

Package ‘rcolgem’

July 26, 2014

Type Package

Title Compute the likelihood of a gene genealogy conditional on a complex demographic history.

Version 0.0.2

Date 2014-07-23

Author {Erik M Volz, Oliver Ratmann}

Maintainer Erik M Volz <erik.volz+rcolgem@gmail.com>

Depends R (>= 2.15.1), ape, deSolve

Description This package provides methods to compute the likelihood of a gene genealogy using coalescent models for populations with complex demographic histories. A common example of a complex demographic history is the Susceptible-Infected-Recovered model of infectious disease dynamics and variations thereof. These methods may be used to obtain MLEs of compartmental models.

License GPL-3

Collate 'rcolgem.R'

R topics documented:

rcolgem-package	2
binaryDatedTree	2
calculate.cluster.size.moments.from.model	3
calculate.cluster.size.moments.from.tree	4
coalescent.log.likelihood	4
simulatedBinaryDatedTree	6

Index	7
--------------	----------

rcolgem-package	<i>Compute the likelihood of a gene genealogy and simulate genealogies given complex models of a demographic history, such as compartmental models of infectious disease epidemics.</i>
-----------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Description

This package provides methods to compute the likelihood of a gene genealogy using structured coalescent models for populations with complex demographic histories. An example of a complex demographic history is the Susceptible-Infected-Recovered model of infectious disease dynamics and variations thereof.

Details

Package:	rcolgem
Type:	Package
Version:	0.0.1
Date:	2014-07-22
License:	GPL-3

Author(s)

Erik M Volz

Maintainer: Erik M Volz <erik.volz+rcolgem@gmail.com>

References

E. M. Volz, Complex population dynamics and the coalescent under neutrality, Genetics, January, 2012

Examples

Refer to vignettes for example usage

binaryDatedTree	<i>Create binary dated tree binaryDatedTree class, includes heights for each node and other helper variables</i>
-----------------	------------------------------------------------------------------------------------------------------------------

Description

Create binary dated tree `binaryDatedTree` class, includes heights for each node and other helper variables like `maxSampleTime`.

Usage

```
binaryDatedTree(phylo, sampleTimes, sampleStates=NULL, sampleStatesAnnotations=NULL)
```

Arguments

<code>phylo</code>	<code>ape::phylo tree</code>
<code>sampleTimes</code>	Named vector of sample times for each taxon. Names must correspond to <code>phylo\$tip.label</code>
<code>sampleStates</code>	Matrix of discrete character information for each taxon. If <code>NULL</code> , this may be inferred from taxon label. Rownames must correspond to <code>phylo\$tip.label</code> . Colnames must correspond to names of demes used in subsequent analysis. Each row is a probability vector summing to 1 (the probability that taxon occupies each deme).
<code>sampleStatesAnnotations</code>	Vector of possible discrete character states for taxa. If inferring taxon state from label, this provides the possible matches for taxon annotations. The annotations are extracted from all text following ' _ ' at the end of the taxon label.

Value

A `binaryDatedTree` object. Includes numeric vector heights for each tip and internal node, the time before most recent sample when each node occurs.

```
calculate.cluster.size.moments.from.model
```

Calculate cluster size moments from model

Description

Calculate cluster size moments from model

Usage

```
calculate.cluster.size.moments.from.model(sampleTime,
  sampleStates, maxTime = NA, minTime = NA,
  timeResolution = 50, discretizeRates = FALSE,
  fgyResolution = 100, integrationMethod = "adams")
```

Arguments

sampleTime
 sampleStates
 maxTime
 minTime
 timeResolution
 discretizeRates

 fgyResolution
 integrationMethod

calculate.cluster.size.moments.from.tree

Calculate cluster size moments from tree

Description

Calculate cluster size moments from tree

Usage

calculate.cluster.size.moments.from.tree(bdt, heights)

Arguments

bdt binary dated tree
 heights vector numeric, heights at which to calculate cluster sizes

coalescent.log.likelihood

Compute the likelihood of a gene genealogy conditonal on a demographic history.

Description

This function

Usage

coalescent.log.likelihood(bdt, births, deaths, nonDemeDynamics, t0, x0, migrations=NA, parms=NA, fg

Arguments

bdt	A binaryDatedTree object which extends ape:phylo and includes dates of tips and internal nodes.
births	A vector or matrix of strings. These are evaluated as equations for the number of births within and between demes. Must have rownames and colnames corresponding to the names of demes.
deaths	A vector of strings. These are evaluated as equations for the rate that lineages in each deme are terminated. Must have rownames corresponding to the names of demes.
nonDemeDynamics	A vector of strings. These are evaluated as equations for the rate of change of state variables that do not correspond to demes. Vector must have names of state variables.
t0	The time of origin of the process. Should be before root of genealogy.
x0	A vector of initial conditions for the demographic process (the state of the system at time t0). Should include the name and value of all variables mentioned in births and nonDemeDynamics.
migrations	A vector or matrix of strings. These are evaluated as equations for the number of migrations between demes. Must have rownames and colnames corresponding to the names of demes. Should be omitted if there is only one deme.
parms	A list of parameters that will be accessible to differential equations specified in births, migrations and nonDemeDynamics.
fgyResolution	Determines timestep of ODEs (larger values gives smaller time step)
integrationMethod	Passed to lsoda. Higher precision may be obtained with methods such as <i>adams</i> at some computational cost.
sensorAtHeight	Boolean or numeric. If numeric, will ignore all parts of the tree older than this value when calculating the likelihood.
forgiveAgtY	If this is zero, function will return -Inf if at any point the number of lineages exceeds population size. If this value is between zero and one, function will return -Inf if the number of lineages exceeds population size AND number of lineages is less than this value times sample size. This can be useful when fitting models where the number of lineages is close to population size, which often happens during early period of exponential growth.

Details

See vignettes for more detailed usage.

Value

The log likelihood (numeric).

Author(s)

Erik M Volz

References

E. M. Volz, Complex population dynamics and the coalescent under neutrality, Genetics, January, 2012

See Also

[binaryDatedTree](#)

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as...
```

simulatedBinaryDatedTree

Simulate a binary dated tree using a coalescent model given a complex demographic process

Usage

```
simulatedBinaryDatedTree(sampleTime, sampleStates, FGY=NULL, discretizeRates=FALSE, fgyResolution =
```

Arguments

sampleTime	Time when sample takes place.
sampleStates	Numeric matrix of sample states for each taxon. Each row is a probability vector, sums to one, describes probability that each taxon is in each deme.
FGY	List with elements F, G, Y, each is a function of time. F(t) and G(t) return a matrix that respectively gives the total rate of births and migrations between demes at time t. Y(t) returns a vector of the population size in each deme at time t. If NULL, will assume F, G, and Y are global variables.
discretizeRates	If TRUE, a fast discrete approximation will be used for rates of births and migrations.
fgyResolution	If discretizeRates is TRUE, this variable describes number of time steps in discrete approximation. Larger values give better approximation at greater computational cost.

Value

A `binaryDatedTree`, extends `ape::phylo` and includes heights of all nodes, time before most recent sample when node occurs.

Index

*Topic **coalescent**

- `coalescent.log.likelihood`, [4](#)
- `rcolgem-package`, [2](#)

`binaryDatedTree`, [2](#), [6](#)

`calculate.cluster.size.moments.from.model`,
[3](#)

`calculate.cluster.size.moments.from.tree`,
[4](#)

`coalescent.log.likelihood`, [4](#)

`rcolgem(rcolgem-package)`, [2](#)

`rcolgem-package`, [2](#)

`simulatedBinaryDatedTree`, [6](#)