Package ‘TransPhylo’

April 20, 2021

Version 1.4.5

Date 2021-04-20

Title Inference of Transmission Tree from a Dated Phylogeny

Maintainer Xavier Didelot `<xavier.didelot@gmail.com>`

Description Inference of transmission tree from a dated phylogeny.
Includes methods to simulate and analyse outbreaks.
The methodology is described in
Didelot et al. (2014) <doi:10.1093/molbev/msu121>,
Didelot et al. (2017) <doi:10.1093/molbev/msw275>.

License GPL (>= 2)

Depends R (>= 3.0.0)

Imports Rcpp (>= 0.12.8), stats, graphics, ape

Suggests knitr, testthat, purrr, coda, grDevices, lattice, rmarkdown

LinkingTo Rcpp

Encoding UTF-8

RoxygenNote 7.1.1

SystemRequirements C++11

VignetteBuilder knitr

NeedsCompilation yes

Author Xavier Didelot [aut, cre] (<https://orcid.org/0000-0003-1885-500X>),
Yuanwei Xu [ctb]

Repository CRAN

Date/Publication 2021-04-20 13:30:03 UTC

R topics documented:

`TransPhylo-package` *Inference of Transmission Tree from a Dated Phylogeny*

Description

Inference of transmission tree from a dated phylogeny. Includes methods to simulate and analyse outbreaks.

Author(s)

Xavier Didelot <`xavier.didelot@gmail.com`>

References

Didelot et al. (2014) <doi:10.1093/molbev/msu121> Didelot et al. (2017) <doi:10.1093/molbev/msw275>.

See Also

https://github.com/xavierdidelot/TransPhylo

`as.mcmc.resTransPhylo` *Convert to coda mcmc format*

Description

Convert to coda mcmc format

Usage

```
as.mcmc.resTransPhylo(x, burnin = 0.5)
```

Arguments

`x`	Output from inferTTree
`burnin`	Proportion of the MCMC output to be discarded as burnin

Value

mcmc object from coda package

`computeMatTDist` *Build a matrix indicating for each pairs of individuals how many intermediates there are in the transmission chain*

Description

Build a matrix indicating for each pairs of individuals how many intermediates there are in the transmission chain

Usage

```
computeMatTDist(record, burnin = 0.5)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin

Value

Matrix of intermediates in transmission chains between pairs of hosts

---

`computeMatWIW` *Build a matrix of probability of who infected whom from a MCMC output*

Description

Build a matrix of probability of who infected whom from a MCMC output

Usage

```
computeMatWIW(record, burnin = 0.5)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin

Value

Matrix of probability of who infected whom

`consTTree` *Build a consensus transmission tree from a MCMC output*

Description

Build a consensus transmission tree from a MCMC output

Usage

```
consTTree(record, burnin = 0.5, minimum = 0.2, debug = F)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin
`minimum`	Minimum probability for inclusion of a partition in the consensus
`debug`	Used for debugging

Value

The consensus transmission tree

`convertToCoda` *Convert to coda mcmc format*

Description

Convert to coda mcmc format

Usage

```
convertToCoda(record, burnin = 0.5)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin

Value

Object of class mcmc from coda package

`dateLastSample` *Return the date of last sample from a ttree or ctree or ptree*

Description

Return the date of last sample from a ttree or ctree or ptree

Usage

```
dateLastSample(x)
```

Arguments

`x`	A transmission tree or colored tree or phylogenetic tree

Value

date of the last sample

`extractCTree` *Return the combined tree corresponding to a given iteration of the TransPhylo results*

Description

Return the combined tree corresponding to a given iteration of the TransPhylo results

Usage

```
extractCTree(res, iteration)
```

Arguments

`res`	Output from inferTTree command
`iteration`	Number of the iteration to be extracted

Value

The colored tree at the specified iteeatino

`extractPTree` *Extracts phylogenetic tree from a combined phylogenetic/transmission tree*

Description

Extracts phylogenetic tree from a combined phylogenetic/transmission tree

Usage

```
extractPTree(ctree)
```

Arguments

`ctree`	Combined tree

Value

phylogenetic tree

Examples

```
extractPTree(simulateOutbreak())
```

`extractTTree` *Extracts transmission tree from a combined phylogenetic/transmission tree*

Description

Extracts transmission tree from a combined phylogenetic/transmission tree

Usage

```
extractTTree(ctree)
```

Arguments

`ctree`	Combined tree

Value

transmission tree

Examples

```
extractTTree(simulateOutbreak())
```

`getGenerationTimeDist` *Extract and return realised generation time distribution*

Description

Extract and return realised generation time distribution

Usage

```
getGenerationTimeDist(
  record,
  burnin = 0.5,
  maxi = 2,
  numBins = 20,
  show.plot = F
)
```

Arguments

`record`	MCMC output produced by inferTTree
`burnin`	Proportion of the MCMC output to be discarded as burnin
`maxi`	Maximum generation time to consider
`numBins`	Number of time bins to compute and display distribution
`show.plot`	Show a barplot of the distribution

Value

Vector of times between becoming infected and infecting others (generation times) in the posterior

---

`getIncidentCases` *Returns and/or plot numbers of sampled and unsampled cases over time*

Description

Returns and/or plot numbers of sampled and unsampled cases over time

Usage

```
getIncidentCases(
  record,
  burnin = 0.5,
  numBins = 10,
  dateT = NA,
  show.plot = FALSE
)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin
`numBins`	Number of time bins to compute and display incident cases
`dateT`	Date when process stops (this can be Inf for fully resolved outbreaks)
`show.plot`	Show a plot of incident cases over time with stacked bars

Value

List with four entries. Time is a vector of the time points. allCases is the average number of cases at each time in the posterior. sampledCases: average number of sampled cases. unsampCases: average number of unsampled cases.

`getInfectionTimeDist` *Extract and return distribution of infection time of given sampled case(s)*

Description

Extract and return distribution of infection time of given sampled case(s)

Usage

```
getInfectionTimeDist(record, burnin = 0.5, k, numBins = 10, show.plot = F)
```

Arguments

`record`	MCMC output produced by inferTTree
`burnin`	Proportion of the MCMC output to be discarded as burnin
`k`	Case(s) whose posterior infection times are to be extracted. Either a string matching one of the case names in the data, or a vector of such strings
`numBins`	Number of bins to use for plot
`show.plot`	Show a barplot of the distribution

Value

Posterior infection times for the case(s) in k. If length(k)==1 then a vector is returned, otherwise a matrix

---

`getOffspringDist` *Extract and return offspring distribution of given sampled case(s)*

Description

Extract and return offspring distribution of given sampled case(s)

Usage

```
getOffspringDist(record, burnin = 0.5, k, show.plot = F)
```

Arguments

`record`	MCMC output produced by inferTTree
`burnin`	Proportion of the MCMC output to be discarded as burnin
`k`	Case(s) whose offspring distribution are to be extracted. Either a string matching one of the case names in the data, or a vector of such strings
`show.plot`	Show a barplot of the distribution

Value

Posterior offspring distribution for the case(s) in k. If length(k)==1 then a vector is returned, otherwise a matrix

---

`getSamplingTimeDist` *Extract and return realised sampling time distribution*

Description

Extract and return realised sampling time distribution

Usage

```
getSamplingTimeDist(
  record,
  burnin = 0.5,
  maxi = 2,
  numBins = 20,
  show.plot = F
)
```

Arguments

`record`	MCMC output produced by inferTTree
`burnin`	Proportion of the MCMC output to be discarded as burnin
`maxi`	Maximum generation time to consider
`numBins`	Number of time bins to compute and display distribution
`show.plot`	Show a barplot of the distribution

Value

Vector of times between becoming infected and becoming sampled in the posterior

---

`inferTTree` *Infer transmission tree given a phylogenetic tree*

Description

Infer transmission tree given a phylogenetic tree

Usage

```
inferTTree(
  ptree,
  w.shape = 2,
  w.scale = 1,
  ws.shape = NA,
  ws.scale = NA,
  w.mean = NA,
  w.std = NA,
  ws.mean = NA,
  ws.std = NA,
  mcmcIterations = 1000,
  thinning = 1,
  startNeg = 100/365,
  startOff.r = 1,
  startOff.p = 0.5,
  startPi = 0.5,
  updateNeg = TRUE,
  updateOff.r = TRUE,
  updateOff.p = FALSE,
  updatePi = TRUE,
  startCTree = NA,
  updateTTree = TRUE,
  optiStart = 2,
  dateT = Inf,
  verbose = F
)
```

Arguments

`ptree`	Phylogenetic tree
`w.shape`	Shape parameter of the Gamma distribution representing the generation time
`w.scale`	Scale parameter of the Gamma distribution representing the generation time
`ws.shape`	Shape parameter of the Gamma distribution representing the sampling time
`ws.scale`	Scale parameter of the Gamma distribution representing the sampling time
`w.mean`	Mean of the Gamma distribution representing the generation time
`w.std`	Std of the Gamma distribution representing the generation time
`ws.mean`	Mean of the Gamma distribution representing the sampling time
`ws.std`	Std of the Gamma distribution representing the sampling time
`mcmcIterations`	Number of MCMC iterations to run the algorithm for
`thinning`	MCMC thinning interval between two sampled iterations
`startNeg`	Starting value of within-host coalescent parameter Ne*g
`startOff.r`	Starting value of parameter off.r
`startOff.p`	Starting value of parameter off.p

`startPi`	Starting value of sampling proportion pi
`updateNeg`	Whether of not to update the parameter Ne*g
`updateOff.r`	Whether or not to update the parameter off.r
`updateOff.p`	Whether or not to update the parameter off.p
`updatePi`	Whether or not to update the parameter pi
`startCTree`	Optional combined tree to start from
`updateTTree`	Whether or not to update the transmission tree
`optiStart`	Type of optimisation to apply to MCMC start point (0=none, 1=slow, 2=fast)
`dateT`	Date when process stops (this can be Inf for fully simulated outbreaks)
`verbose`	Whether or not to use verbose mode (default is false)

Value

posterior sample set of transmission trees

Examples

```
inferTTree(ptreeFromPhylo(ape::rtree(5),2020),mcmcIterations=100)
```

`infer_multittree_share_param`

Simultaneously infer transmission trees given phylogenetic trees User can specify any subset of parameters that will be shared by providing a character vector of parameter names to the argument "share".

Description

Simultaneously infer transmission trees given phylogenetic trees User can specify any subset of parameters that will be shared by providing a character vector of parameter names to the argument "share".

Usage

```
infer_multittree_share_param(
  ptree_lst,
  w.shape = 2,
  w.scale = 1,
  ws.shape = w.shape,
  ws.scale = w.scale,
  mcmcIterations = 1000,
  thinning = 1,
  startNeg = 100/365,
  startOff.r = 1,
  startOff.p = 0.5,
  startPi = 0.5,
```

```
  prior_pi_a = 1,
  prior_pi_b = 1,
  updateNeg = TRUE,
  updateOff.r = TRUE,
  updateOff.p = FALSE,
  updatePi = TRUE,
  share = NULL,
  startCTree_lst = rep(NA, length(ptree_lst)),
  updateTTree = TRUE,
  optiStart = 2,
  dateT = Inf,
  verbose = F
)
```

Arguments

`ptree_lst`	List of phylogenetic tree
`w.shape`	Shape parameter of the Gamma probability density function representing the generation time
`w.scale`	Scale parameter of the Gamma probability density function representing the generation time
`ws.shape`	Shape parameter of the Gamma probability density function representing the sampling time
`ws.scale`	Scale parameter of the Gamma probability density function representing the sampling time
`mcmcIterations`	Number of MCMC iterations to run the algorithm for
`thinning`	MCMC thinning interval between two sampled iterations
`startNeg`	Starting value of within-host coalescent parameter Ne*g
`startOff.r`	Starting value of parameter off.r
`startOff.p`	Starting value of parameter off.p
`startPi`	Starting value of sampling proportion pi
`prior_pi_a`	First shape parameter of Beta prior for pi
`prior_pi_b`	Second shape parameter of Beta prior for pi
`updateNeg`	Whether of not to update the parameter Ne*g
`updateOff.r`	Whether or not to update the parameter off.r
`updateOff.p`	Whether or not to update the parameter off.p
`updatePi`	Whether or not to update the parameter pi
`share`	Character vector of parameters to be shared. For example, share = c("off.r", "off.p") would share the offspring distribution. Allowed parameter names are "neg", "off.r", "off.p" and "pi".
`startCTree_lst`	Optional combined list of trees to start from
`updateTTree`	Whether or not to update the transmission tree
`optiStart`	Type of optimisation to apply to MCMC start point (0=none, 1=slow, 2=fast)
`dateT`	Date when process stops (this can be Inf for fully simulated outbreaks)
`verbose`	Whether or not to use verbose mode (default is false)

Value

list the same size as input, each element contains posterior transmission trees inferred from corresponding phylogenetic tree

Author(s)

Yuanwei Xu

`makeCTreeFromPTree` *Create a transmission tree compatible with the provided phylogenetic tree*

Description

Create a transmission tree compatible with the provided phylogenetic tree

Usage

```
makeCTreeFromPTree(
  ptree,
  off.r = NA,
  off.p = NA,
  neg = NA,
  pi = NA,
  w.shape = NA,
  w.scale = NA,
  ws.shape = NA,
  ws.scale = NA,
  T = NA,
  optiStart = 0
)
```

Arguments

`ptree`	Phylogenetic tree
`off.r`	First parameter of the negative binomial distribution for offspring number
`off.p`	Second parameter of the negative binomial distribution for offspring number
`neg`	the within-host effective population size (Ne) timesgeneration duration (g)
`pi`	probability of sampling an infected individual
`w.shape`	Shape parameter of the Gamma probability density function representing the generation time
`w.scale`	Scale parameter of the Gamma probability density function representing the generation time
`ws.shape`	Shape parameter of the Gamma probability density function representing the sampling time

`ws.scale`	Scale parameter of the Gamma probability density function representing the sampling time
`T`	Date when process stops (this can be Inf for fully simulated outbreaks)
`optiStart`	Method used to optimised colored tree (0=none, 1=slow, 2=fast)

Value

A minimal non-zero probability phylogenetic+transmission tree, or an optimised version if parameters are provided

---

`makeTTree` *Simulate a transmission tree*

Description

Simulate a transmission tree

Usage

```
makeTTree(
  off.r,
  off.p,
  pi,
  w.shape,
  w.scale,
  ws.shape = w.shape,
  ws.scale = w.scale,
  maxTime = Inf,
  nSampled = NA
)
```

Arguments

`off.r`	First parameter of the negative binomial distribution for offspring number
`off.p`	Second parameter of the negative binomial distribution for offspring number
`pi`	probability of sampling an infected individual
`w.shape`	Shape parameter of the Gamma probability density function representing the generation time
`w.scale`	Scale parameter of the Gamma probability density function representing the generation time
`ws.shape`	Shape parameter of the Gamma probability density function representing the sampling time
`ws.scale`	Scale parameter of the Gamma probability density function representing the sampling time
`maxTime`	Duration of simulation (can be Inf)
`nSampled`	Number of sampled individuals (can be NA for any)

Value

A N*3 matrix in the following format with one row per infected host, first column is time of infection, second column is time of sampling, third column is infector

---

`medTTree` *Return the medoid from a MCMC output*

Description

Return the medoid from a MCMC output

Usage

```
medTTree(record, burnin = 0.5)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin

Value

The index of the medoid

`phyloFromPTree` *Converts a phylogenetic tree into an ape phylo object*

Description

Converts a phylogenetic tree into an ape phylo object

Usage

```
phyloFromPTree(ptree)
```

Arguments

`ptree`	phylogenetic tree

Value

phylo object

Examples

```
phyloFromPTree(extractPTree(simulateOutbreak()))
```

`plot.ctree` *Plotting for ctree*

Description

Plotting for ctree

Usage

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

Arguments

`x`	Object of class ctree, ie a colored phylogenetic tree
`...`	Additional parameters are passed on

Value

Plot of ctree

Examples

```
plot(simulateOutbreak())
```

`plot.ptree` *Plotting for ptree*

Description

Plotting for ptree

Usage

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

Arguments

`x`	Object of class ptree, ie a phylogenetic tree
`...`	Additional parameters are passed on to ape::plot.phylo

Value

Plot of ptree

Examples

```
plot(ptreeFromPhylo(ape::rtree(5),2020))
```

`plot.resTransPhylo` *Plotting for resTransPhylo*

Description

Plotting for resTransPhylo

Usage

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

Arguments

`x`	Output from inferTTree
`...`	Additional parameters are passed on

Value

Plot of TransPhylo results

`plot.ttree` *Plotting for ttree*

Description

Plotting for ttree

Usage

```
## S3 method for class 'ttree'
plot(x, type = "summarised", w.shape = NA, w.scale = NA, ...)
```

Arguments

`x`	Object of class ttree, ie a transmission tree
`type`	Type of plot to display, can be 'detailed' or 'summarised' (default)
`w.shape`	Shape parameter of the generation time, needed for detailed plot only
`w.scale`	Scale parameter of the generation time, needed for detailed plot only
`...`	Additional parameters are passed on

Value

Plot of ttree

Examples

```
plot(extractTTree(simulateOutbreak()))
```

`plotCTree` *Plot both phylogenetic and transmission trees using colors on the phylogeny*

Description

Plot both phylogenetic and transmission trees using colors on the phylogeny

Usage

```
plotCTree(
  tree,
  showLabels = TRUE,
  showStars = TRUE,
  cols = NA,
  maxTime = NA,
  cex = 1
)
```

Arguments

`tree`	Combined phylogenetic/transmission tree
`showLabels`	Whether or not to show the labels
`showStars`	Whether or not to show stars representing transmission events
`cols`	Colors to use for hosts
`maxTime`	Maximum time to show on the x axis
`cex`	Expansion factor

Value

Returns invisibly the first parameter

Examples

```
plotCTree(simulateOutbreak())
```

`plotTraces` *Plot MCMC traces*

Description

Plot MCMC traces

Usage

```
plotTraces(record, burnin = 0, extend = F)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin
`extend`	Whether to also show traces of off.r and off.p

Value

Returns invisibly the first parameter

`plotTTree` *Plot a transmission tree in a detailed format*

Description

Plot a transmission tree in a detailed format

Usage

```
plotTTree(ttree, w.shape, w.scale, showLabels = TRUE, maxTime = NA, cex = 1)
```

Arguments

`ttree`	Transmission tree
`w.shape`	Shape parameter of the Gamma probability density function representing the generation time
`w.scale`	Scale parameter of the Gamma probability density function representing the generation time
`showLabels`	Whether or not to show the labels
`maxTime`	Maximum value of time to show on x axis
`cex`	Expansion factor

Value

Returns invisibly the first parameter

Examples

```
plotTTree(extractTTree(simulateOutbreak()),2,1)
```

`plotTTree2` *Plot a transmission tree in an economic format*

Description

Plot a transmission tree in an economic format

Usage

```
plotTTree2(
  ttree,
  showLabels = TRUE,
  showMissingLinks = 0,
  maxTime = NA,
  cex = 1
)
```

Arguments

`ttree`	Transmission tree
`showLabels`	Boolean for whether or not to show the labels
`showMissingLinks`	Option for how to show missing links: (0) as dots, (1) as several gray levels, (2) as a single gray level
`maxTime`	Maximum value of time to show on x axis
`cex`	Expansion factor

Value

Returns invisibly the first parameter

Examples

```
plotTTree2(extractTTree(simulateOutbreak()))
```

`print.ctree` *Print function for ctree objects*

Description

Print function for ctree objects

Usage

```
## S3 method for class 'ctree'
print(x, ...)
```

Arguments

`x`	Object of class ctree, ie a colored phylogenetic tree
`...`	Additional parameters are passed on

Value

Print out details of the ctree

Examples

```
print(simulateOutbreak())
```

`print.ptree` *Print function for ptree objects*

Description

Print function for ptree objects

Usage

```
## S3 method for class 'ptree'
print(x, ...)
```

Arguments

`x`	Object of class ptree, ie a phylogenetic tree
`...`	Additional parameters are passed on

Value

Print out details of the ptree

Examples

```
print(extractPTree(simulateOutbreak()))
```

`print.resTransPhylo` *Print function for resTransPhylo objects*

Description

Print function for resTransPhylo objects

Usage

```
## S3 method for class 'resTransPhylo'
print(x, ...)
```

Arguments

`x`	output from inferTTree
`...`	Additional parameters are passed on

Value

Print out details of TransPhylo results

`print.ttree` *Print function for ttree objects*

Description

Print function for ttree objects

Usage

```
## S3 method for class 'ttree'
print(x, ...)
```

Arguments

`x`	Object of class ttree, ie a transmission tree
`...`	Additional parameters are passed on

Value

Print out details of the ttree

Examples

```
print(extractTTree(simulateOutbreak()))
```

`probPTreeGivenTTree` *Calculate the probability of a phylogenetic tree given a transmission tree*

Description

Calculate the probability of a phylogenetic tree given a transmission tree

Usage

```
probPTreeGivenTTree(ctree, neg, w = integer(0))
```

Arguments

`ctree`	Combined phylogenetic/transmission tree
`neg`	Within-host coalescent rate
`w`	Vector of hosts for which to calculate the probability, or nothing for all

Value

Probability of phylogeny given transmission tree

`probPTreeGivenTTreeR` *Calculate the probability of a phylogenetic tree given a transmission tree*

Description

Calculate the probability of a phylogenetic tree given a transmission tree

Usage

```
probPTreeGivenTTreeR(ctree, neg, w = NULL)
```

Arguments

`ctree`	Combined phylogenetic/transmission tree
`neg`	Within-host coalescent rate
`w`	Vector of hosts for which to calculate the probability, or NULL for all

Value

Probability of phylogeny given transmission tree

`probTTree` *Calculates the log-probability of a transmission tree*

Description

Calculates the log-probability of a transmission tree

Usage

```
probTTree(
  ttree,
  rOff,
  pOff,
  pi,
  shGen,
  scGen,
  shSam,
  scSam,
  dateT,
  delta_t = 0.01
)
```

Arguments

`ttree`	Transmission tree
`rOff`	First parameter of the negative binomial distribution for offspring number
`pOff`	Second parameter of the negative binomial distribution for offspring number
`pi`	probability of sampling an infected individual
`shGen`	Shape parameter of the Gamma probability density function representing the generation time
`scGen`	Scale parameter of the Gamma probability density function representing the generation time
`shSam`	Shape parameter of the Gamma probability density function representing the sampling time
`scSam`	Scale parameter of the Gamma probability density function representing the sampling time
`dateT`	Date when process stops (this can be Inf for fully simulated outbreaks)
`delta_t`	Grid precision

Value

Probability of the transmission tree

`probTTreeR` *Calculates the log-probability of a transmission tree*

Description

Calculates the log-probability of a transmission tree

Usage

```
probTTreeR(
  ttree,
  off.r,
  off.p,
  pi,
  w.shape,
  w.scale,
  ws.shape,
  ws.scale,
  dateT
)
```

Arguments

`ttree`	Transmission tree
`off.r`	First parameter of the negative binomial distribution for offspring number
`off.p`	Second parameter of the negative binomial distribution for offspring number
`pi`	probability of sampling an infected individual
`w.shape`	Shape parameter of the Gamma probability density function representing the generation time
`w.scale`	Scale parameter of the Gamma probability density function representing the generation time
`ws.shape`	Shape parameter of the Gamma probability density function representing the sampling time
`ws.scale`	Scale parameter of the Gamma probability density function representing the sampling time
`dateT`	Date when process stops (this can be Inf for fully simulated outbreaks)

Value

Probability of the transmission tree

`ptreeFromPhylo` *Converts an ape phylo object into a phylogenetic tree*

Description

Converts an ape phylo object into a phylogenetic tree

Usage

```
ptreeFromPhylo(tr, dateLastSample)
```

Arguments

`tr`	phylo object
`dateLastSample`	date of the last sample

Value

phylogenetic tree

Examples

```
ptreeFromPhylo(ape::rtree(5),2020)
```

`selectTTree` *Select the most representative transmission tree from a MCMC output*

Description

Select the most representative transmission tree from a MCMC output

Usage

```
selectTTree(record, burnin = 0.5)
```

Arguments

`record`	Output from inferTTree function
`burnin`	Proportion of the MCMC output to be discarded as burnin

Value

The index of the selected transmission tree

`simulateOutbreak` *Simulate an outbreak*

Description

Simulate an outbreak

Usage

```
simulateOutbreak(
  off.r = 1,
  off.p = 0.5,
  neg = 0.25,
  nSampled = NA,
  pi = 0.5,
  w.shape = 2,
  w.scale = 1,
  ws.shape = NA,
  ws.scale = NA,
  w.mean = NA,
  w.std = NA,
  ws.mean = NA,
  ws.std = NA,
  dateStartOutbreak = 2000,
  dateT = Inf
)
```

Arguments

`off.r`	First parameter of the negative binomial distribution for offspring number
`off.p`	Second parameter of the negative binomial distribution for offspring number
`neg`	the within-host effective population size (Ne) timesgeneration duration (g)
`nSampled`	number of sampled infected individuals, or NA for any
`pi`	probability of sampling an infected individual
`w.shape`	Shape parameter of the Gamma probability density function representing the generation time
`w.scale`	Scale parameter of the Gamma probability density function representing the generation time
`ws.shape`	Shape parameter of the Gamma probability density function representing the sampling time
`ws.scale`	Scale parameter of the Gamma probability density function representing the sampling time
`w.mean`	Mean of the Gamma distribution representing the generation time
`w.std`	Std of the Gamma distribution representing the generation time

`ws.mean`	Mean of the Gamma distribution representing the sampling time
`ws.std`	Std of the Gamma distribution representing the sampling time
`dateStartOutbreak`	Date when index case becomes infected
`dateT`	Date when process stops (this can be Inf for fully simulated outbreaks)

Value

Combined phylogenetic and transmission tree

Examples

```
simulateOutbreak()
simulateOutbreak(off.r=2,dateStartOutbreak=2010,dateT=2015)
```

`summary.resTransPhylo` *Summary function for resTransPhylo objects*

Description

Summary function for resTransPhylo objects

Usage

```
## S3 method for class 'resTransPhylo'
summary(object, ...)
```

Arguments

`object`	output from inferTTree
`...`	Passed on to print.phylo

Value

Print out details of TransPhylo results

Index