

Package ‘surface’

May 9, 2026

Type Package

Title Fitting Hansen Models to Investigate Convergent Evolution

Version 0.6

Date 2024-08-19

Depends R (>= 2.6), ape, ouch, MASS, geiger

Suggests igraph

Imports methods, phytools

Description This data-driven phylogenetic comparative method fits stabilizing selection models to continuous trait data, building on the 'ouch' methodology of Butler and King (2004) <[doi:10.1086/426002](https://doi.org/10.1086/426002)>. The main functions fit a series of Hansen models using stepwise AIC, then identify cases of convergent evolution where multiple lineages have shifted to the same adaptive peak. For more information see Ingram and Mahler (2013) <[doi:10.1111/2041-210X.12034](https://doi.org/10.1111/2041-210X.12034)>.

License GPL (>= 2)

URL <https://www.otago.ac.nz/ecoevotago/research/code/index.html>

NeedsCompilation no

Author Travis Ingram [aut, cre]

Maintainer Travis Ingram <travis.ingram@otago.ac.nz>

Repository CRAN

Date/Publication 2024-08-19 12:00:02 UTC

Contents

surface-package	2
convertTreeData	3
getAIC	4
getBranchTimes	5
ouchDescendants	6
propRegMatch	7
repaint	8
runSurface	9

startingModel	11
surfaceAICPlot	12
surfaceBackward	14
surfaceDemo	16
surfaceForward	17
surfaceSimulate	19
surfaceSummary	22
surfaceTreePlot	23

Index	26
--------------	-----------

surface-package	<i>Fitting Hansen Models to Investigate Convergent Evolution</i>
-----------------	--

Description

surface provides a wrapper to the ouch package, fitting a series of Hansen multiple-peak stabilizing selection models using stepwise AIC, and identifying cases of convergence where independent lineages discovered the same adaptive peak

Details

Package: surface
 Type: Package
 Version: 0.5
 Date: 2020-11-10
 License: GPL (>=2)

surface uses the Hansen model of stabilizing selection around multiple adaptive peaks to infer a macroevolutionary adaptive landscape using only trait data and a phylogenetic tree. The most important functions are surfaceForward and surfaceBackward, which carry out the two stepwise phases of the method, and runSurface, a wrapper function that carries out both phases. Results can be displayed using surfaceSummary, and visualized using surfaceTreePlot, surfaceTraitPlot, and surfaceAICPlot. Hypothesis tests, such as whether the extent of convergence exceeds the expectation under a model without true convergence, can be done with the assistance of surfaceSimulate. The vignette ‘surface_tutorial’ demonstrates the use of the various functions included in the package

Author(s)

Travis Ingram <travis.ingram@otago.ac.nz>

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

See Also

[runSurface](#), [surfaceForward](#), [surfaceBackward](#), [surfaceSimulate](#), [surfaceSummary](#), [surfaceTreePlot](#), [surfaceTraitPlot](#), [surfaceAICPlot](#)

Examples

```
#executable R code and demonstrations of the key functions can be found in the tutorial
#vignette("surface_tutorial", package = "surface")
```

convertTreeData

Utilities for Formatting Objects for SURFACE Analysis

Description

convertTreeData converts a phylo-formatted tree and a data frame into formats ready to be analyzed with the ouch functions called by surface. convertBack converts an ouchtree to a data frame including regime information, and is called internally by surfaceTreePlot. nameNodes adds unique node labels to a phylo tree to ensure reliable conversion between formats

Usage

```
convertTreeData(tree, dat)
convertBack(tree, otree, regshifts)
nameNodes(tree)
```

Arguments

tree	Phylogenetic tree in phylo format: to ensure reliable conversion, should have node labels (e.g. using nameNodes)
dat	Data frame with row names corresponding to the taxa in tree and columns consisting of one or more trait measurements
otree	Phylogenetic tree in ouchtree format
regshifts	Named character vector of regime shifts indicating branches containing shifts (numbered corresponding to otree@nodes) and regime identities (usually lowercase letters)

Value

convertTreeData returns a list with components otree (a phylogenetic tree in ouchtree format) and odata (a data frame containing trait data, with rownames corresponding to otree@labels). convertBack returns a data frame containing original phenotypic data as well as regime assignments of tip taxa. nameNodes returns the input tree, with arbitrary node names added (zzz1, zzz2, etc) to ensure reliable conversion between formats

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

See Also

[surfaceBackward](#), [surfaceForward](#), [surfaceTreePlot](#)

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
```

getAIC

Akaike's Information Criterion for SURFACE Models

Description

Calculates AICc for a Hansen model using combined likelihoods across multiple traits

Usage

```
getAIC(L, np, n, AICc = TRUE)
npSurface(fit)
```

Arguments

fit	Fitted Hansen model object (the list returned by one iteration of <code>surfaceForward</code> or <code>surfaceBackward</code>)
L	Log-likelihood of the model
np	Number of parameters in the model
n	Sample size (total number of trait values)
AICc	A logical indicating whether to use small-sample size corrected AIC; defaults to TRUE, and is currently set to TRUE during all calls within the surface functions

Details

The number of parameters is calculated as $p = k + (k' + 2) m$, where k is the number of regime shifts, k' is the number of distinct regimes, and m is the number of traits. Note that this differs from many applications of Hansen models, in that SURFACE counts regime shifts as "parameters", modeling the complexity of both the adaptive landscape (number of regimes) and the evolutionary history of the clade (number of regime shifts). For AICc, the sample size is taken to be the total number of trait values mn , where n is the number of taxa

Value

npSurface returns an integer number of parameters. getAIC returns a numeric AIC or AICc value

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
startmod<-startingModel(otree, odata, shifts = c("6"="b"))
np<-as.numeric(npSurface(startmod[[1]]))
LnL<-sum(sapply(startmod[[1]]$fit, function(x) summary(x)$loglik))
getAIC(LnL,np,n=ncol(dat)*nrow(dat),AICc=TRUE)
```

getBranchTimes

Extract Branching Times from an ouch Tree

Description

Extracts the time from root of each node in an ouchtree or hansentree formatted phylogenetic tree; used to compute the timing of regime shifts in a Hansen model

Usage

```
getBranchTimes(h)
```

Arguments

h Fitted ouchtree or hansentree object

Value

A vector of branching times

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]
getBranchTimes(otree)
```

ouchDescendants

Obtain Descendants from an ouch Tree

Description

Identifies the nodes and tip taxa descended from a given ancestor in an ouchtree or hansentree object. Used to test whether two ‘convergent’ regimes are actually nested when randomly placing regime shifts in a Hansen model in the function surfaceSimulate

Usage

```
ouchDescendants(node, otree)
```

Arguments

node	Which node in the ouchtree object to identify the descendants of
otree	An ouchtree object

Value

A vector of integers corresponding to the descendents (integers match the @nodes element of the ouchtree)

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Examples

```

data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]
ouchDescendants(6, otree)

```

propRegMatch

Similarity of Two Hansen Models

Description

Calculates the pairwise matching between two alternate paintings of the same phylogenetic tree. This is done by creating a half-matrix for each hansen tree object indicating whether each pairwise comparison of tip species or branches shows they are in the same regime (coded '1') or different regimes (coded '0'). The 'proportion matching' value returned is the proportion of elements of the two matrices that are equal; a measure of correspondence between two Hansen models (one of which may be the 'true' model if data are simulated)

Usage

```
propRegMatch(fit1, fit2, internal = FALSE)
```

Arguments

fit1	First fitted Hansen model; can be the \$fit component of the list returned by either one iteration of an analysis with surfaceForward or surfaceBackward, or the list returned by surfaceSimulate
fit2	Second fitted Hansen model; see fit1
internal	A logical indicating whether internal branches should be included in the calculation of matching in addition to tip taxa; this is only possible if the two trees have identical topology; defaults to FALSE

Value

A single value quantifying the proportion of pairwise regime comparisons that are the same between the two models

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

See Also

[surfaceForward](#), [surfaceBackward](#), [surfaceSimulate](#)

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
startmod<-startingModel(otree, odata, shifts = c("6"="b"))
startmod2<-startingModel(otree, odata, shifts = c("6"="b","17"="c"))
propRegMatch(startmod[[1]]$fit, startmod2[[1]]$fit)
```

 repaint

Paint the Branches of a Tree

Description

A wrapper to the `paint` function in `ouch` to ensure that regime paintings are automatically formatted for SURFACE analysis (painting the stem branch of a clade and ensuring that the root is assigned a regime)

Usage

```
repaint(otree, regshifts, stem = TRUE)
```

Arguments

<code>otree</code>	Phylogenetic tree in <code>ouchtree</code> format
<code>regshifts</code>	Named character vector of regime shifts
<code>stem</code>	A logical indicating whether the painting of a clade should include the stem branch; defaults to <code>TRUE</code> , and is set to <code>TRUE</code> during all calls within the surface functions

Value

A named character vector of regime assignments for each branch, as returned by `paint`

Author(s)

Travis Ingram

References

Butler, M.A. & King, A.A. (2004) Phylogenetic comparative analysis: a modeling approach for adaptive evolution. *American Naturalist* 164: 683-695.

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]
repaint(otree, regshifts = c(c("1"="a","6"="b","17"="c")))
```

runSurface

Run All Steps of a SURFACE Analysis

Description

Carries out both the forward and backward phases of SURFACE's stepwise AIC routine, with sensible default behaviors.

Usage

```
runSurface(tree, dat, exclude = 0, aic_threshold = 0, max_steps = NULL,
  verbose = FALSE, plotaic = FALSE, error_skip = FALSE, only_best = FALSE,
  sample_shifts=FALSE, sample_threshold = 2)
```

Arguments

tree	Phylogenetic tree in phylo format
dat	Data frame with taxa names as rownames matching the tip labels of tree, and one or more columns of trait data
exclude	Optionally, the proportion of the worst models (AICc scores for each shift point) to exclude in the current round of the forward phase (defaults to zero)
aic_threshold	Change in AICc needed to accept a candidate model as a sufficient improvement over the previous iteration of SURFACE. Defaults to zero, meaning any improvement in the AICc will be accepted; more stringent thresholds are specified using <i>negative</i> values of aic_threshold
max_steps	Maximum number of steps to allow to allow each phase to carry out (assuming the model improvement continues to exceed aic_threshold)
verbose	A logical indicating whether to print progress (defaults to FALSE)

plotaic	A logical indicating whether to plot AICc values of all candidate models at each step (defaults to FALSE)
only_best	A logical indicating whether to only allow one pair of regimes to be collapsed at each iteration; if FALSE, igrph functions are used to identify pairs of regimes that can be collapsed to improve the model without any inconsistencies (defaults to FALSE)
error_skip	A logical indicating whether to skip over any candidate model that produces an error message during likelihood optimization (this is rare, but can cause an entire analysis to abort; defaults to FALSE)
sample_shifts	A logical indicating whether to randomly sample from among the best models at each step (those within sample_threshold of the best AICc), rather than deterministically selecting the best candidate model (defaults to FALSE)
sample_threshold	Number of AICc units within which to sample among candidate models that are close to as good as the best model at each step (defaults to 2, but only used if sample_shifts=TRUE, and only used in the backward phase if only_best=TRUE)

Details

Carries out all steps of SURFACE, including converting data structures and running both forward and backward phases of the analysis. The default behavior should be appropriate in most circumstances, but some functionalities require using the functions `surfaceForward` and `surfaceBackward` that are called by `runSurface`

Value

A list with two elements, `fwd` and `bwd`.

<code>fwd</code>	The results of the forward phase, as returned by <code>surfaceForward</code>
<code>bwd</code>	The results of the backward phase, as returned by <code>surfaceBackward</code>

Author(s)

Travis Ingram

References

- Butler, M.A. & King, A.A. (2004) Phylogenetic comparative analysis: a modeling approach for adaptive evolution. *American Naturalist* 164: 683-695.
- Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.
- Mahler, D.L., Ingram, T., Revell, L.J. & Losos, J.B. (2013) Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science* 341: 292-295.

See Also

[surfaceBackward](#), [surfaceForward](#)

Examples

```
## Not run:
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
result<-runSurface(tree,dat)

## End(Not run)
```

startingModel

Create an Initial Model for a SURFACE Analysis

Description

Generate a model to start a SURFACE analysis, or fit specific Hansen or Brownian motion models that can be compared to the models returned by SURFACE

Usage

```
startingModel(otree, odata, shifts = NULL, brownian = FALSE)
```

Arguments

otree	Phylogenetic tree in ouchtree format
odata	Data frame with rownames corresponding to otree@labels
shifts	A named character vector of regime shifts. Names should correspond to otree@nodes, and regime assignments can be any character other than "a" (see details). Defaults to NULL, in which case a single-regime OU model is returned.
brownian	A logical indicating whether to return the fitted Brownian motion model for the data set by calling the ouch function brown and obtaining AICs by adding log-likelihoods across traits. If TRUE, overrides any specified shifts

Details

For most analysis, this function is not accessed by the user, but is called from within surfaceForward to initialize the run with a single-regime OU model. However, the user can optionally supply a starting model that imposes some regime shifts (e.g. if there is strong a priori reason to include them, or to evaluate how their inclusion changes the result of SURFACE analysis). If shifts are supplied, they are always modified so that the first element codes a basal regime 'shift' c("1"="a"). Thus, if any other element in shifts is specified as regime "a", or has name "1", an error will be returned. startingModel can also be used to obtain a fit (with AICc calculated after adding log-likelihoods across traits) for any hypothesized Hansen model or for Brownian motion (if brownian=TRUE) for comparison with models returned by SURFACE

Value

A list of length 1 containing an object with the same structure as the lists returned by each iteration of `surfaceForward` and `surfaceBackward` (containing elements `fit`, `all_aic`, `aic`, `saveshifts`, and `n_regimes`). This allows it to be supplied as argument `starting_list` in a call to `surfaceForward`.

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

See Also

[surfaceForward](#)

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
startmod<-startingModel(otree, odata, shifts = c("6"="b"))
```

surfaceAICPlot

Plot the AIC Throughout a SURFACE Analysis

Description

Plots a line graph showing how the AICc changed over the forward and backward phases of a SURFACE analysis. `surfaceAICPlot` can optionally show the change in the deviance or 'partial AICc' for each trait separately as well as for the analysis as a whole. `surfaceAICMultiPlot` plots lines from multiple runs on the same plot, allowing comparison among analyses done on alternate tree topologies or with stochasticity added using `sample_shifts`

Usage

```
surfaceAICPlot(fwd = NULL, bwd = NULL, out = NULL, summ = NULL,
traitplot = "none", cols = NULL, daic = FALSE, ...)
surfaceAICMultiPlot(fwd = NULL, bwd = NULL, out = NULL, summ = NULL,
cols = NULL, daic = FALSE, ...)
```

Arguments

fwd	List resulting from a surfaceForward run, or a list of such lists if calling surfaceAICMultiPlot
bwd	List resulting from a surfaceBackward run, or a list of such lists if calling surfaceAICMultiPlot
out	List resulting from a runSurface run, consisting of elements fwd and bwd, or a list of such lists if calling surfaceAICMultiPlot
summ	Object returned by surfaceSummary (run on the forward and backward phases of an analysis together), or a list of such objects if calling surfaceAICMultiPlot
traitplot	String indicating what values to use to draw lines corresponding to individual traits: "none", "dev" or "aic" (see details); defaults to "none"
cols	An optional character vector of colors for the AICc lines, used to color the different runs in surfaceAICMultiPlot. Only used in surfaceAICPlot if traitplot = "aic" or traitplot = "dev", in which case the colors are used for the trait lines (the overall AICc line is drawn in black)
daic	A logical indicating whether to rescale all delta-AICc (and delta-deviance) values to the value from the starting model; defaults to FALSE, but is automatically set to TRUE if traitplot = "aic" or traitplot = "dev"
...	Additional arguments to be passed to the plot or points functions

Details

If values are plotted on a trait-by-trait basis, either traitplot="dev" or traitplot="aic" can be specified. If traitplot="dev", the deviance ($-2 \cdot \log$ likelihood) at each step is shown for each trait. If traitplot="aic", a "partial AICc" at each step is shown for each of the m traits, consisting of the deviance and $1/m$ of the "penalty" part of the overall AICc, where m is the number of traits. Note that this is not a proper statistical construct, but its property of adding to give the overall AICc can be useful in visualizing the patterns among traits

Value

Plots AIC values from a SURFACE analysis on the current graphics device

Author(s)

Travis Ingram

References

- Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.
- Mahler, D.L., Ingram, T., Revell, L.J. & Losos, J.B. (2013) Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science* 341: 292-295.

See Also

[surfaceForward](#), [surfaceBackward](#), [surfaceSimulate](#), [surfaceSummary](#), [surfaceTreePlot](#), [surfaceTraitPlot](#)

Examples

```
## Not run:
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
result<-runSurface(tree,dat)
surfaceAICPlot(result$fwd,result$bwd)

## End(Not run)
```

surfaceBackward

Collapsing Convergent Regimes in a Hansen Model

Description

Carries out the backward phase of SURFACE's stepwise AIC routine. Beginning with a fitted Hansen model produced by `surfaceForward`, tests pairwise collapses of regimes and identifies collapses that improve the fit. Continues this iterative process until the model stops improving beyond the given AIC threshold

Usage

```
surfaceBackward(otree, odata, starting_model, aic_threshold = 0,
max_steps = NULL, save_steps = FALSE, filename = "temp_back_list.R",
verbose = FALSE, only_best = FALSE, plotaic = FALSE,
error_skip = FALSE, sample_shifts = FALSE, sample_threshold = 2)
collapseRegimes(otree, odata, oldshifts, oldaic, oldfit, aic_threshold = 0,
only_best = FALSE, verbose = TRUE, plotaic = TRUE, error_skip = FALSE,
sample_shifts = FALSE, sample_threshold = 2)
```

Arguments

<code>otree</code>	Phylogenetic tree in ouchtree format
<code>odata</code>	Data frame with rownames corresponding to <code>otree@labels</code>
<code>starting_model</code>	The Hansen model to attempt regime collapses on; typically the final element of a <code>surfaceForward</code> analysis
<code>aic_threshold</code>	Change in AICc needed to accept a candidate model as a sufficient improvement over the previous iteration of SURFACE. Defaults to zero, meaning any improvement in the AICc will be accepted; more stringent thresholds are specified using <i>negative</i> values of <code>aic_threshold</code>
<code>max_steps</code>	Maximum number of steps in the backward phase to carry out (assuming the model improvement continues to exceed <code>aic_threshold</code>)

save_steps	A logical indicating whether to save the current iteration of the model at each step (overwriting if necessary) to a file filename (defaults to FALSE)
filename	Name of the file to save progress to at each step, if save_steps=TRUE
verbose	A logical indicating whether to print progress (defaults to FALSE)
only_best	A logical indicating whether to only allow one pair of regimes to be collapsed at each iteration; if FALSE, igrph functions are used to identify pairs of regimes that can be collapsed to improve the model without any inconsistencies (defaults to FALSE)
plotaic	A logical indicating whether to plot AICc values of candidate models at each step (defaults to FALSE)
error_skip	A logical indicating whether to skip over any candidate model that produces an error message (this is rare, but can cause an entire analysis to abort; defaults to FALSE)
sample_shifts	A logical indicating whether to sample from among the best models at each step (those within sample_threshold of the best AICc), rather than always selecting the best candidate model (defaults to FALSE; both sample_shifts and only_best must be set to TRUE to use this option during the backward phase)
sample_threshold	Number of AICc units within which to sample among candidate models that are close to as good as the best model at each step (defaults to 2, but only used if sample_shifts=TRUE and only_best=TRUE)
oldshifts	Shifts present in the previous iteration of the Hansen model
oldaic	AICc value for the Hansen model from the previous iteration
oldfit	Previous fitted Hansen model

Details

Can be time-consuming, as the number of likelihood searches at a step is $k(k-1)/2$, where k is the number of regimes in the model.

Value

collapseRegime returns a list corresponding to one iteration of the backward phase of the SURFACE analysis; surfaceBackward returns a list of such lists consisting of each step of the stepwise process

fit	The fitted Hansen model selected for improving the AICc most over the previous iteration; consists of a single hansentree object if the number of traits $m = 1$, or a list of hansentree objects if $m > 1$
all_aic	The AICc for each model tested during the iteration (one per pair of regimes)
aic	The AICc of the current Hansen model
savedshifts	The shifts present in the current Hansen model; represented as a named character vector of regime assignments (lower-case letters), with names indicating branches containing shifts

Author(s)

Travis Ingram

References

Butler, M.A. & King, A.A. (2004) Phylogenetic comparative analysis: a modeling approach for adaptive evolution. *American Naturalist* 164: 683-695.

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Mahler, D.L., Ingram, T., Revell, L.J. & Losos, J.B. (2013) Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science* 341: 292-295.

See Also

[surfaceForward](#), [surfaceSimulate](#), [surfaceTreePlot](#), [surfaceSummary](#)

Examples

```
## Not run:
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
fwd<-surfaceForward(otree, odata, aic_threshold = 0, exclude = 0, verbose = FALSE, plotaic = FALSE)
k<-length(fwd)
bwd<-surfaceBackward(otree, odata, starting_model = fwd[[k]], aic_threshold = 0)

## End(Not run)
```

surfaceDemo

Tree and Data for Demonstrating SURFACE

Description

This simulated tree and data set can be used to demonstrate the functionality of SURFACE. The vignette ‘surface_tutorial’ demonstrates the use of the various functions included in the package using surfaceDemo

Usage

```
data(surfaceDemo)
```

Format

A list containing a tree in phylo format (surfaceDemo\$tree), and a list surfaceDemo\$sim, which contains trait data (surfaceDemo\$sim\$data) and the other features output by surfaceSimulate, including the generating Hansen model (surfaceDemo\$sim\$fit)

Source

simulated data

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
```

surfaceForward

Adding Regimes to a Hansen Model

Description

Carries out the forward phase of SURFACE's stepwise AIC routine, adding regime shifts to a Hansen model. `addRegime` performs one step of this analysis, and is called repeatedly by `surfaceForward`. At each step, the delta-AICc of each possible shift placement (i.e. branch) is calculated, and an updated Hansen model is returned with one shift added. This process is iterated until the model stops improving beyond a threshold delta-AICc

Usage

```
surfaceForward(otree, odata, starting_list=NULL, starting_shifts=NULL,
exclude=0, aic_threshold=0, max_steps=NULL, save_steps=FALSE,
filename="temp_out_list.R", verbose=FALSE, plotaic=FALSE,
error_skip=FALSE, sample_shifts=FALSE, sample_threshold=2)
addRegime(otree, odata, oldshifts, oldaic, oldfit, alloldaic=NULL,
exclude=NULL, aic_threshold=0, verbose=FALSE, plotaic=FALSE,
error_skip=FALSE, sample_shifts=FALSE, sample_threshold=2)
```

Arguments

<code>otree</code>	Phylogenetic tree in ouchtree format
<code>odata</code>	Data frame with rownames corresponding to <code>otree@labels</code>
<code>starting_list</code>	An optional list which may containing either a partially completed analysis (which can be built upon instead of starting over), or a custom starting model created with <code>startingModel</code> , which may include some pre-specified shifts
<code>starting_shifts</code>	An optional named character vector of shifts that are required to be in the Hansen model, which will be passed to <code>startingModel</code> when the initial model is built

exclude	Optionally, the proportion of the worst models (AICc scores for each shift point) to exclude in the current round (defaults to zero; values greater than 0.5 are not recommended)
aic_threshold	Change in AICc needed to accept a candidate model as a sufficient improvement over the previous iteration of SURFACE. Defaults to zero, meaning any improvement in the AICc will be accepted; more stringent thresholds are specified using *negative* values of aic_threshold
max_steps	Maximum number of regimes to allow to be added (assuming the model improvement continues to exceed aic_threshold)
save_steps	A logical indicating whether to save the current iteration of the model at each step (overwriting previous iterations) to a file filename (defaults to FALSE)
filename	Name of the file to save progress to at each step, if savesteps=TRUE
verbose	A logical indicating whether to print progress (defaults to FALSE)
plotaic	A logical indicating whether to plot AICc values of candidate models at each step (defaults to FALSE)
error_skip	A logical indicating whether to skip over any candidate model that produces an error message (this is rare, but can cause an entire analysis to abort; defaults to FALSE)
sample_shifts	A logical indicating whether to sample from among the best models at each step (those within sample_threshold of the best AICc), rather than always selecting the best candidate model (defaults to FALSE)
sample_threshold	Number of AICc units within which to sample among candidate models that are close to as good as the best model at each step (defaults to 2, but only used if sample_shifts=TRUE)
oldshifts	Any shifts present in the previous iteration of the Hansen model
oldaic	AICc value for the Hansen model from the previous iteration
oldfit	Previous fitted Hansen model
alloldaic	AICc values for each tested shift point in the previous iteration

Details

Can be time-consuming, as many likelihood searches are carried out at each iteration. Depending on the number of traits and taxa and the number of regimes that are fitted, surfaceForward can take anywhere from minutes to many hours (only tree sizes up to 128 taxa have been tested). Options to manage computation time include adding regimes one at a time with addRegime or using max_steps to perform the analysis several iterations at a time

Value

addRegime returns a list describing one iteration of the forward phase of the SURFACE analysis; surfaceForward returns a list of such lists consisting of each step of the stepwise process

fit The fitted Hansen model selected for improving the AICc most over the previous iteration; consists of a single hansentree object if the number of traits $m = 1$, or a list of hansentree objects if $m > 1$

all_aic	The AICc for each model tested during the iteration (numbered by branch)
aic	The AICc of the current Hansen model
savedshifts	The shifts present in the current Hansen model; represented as a named character vector of regime shifts (lower-case letters), with names indicating branches containing shifts
n_regimes	A two-element vector of the number of regime shifts and the number of distinct regimes in the current model

Author(s)

Travis Ingram

References

Butler, M.A. & King, A.A. (2004) Phylogenetic comparative analysis: a modeling approach for adaptive evolution. *American Naturalist* 164: 683-695.

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Mahler, D.L., Ingram, T., Revell, L.J. & Losos, J.B. (2013) Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science* 341: 292-295.

See Also

[surfaceBackward](#), [surfaceSimulate](#), [surfaceTreePlot](#), [surfaceSummary](#), [convertTreeData](#), [startingModel](#)

Examples

```
## Not run:
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
fwd<-surfaceForward(otree, odata, aic_threshold = 0, exclude = 0)

## End(Not run)
```

surfaceSimulate

Simulate Data for SURFACE

Description

Provides several ways to simulate data sets on phylogenetic trees in conjunction with SURFACE analyses. Can simulate under simple models without regime shifts, under a Hansen model with sampled shift locations, or under a fitted Hansen model (optionally with resampled optima)

Usage

```
surfaceSimulate(phy, type = "BM", param = 0, n_traits = NULL, dat = NULL,
vcv = NULL, hansenfit = NULL, shifts = NULL, n_shifts = NULL,
n_conv_shifts = NULL, n_regimes = NULL, n_per_regime = NULL,
no_nested = TRUE, optima = NULL, sample_optima = TRUE,
optima_distrib = NULL, optima_type = "rnorm", sigma_squared = NULL,
alpha = NULL, pshift_timefactor = NULL)
```

Arguments

phy	A phylogenetic tree in phylo format on which to simulate data
type	Type of simulation desired - options are "BM", "hansen-fit", and "hansen-paint" (see Details)
param	If type="BM", an optional parameter to rescale the tree (see Details)
n_traits	Number of traits (if not provided will be determined from other inputs or default to 1)
dat	Optional data frame of original trait data (function will use this to extract features of the data set)
vcv	Optional evolutionary variance-covariance matrix
hansenfit	A fitted Hansen model (or a list of such if multiple traits) (if type = "hansen-fit")
shifts	A vector of regime shifts, named for the branches they are to be placed on in the Hansen model to be simulated under (if type = "hansen-paint"). If specified, n_shifts, n_conv_shifts, n_regimes and n_per_regime are all ignored
n_shifts	Number of shifts to add to the Hansen model (if type = "hansen-paint")
n_conv_shifts	Number of convergent shifts to add to the Hansen model (if type = "hansen-paint"). Either n_conv_shifts or n_regimes can be specified along with n_shifts, but not both
n_regimes	Number of regimes to add to the Hansen model (if type = "hansen-paint"). Either n_conv_shifts or n_regimes can be specified along with n_shifts, but not both
n_per_regime	Integer vector of the number of shifts to each regime in the model (if type = "hansen-paint"). If specified, the vector length determines n_regimes, and the sum of the values determines n_shifts, and the number of entries >1 determines n_conv_shifts
no_nested	A logical indicating whether to ensure that a pair of 'convergent' regimes is not in fact two nested clades (if type = "hansen-paint"; defaults to TRUE)
optima	Optional matrix of optima
sample_optima	A logical indicating whether to replace the optima in the fitted model with new values from a distribution based on the inferred optima (if type = "hansen-fit"; defaults to TRUE)
optima_distrib	Optional matrix of optima distribution for each trait (see optima_type). Each column is a two-element vector c(A, B) for the trait.

optima_type	How to sample optima based on optima_distrib. Can be one of "rnorm" (default; distribution is normal with mean=A, sd=B), "runif" (distribution is uniform with center=A, width=B), or "even" (optima are evenly spaced with spacing=B, then randomized)
sigma_squared	Scalar or vector of Brownian rate parameters to use in simulations
alpha	Scalar or vector of OU attraction parameter values to use in simulations
pshift_timefactor	Factor by which to bias sampling of branches to place regimes on to be earlier (if <1) or later (if >1) in the tree. The sampling probability will be pshift_timefactor times higher at the tips than at the root

Details

Type of simulation may be "BM", "hansen-fit", or "hansen-paint".

If type = "BM", simulation uses the `sim.char` function in `geiger`, with Brownian rate `sigma_squared`. If type = "BM", param values other than 0 will transform the tree based on the Early Burst (param < 0) or single-peak Ornstein-Uhlenbeck (param > 0) model before simulating, causing trait disparity to be concentrated earlier or later in the tree, respectively

If type = "hansen-fit", an existing `hansentree` object is used as the basis of simulation using `ouch` functions, optionally with new parameter values

If type = "hansen-paint", a new `hansentree` object is produced for simulation using `ouch` functions, with specified parameter values and numbers of regimes and/or regime shifts

Value

A list with the following components (most are NULL if type = "BM"):

data	Simulated trait data in a data frame
optima	Matrix of optima for each regime for each trait in the generating model
savedshifts	Shift locations in the generating Hansen model
regimes	Regime assignments of tip taxa
shifftimes	Timing of each shift in the Hansen model (measured from the root of the tree)
fit	Generating Hansen model used in the simulation

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

See Also

[surfaceForward](#), [surfaceBackward](#), [surfaceTreePlot](#), [surfaceTraitPlot](#)

Examples

```

data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
sim<-surfaceSimulate(otree,type="hansen-paint",dat=dat,shifts=c(c("1"="a","6"="b","17"="c")))

```

surfaceSummary

Summarize SURFACE Output

Description

Extracts the most important results from the output of the forward, backward, or both phases of a SURFACE analysis

Usage

```
surfaceSummary(fwd = NULL, bwd = NULL)
```

Arguments

fwd	A list returned by surfaceForward (which may be stored as the \$fwd component of the list returned by runSurface)
bwd	A list returned by surfaceBackward (which may be stored as the \$bwd component of the list returned by runSurface)

Details

If both fwd and bwd are provided, both phases of the analysis will be summarized together

Value

A list with the following components:

n_steps	number of iterations in the stepwise analysis
lnls	matrix of traits-by-iterations, giving the log-likelihood for each trait at each iteration of the analysis
n_regimes_seq	matrix of the summaries of regime structure at each iteration of the model
aics	vector giving the AICc value at each step
shifts	shifts present in the final fitted Hansen model
n_regimes	summary of regime structure of the final fitted Hansen model (see note below)
alpha	estimate of alpha for each trait in the final model
sigma_squared	estimate of sigma_squared for each trait in the final model
theta	matrix of estimated optima (one per regime per trait) in the final model

Note

The elements `n_regimes_seq` and `n_regimes` contain measures of the regime structure in a SURFACE analysis (for each iteration, and in the final model, respectively). The measures returned are: `k` (the number of regime shifts, counting the basal regime as 1), `kprime`, (the number of regimes, some of which may be reached by multiple shifts), `deltak` (`k-kprime`, a measure of convergence), `c` (the number of shifts to convergent regimes, another measure of convergence), `kprime_conv` (the number of convergent regimes shifted to multiple times), and `kprime_nonconv` (the number of nonconvergent regimes only shifted to once)

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

See Also

[surfaceForward](#), [surfaceBackward](#)

Examples

```
## Not run:
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
result<-runSurface(tree,dat)
surfaceSummary(result$fwd,result$bwd)

## End(Not run)
```

surfaceTreePlot

Visualize Results of a SURFACE Analysis

Description

Plotting functions to visualize the results of a SURFACE analysis, with colors depicting regime structure: `surfaceTreePlot` produces a customized `plot.phylo` figure, and `surfaceTraitPlot` produces a scatterplot of trait values and optima

Usage

```

surfaceTreePlot(tree, hansenfit, cols = NULL, convcol = TRUE, labelshifts = FALSE, ...)
surfaceTraitPlot(dat, hansenfit, whattraits = c(1, 2), cols = NULL,
convcol = TRUE, pchs = c(21, 21), cex.opt = 2.5, optellipses = FALSE,
ellipsescale = 1, flatten1D = FALSE, add = FALSE, ypos = 0,
plotoptima = TRUE, plottraits = TRUE, y.lim = NULL, x.lim = NULL,
y.lab = NULL, x.lab = NULL, ...)

```

Arguments

tree	Phylogenetic tree in phylo format
dat	Trait data formatted as a data frame with named rows and at least two columns
hansenfit	An object containing the fitted Hansen model to use in plotting, with elements fit and savedshifts. This may be the list produced by any one iteration of surfaceForward or surfaceBackward, or the list produced by surfaceSimulate
whattraits	A two-element integer (or a single integer; see Details) indicating which traits to use for the (x,y) axes of a trait plot (defaults to c(1, 2))
cols	An optional character vector of colors for painting branches in surfaceTreePlot or coloring symbols in surfaceTraitPlot. One color should be provided per regime in hansenfit; if cols=NULL the function will attempt an appropriate default
convcol	A logical indicating whether to select separate colors for convergent (colorful) and non-convergent (greyscale) regimes (defaults to TRUE)
labelshifts	A logical indicating whether to add integer labels to branches in the tree to show the order in which regime shifts were added in the forward phase (defaults to FALSE)
pchs	Vector with two integers representing the plotting characters to use for trait values and optima, respectively, in surfaceTraitPlot; both default to 21 (filled circles)
cex.opt	Character expansion for symbols representing the optima in surfaceTraitPlot; defaults to 2.5 (symbols representing data points can be specified with cex)
optellipses	A logical indicating whether to draw ellipses based on the fitted OU model instead of denoting optimum positions with pchs and cex.opt. The ellipses are drawn as the optima +/- the standard deviation of the stationary distribution of the inferred OU process: $\sigma^2 / (2 * \alpha)$, multiplied by ellipsescale
ellipsescale	A scalar or vector indicating how many standard deviations to draw ellipses above and below the optima; if a vector, concentric ellipses of various sizes will be drawn; defaults to 1
flatten1D	A logical indicating whether all regimes should be placed on a single line when surfaceTraitPlot is called for a single trait; defaults to FALSE
add	A logical indicating whether to add a new element to an existing surfaceTraitPlot graph instead of creating a new one; defaults to FALSE
ypos	Position on the y axis to place the traits and optima on; only applies if a single trait is used and flatten1D = TRUE

plotoptima	A logical indicating whether the optima should be displayed in surfaceTraitPlot; defaults to TRUE
plottraits	A logical indicating whether the trait values should be displayed in surfaceTraitPlot; defaults to TRUE
y.lim	Lower and upper limits for the y-axis; by default will be calculated to fit all points and ellipses fit in the frame
x.lim	Lower and upper limits for the x-axis; by default will be calculated to fit all points and ellipses fit in the frame
y.lab	y-axis label; defaults to the column name in the data frame
x.lab	x-axis label; defaults to the column name in the data frame
...	Additional arguments to be passed to the plot or points functions

Details

For trait plots using the option `optellipses=TRUE`, note that in some cases (e.g. if `alpha` is very small) the ellipses will not convey useful information. If trait data are unidimensional, or if `whattraits` is provided as a single integer, data will be plotted on the x-axis and the y-axis will separate different regimes (and ellipse width in the y-dimension will not be meaningful)

Value

Creates one tree or trait plot on the current graphics device

Author(s)

Travis Ingram

References

Ingram, T. & Mahler, D.L. (2013) SURFACE: detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise AIC. *Methods in Ecology and Evolution* 4: 416-425.

Mahler, D.L., Ingram, T., Revell, L.J. & Losos, J.B. (2013) Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science* 341: 292-295.

See Also

[surfaceForward](#), [surfaceBackward](#), [surfaceSimulate](#), [surfaceSummary](#), [surfaceAICPlot](#)

Examples

```
data(surfaceDemo)
tree<-surfaceDemo$tree
dat<-surfaceDemo$sim$dat
olist<-convertTreeData(tree,dat)
otree<-olist[[1]]; odata<-olist[[2]]
startmod<-startingModel(otree, odata, shifts = c("6"="b"))
surfaceTreePlot(tree,startmod[[1]],labelshifts=TRUE,cols=c("black","red"))
surfaceTraitPlot(dat,startmod[[1]],whattraits=c(1,2),cols=c("black","red"))
```

Index

- * **arith**
 - getAIC, 4
 - propRegMatch, 7
 - * **datagen**
 - surfaceSimulate, 19
 - * **datasets**
 - surfaceDemo, 16
 - * **hplot**
 - surfaceAICPlot, 12
 - surfaceTreePlot, 23
 - * **iteration**
 - runSurface, 9
 - surfaceBackward, 14
 - surfaceForward, 17
 - * **manip**
 - convertTreeData, 3
 - getBranchTimes, 5
 - ouchDescendants, 6
 - repaint, 8
 - startingModel, 11
 - surfaceSummary, 22
 - * **models**
 - runSurface, 9
 - surfaceBackward, 14
 - surfaceForward, 17
 - * **package**
 - surface-package, 2
- addRegime (surfaceForward), 17
- collapseRegimes (surfaceBackward), 14
- convertBack (convertTreeData), 3
- convertTreeData, 3, 19
- getAIC, 4
- getBranchTimes, 5
- nameNodes (convertTreeData), 3
- npSurface (getAIC), 4
- ouchDescendants, 6
- propRegMatch, 7
- repaint, 8
- runSurface, 3, 9
- startingModel, 11, 19
- surface (surface-package), 2
- surface-package, 2
- surfaceAICMultiPlot (surfaceAICPlot), 12
- surfaceAICPlot, 3, 12, 25
- surfaceBackward, 3, 4, 8, 10, 14, 14, 19, 21, 23, 25
- surfaceDemo, 16
- surfaceForward, 3, 4, 8, 10, 12, 14, 16, 17, 21, 23, 25
- surfaceSimulate, 3, 8, 14, 16, 19, 19, 25
- surfaceSummary, 3, 14, 16, 19, 22, 25
- surfaceTraitPlot, 3, 14, 21
- surfaceTraitPlot (surfaceTreePlot), 23
- surfaceTreePlot, 3, 4, 14, 16, 19, 21, 23