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>,
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 .................................................. 2
as.mcmc.resTransPhylo ............................................. 3
computeMatTDDist .................................................. 3
computeMatWDist .................................................. 4
Inference of transmission tree from a dated phylogeny. Includes methods to simulate and analyse outbreaks.
as.mcmc.resTransPhylo

Author(s)

Xavier Didelot <xavier.didelot@gmail.com>

References


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

<table>
<thead>
<tr>
<th>record</th>
<th>Output from inferTTree function</th>
</tr>
</thead>
<tbody>
<tr>
<td>burnin</td>
<td>Proportion of the MCMC output to be discarded as burnin</td>
</tr>
</tbody>
</table>

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

Arguments

<table>
<thead>
<tr>
<th>record</th>
<th>Output from inferTTree function</th>
</tr>
</thead>
<tbody>
<tr>
<td>burnin</td>
<td>Proportion of the MCMC output to be discarded as burnin</td>
</tr>
</tbody>
</table>

Value

Matrix of probability of who infected whom

consTTree

Build a consensus transmission tree from a MCMC output

Arguments

<table>
<thead>
<tr>
<th>record</th>
<th>Output from inferTTree function</th>
</tr>
</thead>
<tbody>
<tr>
<td>burnin</td>
<td>Proportion of the MCMC output to be discarded as burnin</td>
</tr>
<tr>
<td>minimum</td>
<td>Minimum probability for inclusion of a partition in the consensus</td>
</tr>
<tr>
<td>debug</td>
<td>Used for debugging</td>
</tr>
</tbody>
</table>

Usage

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

**Value**  
The consensus transmission tree

---

**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 iteration

---

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**

```r
evaluateTTree(ctree)
```

**Arguments**

- **ctree**: Combined tree

**Value**

transmission tree

**Examples**

```r
evaluateTTree(simulateOutbreak())
```

**getGenerationTimeDist**

*Extract and return realised generation time distribution*

**Description**

Extract and return realised generation time distribution

**Usage**

```r
generateGenerationTimeDist(
  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
getIncidentCases

Value

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

Usage

getIncidentCases(
  record,  # Output from inferTTree function
  burnin = 0.5,  # Proportion of the MCMC output to be discarded as burnin
  numBins = 10,  # Number of time bins to compute and display incident cases
  dateT = NA,  # Date when process stops (this can be Inf for fully resolved outbreaks)
  show.plot = FALSE  # Show a plot of incident cases over time with stacked bars
)

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**

```r
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**

```r
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**

```r
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

```r
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
infer_multittree_share_param

- **startPi**: Starting value of sampling proportion \(\pi\)
- **updateNeg**: Whether or not to update the parameter \(N_{\text{e}*g}\)
- **updateOff.r**: Whether or not to update the parameter \(\text{off.r}\)
- **updateOff.p**: Whether or not to update the parameter \(\text{off.p}\)
- **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**

```r
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**

```r
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,
```
infer_multitree_share_param

```r
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 $N_e^g$
- `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 or not to update the parameter $N_e^g$
- `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**

```r
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) times generation 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
makeTTree

- `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

---

**Description**

Simulate a transmission tree

**Usage**

```r
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

Description
Plotting for ctree

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

Arguments
x Object of class ctree, ie a colored phylogenetic tree
...

Value
Plot of ctree

Examples
plot(simulateOutbreak())

plot.ptree

Description
Plotting for ptree

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

Arguments
x Object of class ptree, ie a phylogenetic tree
...

Value
Plot of ptree
plot.ttree

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
Plot both phylogenetic and transmission trees using colors on the phylogeny

Description

Plot both phylogenetic and transmission trees using colors on the phylogeny

Usage

```r
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

```r
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
plotTTree2

Value

Returns invisibly the first parameter

Examples

plotTTree2(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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttree</td>
<td>Transmission tree</td>
</tr>
<tr>
<td>showLabels</td>
<td>Boolean for whether or not to show the labels</td>
</tr>
<tr>
<td>showMissingLinks</td>
<td>Option for how to show missing links: (0) as dots, (1) as several gray levels, (2) as a single gray level</td>
</tr>
<tr>
<td>maxTime</td>
<td>Maximum value of time to show on x axis</td>
</tr>
<tr>
<td>cex</td>
<td>Expansion factor</td>
</tr>
</tbody>
</table>

Value

Returns invisibly the first parameter

Examples

plotTTree2(extractTTree(simulateOutbreak()))
### print.ctree

**Print function for ctree objects**

**Description**

Print function for ctree objects

**Usage**

```r
## 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**

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

### print.ptree

**Print function for ptree objects**

**Description**

Print function for ptree objects

**Usage**

```r
## 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
print.resTransPhylo

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
...

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
...

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**

```r
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 \hspace{1cm} \textit{Calculates the log-probability of a transmission tree}

**Description**

Calculates the log-probability of a transmission tree

**Usage**

\begin{verbatim}
probTTree(
tree,
    rOff,
pOff,
    pi,
    shGen,
    scGen,
    shSam,
    scSam,
    dateT,
    delta_t = 0.01
)
\end{verbatim}

**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  \textit{Calculates the log-probability of a transmission tree}

\textbf{Description}

Calculates the log-probability of a transmission tree

\textbf{Usage}

\begin{verbatim}
probTTreeR(
ttree,
off.r,
off.p,
pi,
w.shape,
w.scale,
ws.shape,
ws.scale,
dateT
)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{ttree}  \hspace{1cm} Transmission tree
  \item \texttt{off.r}  \hspace{1cm} First parameter of the negative binomial distribution for offspring number
  \item \texttt{off.p}  \hspace{1cm} Second parameter of the negative binomial distribution for offspring number
  \item \texttt{pi}  \hspace{1cm} Probability of sampling an infected individual
  \item \texttt{w.shape}  \hspace{1cm} Shape parameter of the Gamma probability density function representing the generation time
  \item \texttt{w.scale}  \hspace{1cm} Scale parameter of the Gamma probability density function representing the generation time
  \item \texttt{ws.shape}  \hspace{1cm} Shape parameter of the Gamma probability density function representing the sampling time
  \item \texttt{ws.scale}  \hspace{1cm} Scale parameter of the Gamma probability density function representing the sampling time
  \item \texttt{dateT}  \hspace{1cm} Date when process stops (this can be Inf for fully simulated outbreaks)
\end{itemize}

\textbf{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**

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

**Arguments**

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

**Value**

phylogenetic tree

**Examples**

```r
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**

```r
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**

```r
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) times generation 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
summary.resTransPhylo

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

as.mcmc.resTransPhylo, 3
computeMatTDist, 3
computeMatWIW, 4
consTTree, 4
convertToCoda, 5
dateLastSample, 5
extractCTree, 6
extractPTree, 6
extractTTree, 7
getGenerationTimeDist, 7
getIncidentCases, 8
getInfectionTimeDist, 9
getOffspringDist, 9
getSamplingTimeDist, 10
infer_multittree_share_param, 12
inferTTree, 10
makeCTreeFromPTree, 14
makeTTree, 15
medTTree, 16
phyloFromPTree, 16
plot.ctree, 17
plot.ptree, 17
plot.resTransPhylo, 18
plot.ttree, 18
plotCTree, 19
plotTraces, 20
plotTTree, 20
plotTTree2, 21
print.ctree, 22
print.ptree, 22
print.resTransPhylo, 23
print.ttree, 23
probPTreeGivenTTree, 24
probPTreeGivenTTreeR, 24
probTTTree, 25
probTTreeR, 26
ptreeFromPhylo, 27
selectTTree, 27
simulateOutbreak, 28
summary.resTransPhylo, 29
TransPhylo (TransPhylo-package), 2
TransPhylo-package, 2