Package 'ouch'

August 17, 2021

R topics documented:

---

ouch-package *Ornstein-Uhlenbeck methods for comparative phylogenetic hypotheses*

Description

The **ouch** package provides facilities for phylogenetic comparative analysis based on Ornstein-Uhlenbeck models of trait evolution along a phylogeny. Multivariate data and complex adaptive hypotheses are supported.

Classes

The basic class, `ouchtree`, is provided to encode a phylogenetic tree. Plot and print methods are provided.

The class `browntree` derives from class `ouchtree` and encodes the results of fitting a Brownian Motion model to data.

The class `hansentree` derives from class `ouchtree` and encodes the results of fitting a Hansen model to data.

Detailed Documentation

- Phylogenies in **ouch** format: `ouchtree()`, `ape2ouch()`
- Brownian motion models: `brown()`
- Ornstein-Uhlenbeck models: `hansen()`, `paint()`
- Simulation of models: `simulate()`
- Display of data: `plot()`
- Extraction of information from fitted models: `summary()`, `logLik()`, `coef()`
- Example datasets: `anolis.ssd`, `bimac`

Author(s)

Aaron A. King

References

Butler, M.A. and A.A. King. 2004. Phylogenetic comparative analysis: a modeling approach for adaptive evolution. Am. Nat. 164:683–695.

Cressler, C. E., Butler, M. A., and King, A. A. 2015. Detecting adaptive evolution in phylogenetic comparative analysis using the Ornstein-Uhlenbeck model. Syst. Biol., 64:953–968.

See Also

Other phylogenetic comparative models: `brown()`, `hansen()`, `ouchtree`, `paint()`

Other methods for ouch trees: `bootstrap()`, `coef()`, `logLik`, `paint()`, `plot()`, `print()`, `simulate()`, `summary()`, `update()`

Other examples: `anolis.ssd`, `bimac`

anolis.ssd *Greater Antillean anolis lizard sexual size dimorphism data*

Description

The dataset consists of sexual size-dimorphism data for 38 species of anoles from Cuba, Hispaniola, Jamaica, and Puerto Rico (Butler, Schoener, and Losos 2000). Each of these species belongs to one of six microhabitat types, or "ecomorphs" (sensu Williams, 1972): trunk-ground, grass-bush, trunk, trunk-crown, twig, and crown-giant. The data were used to demonstrate an evolutionary association between habitat type and degree of sexual size dimorphism.

Format

A data frame with 38 observations on the following 6 variables.

- `node`: Labels for the nodes.
- `species`: Names of extant species.
- `log.SSD`: Log sexual size dimorphism of extant species.
- `ancestor`: Name of ancestor node.
- `time`: Time of node.
- `OU.1`: a factor with one level, `ns`.
- `OU.7`: a factor with levels corresponding to ecomorph (`tg`, `tc`, `gb`, `cg`, `tw`, `tr`, `anc`).

Details

Size dimorphism was calcuated as the log-ratio of male snout-to-vent length to female snout-to-vent length (males are larger).

In this example, we tested three models of evolution: Brownian motion, Ornstein-Uhlenbeck with one global optimum, and Ornstein-Uhlenbeck with seven optima (one for each ecomorph type plus an additional one for an "unknown" type).

For the seven-optima model, we assigned each terminal branch to an optimum according to the ecomorph type of the extant species. Because we had no information to help guide hypotheses about internal branches, we assigned internal branches to the "unknown" selective regime. The phylogeny of these species is consistent with and adaptive radiation, with a burst of speciation events early in the evolutionary history of this clade (see phylogeny in Butler & King (2004) or example below).

Author(s)

Marguerite A. Butler, Aaron A. King

Source

Butler, M.A. and A.A. King. 2004. Phylogenetic comparative analysis: a modeling approach for adaptive evolution. Am. Nat. 164:683–695.

References

Butler, M. A., T. W. Schoener, and J. B. Losos. 2000. The relationship between sexual size dimorphism and habitat use in Greater Antillean Anolis lizards. Evolution, 54:259–272.

Williams, E. E. 1972. The origin of faunas. Evolution of lizard congeners in a complex island fauna: a trial analysis. Evol. Biol., 6:47–89.

See Also

Other examples: bimac, ouch-package

Examples

```
## Analysis of sexual size dimorphism data
tree <- with(anolis.ssd,ouchtree(node,ancestor,time/max(time),species))
plot(tree,node.names=TRUE)

h1 <- brown(anolis.ssd['log.SSD'],tree)
h1
plot(h1)

h2 <- hansen(anolis.ssd['log.SSD'],tree,anolis.ssd['OU.1'],sqrt.alpha=1,sigma=1)
h2
plot(h2)

h3 <- hansen(anolis.ssd['log.SSD'],tree,anolis.ssd['OU.7'],sqrt.alpha=1,sigma=1)
h3
plot(h3)
```

---

bimac *Anolis bimaculatus lizard size data*

Description

This is the *Anolis bimaculatus* dataset used in Butler & King (2004). It is used to test a hypothesis of character displacement using an interspecific dataset of body sizes and current data on sympatry/allopatry.

Format

A data frame with 45 observations on the following 11 variables.

- node: Labels for the nodes.
- spcode: Two-letter code for each taxon.
- species: Species names for extant species.
- island: Name of the island on which the population is found.
- size: Body size (head length in mm) of extant species.
- ancestor: Ancestral node.

- `time`: Time of node.
- `OU.1`: a factor with levels `ns`
- `OU.3`: a factor with levels `small`, `medium`, `large`
- `OU.4`: a factor with levels `small`, `medium`, `large`, `anc`
- `OU.LP`: a factor with levels `small`, `medium`, `large`

Details

Explanations of the data follow:

- **Body size.** We use the phenotypic data and phylogeny of Losos (1990), which employed the head lengths (of males) as a proxy for body size. In this group of lizards, head length correlates very strongly with snout-to-vent length and the cube root of mass, which are standard measures of body size. The data are head lengths in mm; note that we use the log of this value in analyses.
- **Tree structure.** The phylogenetic tree is encoded via three variables: `node`, `ancestor`, and `time`. The `node` variable gives a name to each node. The `ancestor` variable names the ancestor of each node. The root node has no ancestor (i.e., `ancestor=NA`). The variable `time` specifies the temporal location of each node, the root node being at time 0.
- **Specifications of selective regimes.** (Columns `OU.1`, `OU.3`, `OU.4`, `OU.LP`). These columns are factors, the levels of which correspond to the "paintings" of the respective adaptive regime hypotheses onto the phylogeny (see `paint()`). Each selective regime is named (small, medium, large, etc.). Each column corresponds to a different painting of the selective regimes, and thus to a different hypothesis. In this example, there are 3 alternative models (see Butler & King 2004): `OU.4` is 4-regime model, `OU.3` is 3-regime model (all ancestors are medium), `OU.LP` is the linear parsimony model.
- **Other variables.** In addition to the above, there is a two-letter code for each taxon (`spcode`) and the name of the island on which the taxon is found (`island`).

Author(s)

Marguerite A. Butler and Aaron A. King

Source

Butler, M.A. and A.A. King. 2004. Phylogenetic comparative analysis: a modeling approach for adaptive evolution. Am. Nat. 164:683–695.

References

Lazell, J. D. 1972. The anoles (Sauria: Iguanidae) of the Lesser Antilles. Bull. Mus. Comp. Zool., 143:1–115.

Losos, J. B. 1990. A phylogenetic analysis of character displacement in Caribbean Anolis lizards. Evolution, 44:558–569.

See Also

Other examples: `anolis.ssd`, `ouch-package`

Examples

```
## Analysis of Anolis bimaculatus data
tree <- with(bimac,ouchtree(node,ancestor,time/max(time),spcode))
plot(tree,node.names=TRUE)

h1 <- brown(log(bimac['size']),tree)
h1
plot(h1)

h2 <- hansen(log(bimac['size']),tree,bimac['OU.1'],sqrt.alpha=1,sigma=1)
h2
plot(h2)

h3 <- hansen(log(bimac['size']),tree,bimac['OU.3'],sqrt.alpha=1,sigma=1)
h3
plot(h3)

h4 <- hansen(log(bimac['size']),tree,bimac['OU.4'],sqrt.alpha=1,sigma=1)
h4
plot(h4)

h5 <- hansen(log(bimac['size']),tree,bimac['OU.LP'],sqrt.alpha=1,sigma=1,reltol=1e-5)
h5 <- update(h5,method='subplex',reltol=1e-11,parscale=c(0.1,0.1),hessian=TRUE)
h5
plot(h5)

simdat <- simulate(h5,nsim=10)
hsim <- update(h5,data=simdat[[1]])
summary(hsim)
bsim <- update(h1,data=simdat[[1]])
summary(bsim)
```

---

bootstrap *Bootstrapping for uncertainty quantification*

Description

Parametric bootstrapping for **ouch** models.

Usage

```
## S4 method for signature 'missing'
bootstrap(object, ...)

## S4 method for signature 'ANY'
bootstrap(object, ...)

## S4 method for signature 'browntree'
bootstrap(object, nboot = 200, seed = NULL, ...)
```

```
## S4 method for signature 'hansentree'
bootstrap(object, nboot = 200, seed = NULL, ...)
```

Arguments

object	A fitted model object.
...	Additional arguments are passed to update.
nboot	integer; number of bootstrap replicates.
seed	integer; setting seed to a non-NULL value allows one to fix the random seed (see simulate).

Details

bootstrap performs a parametric bootstrap for estimation of confidence intervals.

See Also

Other methods for ouch trees: coef(), logLik, ouch-package, paint(), plot(), print(), simulate(), summary(), update()

Examples

```
## Not run:
## Fit BM and a 5-regime OU model to the A. bimaculatus data
tree <- with(bimac,ouchtree(node,ancestor,time/max(time),species))

h1 <- brown(
  data=log(bimac['size']),
  tree=tree
)

h5 <- hansen(
  data=log(bimac['size']),
  tree=tree,
  regimes=bimac['OU.LP'],
  sqrt.alpha=1,
  sigma=1,
  reltol=1e-11,
  parscale=c(0.1,0.1),
  hessian=TRUE
)

## What are appropriate AIC.c cutoffs?
simdat <- simulate(h1,nsim=100,seed=92759587)
b1 <- sapply(simdat,function(x)summary(update(h1,data=x))$aic.c)
tic <- Sys.time()
b5 <- sapply(simdat,function(x)summary(update(h5,data=x))$aic.c)
toc <- Sys.time()
print(toc-tic)
cat("approximate 95% AIC.c cutoff",signif(quantile(b1-b5,0.95),digits=3),"\n")
```

```
## Bootstrap confidence intervals
boots.h1 <- bootstrap(h1,nboot=200,seed=92759587)
cat("bootstrap 95% confidence intervals for h1:\n")
print(t(sapply(boots.h1,quantile,probs=c(0.025,0.975))),digits=3)

boots.h5 <- bootstrap(h5,nboot=200,seed=92759587)
cat("bootstrap 95% confidence intervals for h5:\n")
print(t(sapply(boots.h5,quantile,probs=c(0.025,0.975))),digits=3)

## End(Not run)
```

`brown` *Phylogenetic Brownian motion models*

Description

The function `brown` creates a `browntree` object by fitting a Brownian-motion model to data.

Usage

```
brown(data, tree)
```

Arguments

`data`	Phenotypic data for extant species, i.e., at the terminal ends of the phylogenetic tree. This can either be a numeric vector or a list. If it is a numeric vector, there must be one entry for every node. If it is a list, it must consist entirely of numeric vectors, each of which has one entry per node. A data-frame is coerced to a list.
`tree`	A phylogenetic tree, specified as an `ouchtree` object.

Value

`brown` returns an object of class `browntree`.

Author(s)

Aaron A. King

References

Butler, M.A. and A.A. King. 2004. Phylogenetic comparative analysis: a modeling approach for adaptive evolution. Am. Nat. 164:683–695.

See Also

`bimac`, `anolis.ssd`, `hansen`

Other phylogenetic comparative models: `hansen()`, `ouch-package`, `ouchtree`, `paint()`

Examples

```
## Analysis of Anolis bimaculatus data
tree <- with(bimac,ouchtree(node,ancestor,time/max(time),spcode))
plot(tree,node.names=TRUE)

h1 <- brown(log(bimac['size']),tree)
h1
plot(h1)

h2 <- hansen(log(bimac['size']),tree,bimac['OU.1'],sqrt.alpha=1,sigma=1)
h2
plot(h2)

h3 <- hansen(log(bimac['size']),tree,bimac['OU.3'],sqrt.alpha=1,sigma=1)
h3
plot(h3)

h4 <- hansen(log(bimac['size']),tree,bimac['OU.4'],sqrt.alpha=1,sigma=1)
h4
plot(h4)

h5 <- hansen(log(bimac['size']),tree,bimac['OU.LP'],sqrt.alpha=1,sigma=1,reltol=1e-5)
h5 <- update(h5,method='subplex',reltol=1e-11,parscale=c(0.1,0.1),hessian=TRUE)
h5
plot(h5)

simdat <- simulate(h5,nsim=10)
hsim <- update(h5,data=simdat[[1]])
summary(hsim)
bsim <- update(h1,data=simdat[[1]])
summary(bsim)
```

---

coef *Model coefficients*

Description

coef extracts the parameters from a fitted model object.

Usage

```
## S4 method for signature 'browntree'
coef(object, ...)

## S4 method for signature 'hansentree'
coef(object, ...)
```

Arguments

`object`	fitted model object.
`...`	additional arguments, ignored.

Value

`coef` applied to a `browntree` object extracts a list with three elements:

`sigma` the coefficients of the sigma matrix.

`theta` a list of the estimated optima, one per character.

`sigma..sq.matrix` the sigma-squared matrix itself.

`coef` applied to a `hansentree` object returns a named list containing the estimated α and σ^2 matrices(given as the `alpha.matrix` and `sigma.sq.matrix` elements, respectively) but also the MLE returned by the optimizer (as `sqrt.alpha` and `sigma`, respectively). **The latter elements should not be interpreted, but can be used to restart the algorithm, etc.**

See Also

Other methods for ouch trees: `bootstrap()`, `logLik`, `ouch-package`, `paint()`, `plot()`, `print()`, `simulate()`, `summary()`, `update()`

---

glssoln *Generalized least-squares solver*

Description

Solves the generalized least squares problem.

Usage

```
glssoln(a, x, v, tol = sqrt(.Machine$double.eps))
```

Details

Given matrices a, x, v, `glssoln` computes y such that

$$(x - ay)^T v^{-1} (x - ay)$$

is minimized. This is accomplished by first computing the Choleski decomposition of v:

$$v = LL^T.$$

One then solves for y in the equation

$$L^{-1}ay = L^{-1}x.$$

This is accomplished by means of a singular-value decomposition of $L^{-1}a$.

The resulting y then satisfies

$$x = ay + e,$$

where the entries of e are the residuals.

Value

glssoln returns a list of two named components:

- coeff is y as above.
- residuals is e as above.

---

hansen *Ornstein-Uhlenbeck models of trait evolution*

Description

The function hansen fits an Ornstein-Uhlenbeck model to data. The fitting is done using optim or subplex.

Usage

```
hansen(
  data,
  tree,
  regimes,
  sqrt.alpha,
  sigma,
  fit = TRUE,
  method = c("Nelder-Mead", "subplex", "BFGS", "L-BFGS-B"),
  hessian = FALSE,
  ...
)
```

Arguments

data	Phenotypic data for extant species, i.e., species at the terminal twigs of the phylogenetic tree. This can either be a single named numeric vector, a list of nchar named vectors, or a data frame containing nchar data variables. There must be an entry per variable for every node in the tree; use NA to represent missing data. If the data are supplied as one or more named vectors, the names attributes are taken to correspond to the node names specified when the ouchtree was constructed (see ouchtree). If the data are supplied as a data-frame, the rownames serve that purpose.
tree	A phylogenetic tree, specified as an ouchtree object.
regimes	A vector of codes, one for each node in the tree, specifying the selective regimes hypothesized to have been operative. Corresponding to each node, enter the code of the regime hypothesized for the branch segment terminating in that node. For the root node, because it has no branch segment terminating on it, the regime specification is irrelevant. If there are nchar quantitative characters, then one can specify a single set of regimes for all characters or a list of nchar regime specifications, one for each character.

`sqrt.alpha, sigma`
: These are used to initialize the optimization algorithm. The selection strength matrix α and the random drift variance-covariance matrix σ^2 are parameterized by their matrix square roots. Specifically, these initial guesses are each packed into lower-triangular matrices (column by column). The product of this matrix with its transpose is the α or σ^2 matrix. See Details for more information.

`fit`
: If `fit=TRUE`, then the likelihood will be maximized. If `fit=FALSE`, the likelihood will be evaluated at the specified values of `sqrt.alpha` and `sigma`; the optima `theta` will be returned as well.

`method`
: The method to be used by the optimization algorithm. See `subplex::subplex` and `stats::optim` for information on the available options.

`hessian`
: If `hessian=TRUE`, then the Hessian matrix will be computed by `optim`.

`...`
: Additional arguments will be passed as `control` options to `optim` or `subplex`. See `stats::optim()` and `subplex::subplex()` for information on the available options.

Details

The Hansen model for the evolution of a multivariate trait X along a lineage can be written as a stochastic differential equation (Ito diffusion)

$$dX = \alpha(\theta(t) - X(t))dt + \sigma dB(t),$$

where t is time along the lineage, $\theta(t)$ is the optimum trait value, $B(t)$ is a standard Wiener process (Brownian motion), and α and σ are matrices quantifying, respectively, the strength of selection and random drift. Without loss of generality, one can assume σ is lower-triangular. This is because only the infinitesimal variance-covariance matrix $\sigma^2 = \sigma\sigma^T$ is identifiable, and for any admissible variance-covariance matrix, we can choose σ to be lower-triangular. Moreover, if we view the basic model as describing evolution on a fitness landscape, then α will be symmetric. If we further restrict ourselves to the case of stabilizing selection, α will be positive definite as well. We make these assumptions and therefore can assume that the matrix α has a lower-triangular square root.

The `hansen` code uses unconstrained numerical optimization to maximize the likelihood. To do this, it parameterizes the α and σ^2 matrices in a special way: each matrix is parameterized by `nchar*(nchar+1)/2` parameters, where `nchar` is the number of quantitative characters. Specifically, the parameters initialized by the `sqrt.alpha` argument of `hansen` are used to fill the nonzero entries of a lower-triangular matrix (in column-major order), which is then multiplied by its transpose to give the selection-strength matrix. The parameters specified in `sigma` fill the nonzero entries in the lower triangular σ matrix. When `hansen` is executed, the numerical optimizer maximizes the likelihood over these parameters.

Value

`hansen` returns an object of class `hansentree`.

Author(s)

Aaron A. King

References

Butler, M.A. and A.A. King. 2004. Phylogenetic comparative analysis: a modeling approach for adaptive evolution. Am. Nat. 164:683–695.

Cressler, C. E., Butler, M. A., and King, A. A. 2015. Detecting adaptive evolution in phylogenetic comparative analysis using the Ornstein-Uhlenbeck model. Syst. Biol., 64:953–968.

See Also

stats::optim, subplex::subplex, bimac, anolis.ssd

Other phylogenetic comparative models: brown(), ouch-package, ouchtree, paint()

Examples

```
## Analysis of sexual size dimorphism data
tree <- with(anolis.ssd,ouchtree(node,ancestor,time/max(time),species))
plot(tree,node.names=TRUE)

h1 <- brown(anolis.ssd['log.SSD'],tree)
h1
plot(h1)

h2 <- hansen(anolis.ssd['log.SSD'],tree,anolis.ssd['OU.1'],sqrt.alpha=1,sigma=1)
h2
plot(h2)

h3 <- hansen(anolis.ssd['log.SSD'],tree,anolis.ssd['OU.7'],sqrt.alpha=1,sigma=1)
h3
plot(h3)
### Darwin's finches
library(geiger)
data(geospiza)

### check the correspondence between data and tree tips:
print(nc <- with(geospiza,name.check(geospiza.tree,geospiza.data)))
### looks like one of the terminal twigs has no data associated
### drop that tip:
tree <- with(geospiza,drop.tip(geospiza.tree,nc$tree_not_data))
dat <- as.data.frame(geospiza$dat)

### make an ouchtree out of the phy-format tree
ot <- ape2ouch(tree)

### merge data with tree info
otd <- as(ot,"data.frame")
### in these data, it so happens that the rownames correspond to node names
### we will exploit this correspondence in the 'merge' operation:
dat$labels <- rownames(dat)
otd <- merge(otd,dat,by="labels",all=TRUE)
rownames(otd) <- otd$nodes
print(otd)
### this data-frame now contains the data as well as the tree geometry
```

```
### now remake the ouch tree
ot <- with(otd,ouchtree(nodes=nodes,ancestors=ancestors,times=times,labels=labels))

b1 <- brown(tree=ot,data=otd[c("tarsusL","beakD")])
summary(b1)

### evaluate an OU model with a single, global selective regime
otd$regimes <- as.factor("global")
h1 <- hansen(
  tree=ot,
  data=otd[c("tarsusL","beakD")],
  regimes=otd["regimes"],
  sqrt.alpha=c(1,0,1),
  sigma=c(1,0,1),
  maxit=10000
)
summary(h1)
```

`logLik` *Log likelihood of a fitted model*

Description

`logLik` extracts the log likelihood from a fitted model object.

Usage

```
## S4 method for signature 'browntree'
logLik(object)

## S4 method for signature 'hansentree'
logLik(object)
```

Arguments

`object`	any object from which a log-likelihood value, or a contribution to a log-likelihood value, can be extracted.

Value

`logLik` returns a numeric value.

See Also

Other methods for ouch trees: `bootstrap()`, `coef()`, `ouch-package`, `paint()`, `plot()`, `print()`, `simulate()`, `summary()`, `update()`

ouchtree *Phylogenetic tree object in* **ouch** *format*

Description

ouchtree constructs a representation of a phylogenetic tree.

ape2ouch translates **ape**'s phylo representation of a phylogenetic tree into **ouch**'s ouchtree representation. Optionally, the user can adjust the branch lengths while preserving the topology.

Usage

```
ouchtree(nodes, ancestors, times, labels = as.character(nodes))

ape2ouch(tree, scale = TRUE, branch.lengths = tree$edge.length)
```

Arguments

nodes	A character vector giving the name of each node. These are used internally and must be unique.
ancestors	Specification of the topology of the phylogenetic tree. This is in the form of a character vector specifying the name (as given in the nodes argument) of the immediate ancestor of each node. In particular, the i-th name is that of the ancestor of the i-th node. The root node is distinguished by having no ancestor (i.e., NA).
times	A vector of nonnegative numbers, one per node in the tree, specifying the time at which each node is located. Time should be increasing from the root node to the terminal twigs.
labels	Optional vector of node labels. These will be used in plots to label nodes. It is not necessary that these be unique.
tree	a tree of class ape::phylo.
scale	optional. If scale=TRUE, the tree's depth will be scaled to 1. If scale is a number, then the branch lengths will be scaled by this number.
branch.lengths	optional vector of branch lengths.

Details

ouchtree() creates an ouchtree object given information on the phylogeny's topology and node times. An ouchtree object also (optionally) holds names of taxa for display purposes.

Author(s)

Aaron A. King

A. A. King, D. Ackerly

See Also

Other phylogenetic comparative models: brown(), hansen(), ouch-package, paint()

Examples

```
tree <- with(
  bimac,
  ouchtree(nodes=node,ancestors=ancestor,times=time,labels=spcode)
)
tree

plot(tree)
plot(tree, node.names=TRUE)    # display node names
```

---

paint *Painting regimes on a phylogenetic tree*

Description

Function to paint selective regimes on a phylogenetic tree.

Usage

```
paint(tree, subtree, branch, which = 1)
```

Arguments

tree	An object of class ouchtree.
subtree	An optional named vector specifying the root nodes of subtrees. Each branch that descends from this node will be painted with the specified regime.
branch	An optional named vector specifying the end nodes of branches. The unique branch that terminates at the named node will be painted with the specified regime.
which	integer; if tree is a hansentree, start not with a blank canvas but with the regime specifications tree contains for the character indicated by which.

Details

The names of subtree and branch must be the names of nodes of tree. The painting proceeds in a particular order: one can overpaint a branch. The subtrees indicated by the elements of subtree are painted first, in order. Then the branches indicated by branch are painted. If tree is of class hansentree, then paint begins with the regimes specified in the regimes slot of tree. Otherwise, paint begins with a blank canvas, i.e., a tree painted with the single regime "nonspec". Note that, if tree is a multivariate hansentree, then there are multiple regime specifications contained in tree. In this case, the argument which lets you pick which one you wish to begin with; by default, the first is used.

Value

A vector of class 'factor' with names corresponding to the nodes in `tree`, specifying selective regimes.

Author(s)

Aaron A. King

See Also

Other methods for ouch trees: `bootstrap()`, `coef()`, `logLik`, `ouch-package`, `plot()`, `print()`, `simulate()`, `summary()`, `update()`

Other phylogenetic comparative models: `brown()`, `hansen()`, `ouch-package`, `ouchtree`

Examples

```
x <- with(
  bimac,
  ouchtree(nodes=node,times=time/max(time),ancestors=ancestor,labels=species)
)

r <- paint(x,subtree=c("1"="medium","9"="large","2"="small"),
  branch=c("38"="large","2"="medium"))
plot(x,regimes=r,node.names=TRUE)

## compare to bimac['OU.LP']
h5 <- hansen(data=log(bimac['size']),tree=x,regimes=bimac['OU.LP'],
  sqrt.alpha=1,sigma=1,reltol=1e-5)
r <- paint(h5,branch=c("18"="large"),subtree=c("9"="small"))
plot(x,regimes=r,node.names=TRUE)
```

`plot` **ouch** *plotting functions*

Description

Plot phylogenetic trees, with or without regime paintings.

Usage

```
## S4 method for signature 'ouchtree'
plot(
  x,
  ...,
  regimes = NULL,
  ladderize = TRUE,
  node.names = FALSE,
  legend = !is.null(regimes),
```

```
  labels,
  frame.plot = FALSE,
  palette = rainbow,
  margin = 0.1,
  text_opts = list(),
  legend_opts = list()
)

## S4 method for signature 'hansentree'
plot(x, ..., regimes, legend = TRUE)
```

Arguments

x	object to plot.
...	additional arguments, passed to plot.
regimes	factor or character; a vector of regime paintings.
ladderize	logical; should the tree be ladderized?
node.names	logical; should node names be displayed?
legend	logical; display a legend?
labels	character; taxon labels.
frame.plot	a logical indicating whether a box should be drawn around the plot.
palette	function or character; specifies the colors to be used for the several regimes on the tree. Specified as a function, when given an integer, n, the function should create a vector of n colors. See, for example rainbow. One can also specify the n colors as a vector of color codes. There must be at least as many colors as levels in the regimes.
margin	numeric; width of the right margin (as a fraction of the plot width). Adjust this if labels are clipped (see Examples below). One can also adjust the width of the left margin (for example to aid in the formatting of the figure legend). To do this, furnish margin=c(L,R), where L and R are the widths of the right and left margins, respectively, as fractions of the plot width. Obviously, in this case, we must have L+R<1.
text_opts	options for the labels; passed to text.
legend_opts	options for the the legend; passed to legend.

See Also

Other methods for ouch trees: bootstrap(), coef(), logLik, ouch-package, paint(), print(), simulate(), summary(), update()

Examples

```
tree <- with(
  bimac,
  ouchtree(nodes=node,ancestors=ancestor,times=time,labels=spcode)
)
```

```
plot(tree)
plot(tree, node.names=TRUE)    # display node names

## When taxon names are long, they are cut off when the
## default settings are used.  For example:
tree2 <- with(
  bimac,
  ouchtree(nodes=node,ancestors=ancestor,times=time,
    labels=ifelse(is.na(species),NA,paste(species,island,sep=", "))
  )
)

plot(tree2) # long species names are cut off
## This is fixed by increasing right margin and font size:
plot(tree2,margin=0.35,text_opts=list(cex=0.7))
```

---

print *Print and show methods*

Description

Print and show methods

Usage

```
## S4 method for signature 'ouchtree'
print(x, ...)

## S4 method for signature 'ouchtree'
show(object)

## S4 method for signature 'browntree'
show(object)

## S4 method for signature 'browntree'
print(x, ...)

## S4 method for signature 'hansentree'
print(x, ...)

## S4 method for signature 'hansentree'
show(object)
```

Arguments

...	additional arguments, ignored.
object, x	object to display.

Details

print displays the tree as a table, along with the coefficients of the fitted model and diagnostic information. show provides a similar display.

Value

print returns x, invisibly.

show returns NULL, invisibly.

See Also

Other methods for ouch trees: bootstrap(), coef(), logLik, ouch-package, paint(), plot(), simulate(), summary(), update()

---

simulate *Simulations of a phylogenetic trait model*

Description

simulate generates random deviates from a fitted model.

Usage

```
## S4 method for signature 'browntree'
simulate(object, nsim = 1, seed = NULL, ...)

## S4 method for signature 'hansentree'
simulate(object, nsim = 1, seed = NULL, ...)
```

Arguments

object	fitted model object
nsim	integer; number of independent simulations.
seed	integer; if non-NULL, the RNG will be initialized with this seed for the simulations. The RNG will be reset to its pre-existing state when simulate returns.
...	additional arguments, ignored.

Value

simulate returns a list of data-frames, each comparable to the original data.

See Also

Other methods for ouch trees: bootstrap(), coef(), logLik, ouch-package, paint(), plot(), print(), summary(), update()

summary — **ouch** *summary methods*

Description

ouch summary methods

Usage

```
## S4 method for signature 'browntree'
summary(object, ...)

## S4 method for signature 'hansentree'
summary(object, ...)
```

Arguments

object	fitted model object.
...	additional arguments, ignored.

Value

summary applied to a browntree object returns information about the fitted model, including parameter estimates and quantities describing the goodness of fit.

summary applied to a hansentree method displays the estimated α and σ^2 matrices as well as various quantities describing the goodness of model fit.

See Also

Other methods for ouch trees: bootstrap(), coef(), logLik, ouch-package, paint(), plot(), print(), simulate(), update()

update — *Update and refit an* **ouch** *model*

Description

update will update a model and re-fit. This allows one to change the data and/or parameters.

Usage

```
## S4 method for signature 'browntree'
update(object, data, ...)

## S4 method for signature 'hansentree'
update(object, data, regimes, sqrt.alpha, sigma, ...)
```

Arguments

object	fitted model object.
data	data that replace those used in the original fit.
...	Additional arguments replace the corresponding arguments in the original call.
regimes	A vector of codes, one for each node in the tree, specifying the selective regimes hypothesized to have been operative. Corresponding to each node, enter the code of the regime hypothesized for the branch segment terminating in that node. For the root node, because it has no branch segment terminating on it, the regime specification is irrelevant. If there are nchar quantitative characters, then one can specify a single set of regimes for all characters or a list of nchar regime specifications, one for each character.
sqrt.alpha	These are used to initialize the optimization algorithm. The selection strength matrix α and the random drift variance-covariance matrix σ^2 are parameterized by their matrix square roots. Specifically, these initial guesses are each packed into lower-triangular matrices (column by column). The product of this matrix with its transpose is the α or σ^2 matrix. See Details for more information.
sigma	These are used to initialize the optimization algorithm. The selection strength matrix α and the random drift variance-covariance matrix σ^2 are parameterized by their matrix square roots. Specifically, these initial guesses are each packed into lower-triangular matrices (column by column). The product of this matrix with its transpose is the α or σ^2 matrix. See Details for more information.

Value

update returns a new fitted-model object of the same class as object.

See Also

Other methods for ouch trees: bootstrap(), coef(), logLik, ouch-package, paint(), plot(), print(), simulate(), summary()

Index