Value of second moment at time t

U_xy *Second cross-moments starting from HSC*

Description

This function computes the model-based second cross-moments, denoted U_mnl0 in the manuscript, between two mature cell types, given rates and that the process begins with one initial hematopoietic stem cell (compartment 1).

Usage

```
U_xy(t, rates, progType1, progType2, type1, type2, progStructure)
```

Arguments

`t`	The length of time
`rates`	A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`progType1`	An index indicating the progenitor type that mature cell 1 can descend from
`progType2`	An index indicating the progenitor type that mature cell 2 can descend from
`type1`	An index indicating the type of mature cell 1
`type2`	An index indicating the type of mature cell 2
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Value of second cross-moment at time t

Var_xFrom1 *Model-based variance from HSC*

Description

This function computes the model-based variance of a mature cell type given that the process begins with one HSC. The function simply converts the raw moments U, M into the corresponding variance expression.

Usage

```
Var_xFrom1(t, rates, progType, type, progStructure)
```

Arguments

`t` The length of time

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progType` An index indicating the progenitor type that the mature cell type can descend from

`type` An index indicating the mature cell type

`progStructure`
A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Variance of specified mature cell compartment at time t

`Var_xFrom2` *Model-based variance from progenitor*

Description

This function computes the model-based variance of a mature cell type given that the process begins with one progenitor. The function simply converts the raw moments U, M into the corresponding variance expression.

Usage

```
Var_xFrom2(t, rates, progType, type, progStructure)
```

Arguments

`t` The length of time

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progType` An index indicating the initial progenitor type

`type` An index indicating the mature cell type

`progStructure`
A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Variance of specified mature cell compartment at time t

V_xx — *Second moments starting from progenitor*

Description

This function computes the model-based second moments, denoted U_mm|a in the manuscript, of a mature cell type, given rates and that the process begins with one initial progenitor cell.

Usage

```
V_xx(t, rates, progType, type, progStructure)
```

Arguments

`t`	The length of time
`rates`	A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)
`progType`	An index indicating the initial progenitor type
`type`	An index indicating the mature cell type
`progStructure`	A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Value of second moment at time t

V_xy — *Second cross-moments starting from progenitor*

Description

This function computes the model-based second cross-moments, denoted U_mn|a in the manuscript, between two mature cell types, given rates and that the process begins with one progenitor cell.

Usage

```
V_xy(t, rates, progType1, progType2, type1, type2, progStructure)
```

Arguments

`t` The length of time

`rates` A rate vector containing self-renewal rates of first compartment, followed by differentiation rates of intermediate types, death rates of intermediate types, differentiation rates of mature types, and death rates of mature types, i.e. (renewHSC, diffProgA, diffProgB, ..., deathProgA, deathProgB, ... diffMature1, diffMature2, ..., deathMature1, deathMature2, ...)

`progType1` An index indicating the progenitor type that mature cell 1 can descend from

`progType2` An index indicating the progenitor type that mature cell 2 can descend from

`type1` An index indicating the type of mature cell 1

`type2` An index indicating the type of mature cell 2

`progStructure`
A vector of length equal to number of mature types whose i'th entry contains the corresponding hidden progenitor type from which mature type i descends

Value

Value of second cross-moment at time t