

node.dating

February 5, 2018

node.dating

node.dating

Description

Estimate the dates of a rooted phylogenetic tree from the tip dates.

Usage

```
estimate.mu(t, node.dates, p = 0.05)
estimate.dates(t, node.dates, mu = estimate.mu(t, node.dates), min.date =
  -.Machine$double.xmax, show.steps = 0, opt.tol = 1e-8, nsteps = 1000, lik.tol =
  if (nsteps <= 0) opt.tol else 0, is.binary = is.binary.tree(t))
```

Arguments

t	an object of class "phylo"
node.dates	a numeric vector of dates for the tips, in the same order as 't\$tip.label' or a vector of dates for all of the nodes.
p	p-value cutoff for failed regression.
mu	mutation rate.
min.date	the minimum bound on the dates of nodes
show.steps	print the log likelihood every show.steps. If 0 will suppress output.
opt.tol	tolerance for optimization precision.
lik.tol	tolerance for likelihood comparison.
nsteps	the maximum number of steps to run.
is.binary	if TRUE, will run a faster optimization method that only works if the tree is binary; otherwise will use optimize() as the optimization method.

Details

This code duplicates the functionality of the program Tip.Dates (see references). The dates of the internal nodes of 't' are estimated using a maximum likelihood approach.

't' must be rooted and have branch lengths in units of expected substitutions per site.

'node.dates' can be either a numeric vector of dates for the tips or a numeric vector for all of the nodes of 't'. 'estimate.mu' will use all of the values given in 'node.dates' to estimate the mutation rate. Dates can be censored with NA. 'node.dates' must contain all of the tip dates when it is a parameter of 'estimate.dates'. If only tip dates are given, then 'estimate.dates' will run an initial step to estimate the dates of the internal nodes. If 'node.dates' contains dates for some of the nodes, 'estimate.dates' will use those dates as priors in the initial step. If all of the dates for nodes are given, then 'estimate.dates' will not run the initial step.

If 'is.binary' is set to FALSE, 'estimate.dates' uses the "optimize" function as the optimization method. By default, R's "optimize" function uses a precision of ".Machine\$double.eps^0.25", which is about 0.0001 on a 64-bit system. This should be set to a smaller value if the branch lengths of 't' are very short. If 'is.binary' is set to TRUE, estimate dates uses calculus to determine the maximum likelihood at each step, which is faster. The bounds of permissible values are reduced by 'opt.tol'.

'estimate.dates' has several criteria to decide how many steps it will run. If 'lik.tol' and 'nsteps' are both 0, then 'estimate.dates' will only run the initial step. If 'lik.tol' is greater than 0 and 'nsteps' is 0, then 'estimate.dates' will run until the difference between successive steps is less than 'lik.tol'. If 'lik.tol' is 0 and 'nsteps' is greater than 0, then 'estimate.dates' will run the initial step and then 'nsteps' steps. If 'lik.tol' and 'nsteps' are both greater than 0, then 'estimate.dates' will run the initial step and then either 'nsteps' steps or until the difference between successive steps is less than 'lik.tol'.

Value

The estimated mutation rate as a numeric vector of length one for estimate.mu.

The estimated dates of all of the nodes of the tree as a numeric vector with length equal to the number of nodes in the tree.

Note

This model assumes that the tree follows a molecular clock. It only performs a rudimentary statistical test of the molecular clock hypothesis.

Author(s)

Bradley R. Jones <email: brj1@sfu.ca>

References

- Felsenstein, J. (1981) Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, **17**, 368–376.
- Rambaut, A. (2000) Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies. *Bioinformatics*, **16**, 395–399.
- Jones, Bradley R., and Poon, Art F. Y. (2016) node.dating: dating ancestors in phylogenetic trees in R *Bioinformatics*, **33**, 932–934.

See Also[optimize](#), [rtt](#)**Examples**

```
t <- rtree(100)
tip.date <- rnorm(t$tip.label, mean = node.depth.edglength(t)[1:Ntip(t)])^2
t <- rtt(t, tip.date)
mu <- estimate.mu(t, tip.date)

## Run for 100 steps
node.date <- estimate.dates(t, tip.date, mu, nsteps = 100)

## Run until the difference between successive log likelihoods is
## less than  $10^{-4}$  starting with the 100th step's results
node.date <- estimate.dates(t, node.date, mu, nsteps = 0, lik.tol = 1e-4)

## To rescale the tree over time
t$edge.length <- node.date[t$edge[, 2]] - node.date[t$edge[, 1]]
```

Index

`estimate.dates (node.dating), 1`

`estimate.mu (node.dating), 1`

`node.dating, 1`

`optimize, 3`

`rtt, 3`