Package ‘hisse’

May 15, 2021

Version 1.9.19

Date 2021-05-13

Title Hidden State Speciation and Extinction

Maintainer Jeremy Beaulieu <jmbeauli@uark.edu>

Depends ape, deSolve, GenSA, subplex, nloptr

Suggests testthat, knitr, rmarkdown

Imports parallel, phytools, data.table, methods, diversitree, plotrix

Description Sets up and executes a HiSSE model (Hidden State Speciation and Extinction) on a phylogeny and character sets to test for hidden shifts in trait dependent rates of diversification. Beaulieu and O'Meara (2016) <doi:10.1093/sysbio/syw022>.

License GPL (>= 2)

VignetteBuilder knitr

RoxygenNote 7.1.1

NeedsCompilation yes

Author Jeremy Beaulieu [aut, cre],
Brian O'Meara [aut],
Daniel Caetano [aut],
James Boyko [aut]

Repository CRAN

Date/Publication 2021-05-15 06:10:02 UTC

R topics documented:

`Compute Model Weights` *Compute model weights*

Description

Computes the Akaike model weights for a list of HiSSE, MiSSE, and/or GeoHiSSE model fits.

Usage

```
GetAICWeights(hisse.results, criterion="AIC")
```

Arguments

`hisse.results`	A list of models (such as from MiSSEGreedy or just putting individual hisse runs in a list)
`criterion`	Which criterion to use, AIC or AICc (with correction for sample size)

Details

Function computes the model weight from their AIC values using the formula:

“delta <- mod.AIC - min(mod.AIC)”

“AICw <- exp(-0.5 * delta) / sum(exp(-0.5 * delta))”

Function will return vector of weights

Author(s)

Daniel Caetano

`generateMiSSEGreedyCombinations`

Combinations for Character-free State Speciation and Extinction Searching Greedily

---

Description

Sets up a set of MiSSE models (Missing State Speciation and Extinction) on a phylogeny, varying the number of parameters for turnover and extinction fraction.

Usage

```
generateMiSSEGreedyCombinations(max.param=52, turnover.tries=sequence(26),
eps.tries=sequence(26), fixed.eps.tries=c(0, 0.9, NA), vary.both=TRUE)
```

Arguments

`max.param`	how many parameters to try estimating at the most.
`turnover.tries`	vector of number of free turnover parameters.
`eps.tries`	vector of number of free eps parameters.
`fixed.eps.tries`	fixed eps values to use (and an NA to allow estimating, as well).
`vary.both`	if TRUE, allows models that have multiple parameters for turnover and eps; if FALSE, if turnover has multiple hidden states eps can only have one estimated value and vice versa.

Details

This creates the set of combinations of models to run through MiSSEGreedy as a data.frame. It has columns for the number of rates to estimate for turnover, the number of values to estimate for eps, and any fixed values for eps to apply to the whole tree. You can add your own rows or delete some to this data.frame to add more or fewer combinations. By default, this comes out ordered so that simpler models are run first in `MiSSEGreedy` but that is not required (but wise for most use cases), and you can reorder if you wish.

It can be worth considering how much information your tree has for diversification rates. A fully resolved tree with N taxa has 2N-2 branches and only N-1 internal node heights, which is more relevant for diversification models. So on a 50 taxon tree, there are 49 node heights – it's a bit ambitious to think you can extract, say, 5 different parameters (3 turnover rates, 1 eps value, 1 transition rate) from such a tree. People clearly try to extract more – some methods claim to use information in a tree to extract a different speciation or net diversification rate for every single tip – but that could be a tad ambitious. So setting `max.param` to a low value makes a lot of sense. `max.param = floor(ape::Ntip(phy)/10)` is ridiculously optimistic (dividing by 100 or more is probably more conservative) but if things are running well `MiSSEGreedy` should stop before getting to the models that are too complex.

`turnover.tries` and `eps.tries` set how many turnover and eps hidden states to try, respectively. If you wanted to try only 1, 3, and 7 hidden states for turnover you would set `turnover.tries = c(1,3,7)` for example.

Estimating extinction rates is hard. This affects all diversification models (even if all you want and look at is speciation rate, extinction rate estimates still affect what this is as they affect the likelihood). It is most noticeable in MiSSE with eps, the extinction fraction (extinction rate divided by speciation rate). One option, following Magallon & Sanderson (2001), is to set extinction fraction at set values. By default, we use theirs, 0 (meaning a Yule model - no extinction) or 0.9 (a lot of extinction, though still less than paleontoligists find). You can set your own in `fixed.eps.tries`. If you only want to use fixed values, and not estimate, get rid of the NA, as well. However, don't "cheat" – if you use a range of values for fixed.eps, it's basically doing a search for this, though the default AICc calculation doesn't dQuoteknow this to penalize it for another parameter.

HiSSE and thus MiSSE assume that a taxon has a particular hidden state (though they recognize that there can be uncertainty in which state it actually has). Thus, they're written to assume that we dQuotepaint these states on the tree and a given state affects both turnover and eps. So if turnover has four hidden states, eps has four hidden states. They can be constrained: the easiest way is to have, say, turnover having an independent rate for each hidden state and eps having the same rate for all the hidden states. If `vary.both` is set to FALSE, all models are of this sort: if turnover varies, eps is constant across all hidden states, or vice versa. Jeremy Beaulieu prefers this. If `vary.both` is set to TRUE, both can vary: for example, there could be five hidden states for both turnover and eps, but turnover lets each of these have a different rate, but eps only allows three values (so that eps_A and eps_D might be forced to be equal, and eps_B and eps_E might be forced to be equal). Brian O'Meara would consider allowing this, while cautioning you about the risks of too many parameters.

Value

`generateMiSSEGreedyCombinations` returns a data.frame to pass to MiSSEGreedy().

Author(s)

Brian C. O'Meara

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Herrera-Alsina, L., P. van Els, and R.S. Etienne. 2018. Detecting the dependence of diversification on multiples traits from phylogenetic trees and trait data. Systematic Biology, 68:317-328.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Magallon S. and Sanderson M.J. 2001. Absolute diversification rates in angiosperm clades. Evolution 55:1762-1780.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`GeoHiSSE` *Hidden Geographic State Speciation and Extinction*

Description

Sets up and executes a GeoHiSSE model (Hidden Geographic State Speciation and Extinction) on a very large phylogeny and character distribution.

Usage

```
GeoHiSSE(phy, data, f=c(1,1,1), turnover=c(1,2,3), eps=c(1,2),
hidden.states=FALSE, trans.rate=NULL, assume.cladogenetic=TRUE,
condition.on.survival=TRUE, root.type="madfitz", root.p=NULL, sann=TRUE,
sann.its=1000, bounded.search=TRUE,  max.tol=.Machine$double.eps^.50,
mag.san.start=0.5, starting.vals=NULL, turnover.upper=1000,
eps.upper=3, trans.upper=100, restart.obj=NULL, ode.eps=0, dt.threads=1)
```

Arguments

`phy` a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a matrix (or dataframe) with two columns containing species information. First column has the species names and second column has area codes. Values for the areas need to be 0, 1, or 2, where 0 is the widespread area ’01’, 1 is endemic area ’00’ and 2 is endemic area ’11’. See ’Details’.

`f` vector of length 3 with the estimated proportion of extant species in state 1 (area ’00’), state 2 (area ’1’), and state 0 (widespread area ’01’) that are included in the phylogeny. A value of c(0.25, 0.25, 0.5) means that 25 percent of species in areas ’00’ and ’11’ and 50 percent of species in area ’01’ are included in the phylogeny. By default all species are assumed to be sampled.

`turnover` a numeric vector of length equal to 3+(number of `hidden.states` * 3). A GeoSSE model has 3 turnover parameters: tau00, tau11 and tau01. A GeoHiSSE model with one hidden area has 6 speciation parameters: tau00A, tau11A, tau01A, tau00B, tau11B, and tau01B, and so on. The length of the numeric vector needs to match the number of speciation parameters in the model.

`eps` a numeric vector of length equal to 2+(number of `hidden.states` * 2). A GeoSSE model has 2 extinct fraction parameters: ef00 and ef11. A GeoHiSSE model with one hidden area has 4 extinct.frac parameters: ef00A, ef11A, ef00B, and ef11B, and so on. The length of the numeric vector needs to match the number of extinct.frac parameters in the model.

`hidden.states` a logical indicating whether the model includes hidden.states. The default is `FALSE`.

`trans.rate` provides the transition rate model. See function `TransMatMakerGeoHiSSE`.

`assume.cladogenetic`
assumes that cladogenetic events occur at nodes. The default is `TRUE`.

`condition.on.survival`
a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`. Order of the areas in the vector need to follow: root.p[1] = 1 (endemic area ’0’); root.p[2] = 2 (endemic area ’1’); root.p[3] = 0 (widespread area ’01’).

`sann` a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `TRUE`.

`sann.its` a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search` a logical indicating whether or not bounds should be enforced during optimization. The default is is `TRUE`.

`max.tol` supplies the relative optimization tolerance to `subplex`.

`mag.san.start` Sets the extinction fraction to estimate the starting values for the diversification parameters. The equation used is based on Magallon and Sanderson (2001), and follows the procedure used in the original GeoSSE implementation.

`starting.vals` a numeric vector of length 3 with starting values for the model for all areas and hidden states. Position [1] sets turnover, [2] sets extinction fraction, and [3] dispersal rates.

`turnover.upper` sets the upper bound for the speciation parameters.

`eps.upper` sets the upper bound for the extirpation parameters.

`trans.upper` sets the upper bound for the transition rate parameters.

`restart.obj` an object that contains everything to restart an optimization.

`ode.eps` sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

`dt.threads` sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

This function sets up and executes a more complex and faster version of the GeoHiSSE model (for the original function see `GeoHisse.old`). One of the main differences here is that the model allows up to 10 hidden categories, and implements a more efficient means of carrying out the branch calculation. Specifically, we break up the tree into carry out all descendent branch calculations simultaneously, combine the probabilities based on their shared ancestry, then repeat for the next set of descendent . In testing, we've found that as the number of taxa increases, the calculation becomes much more efficient. In future versions, we will likely allow for multicore processing of these calculations to further improve speed. Also, note this function has replaced the version of `GeoSSE` that is currently available (see `GeoHisse.old`).

The other main difference is that, like `HiSSE`, we employ a modified optimization procedure. In other words, rather than optimizing birth and death separately, `GeoHisse` optimizes orthogonal transformations of these variables: we let tau = birth+death define "net turnover", and we let eps = death/birth define the "extinction fraction". This reparameterization alleviates problems associated with overfitting when birth and death are highly correlated, but both matter in explaining the diversity pattern.

To setup a model, users input vectors containing values to indicate how many free parameters are to be estimated for each of the variables in the model. This is done using the `turnover` and `extinct.frac` parameters. One needs to specify a value for each of the parameters of the model, when two parameters show the same value, then the parameters are set to be linked during the estimation of the model. For example, a GeoHiSSE model with 1 hidden area and all free parameters has `turnover = 1:6`. The same model with speciation rates constrained to be the same for all hidden states has `turnover = c(1,2,3,1,2,3)`. This same format applies to `extinct.frac`.

Once the model is specified, the parameters can be estimated using the subplex routine (default), or use a two-step process (i.e., sann=TRUE) that first employs a stochastic simulated annealing procedure, which is later refined using the subplex routine.

The "trans.rate" input is the transition model and has an entirely different setup than speciation and extirpation rates. See `TransMatMakerGeoHiSSE` function for more details.

For user-specified "root.p", you should specify the probability for each area. If you are doing a hidden model, there will be six areas: 0A, 1A, 2A, 0B, 1B, 2B. So if you wanted to say the root had to be in area 0 (widespread distribution), you would specify "root.p = c(0.5, 0, 0, 0.5, 0, 0)". In other words, the root has a 50% chance to be in one of the areas 0A or 0B.

For the "root.type" option, we are currently maintaining the previous default of "madfitz". However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing "herr_als". It is unclear to us which is exactly correct, but it does seem that both "madfitz" and "herr_als" behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We've also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the "madfitz" conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

Also, note, that in the case of "root.type=user" and "root.type=equal" are no longer explicit "root.type" options. Instead, either "madfitz" or "herr_als" are specified and the "root.p" can be set to allow for custom root options.

Value

`GeoHiSSE` returns an object of class `geohisse.fit`. This is a list with elements:

`$loglik` the maximum negative log-likelihood.

`$AIC` Akaike information criterion.

`$AICc` Akaike information criterion corrected for sample-size.

`$solution` a matrix containing the maximum likelihood estimates of the model parameters.

`$index.par` an index matrix of the parameters being estimated.

`$f` user-supplied sampling frequencies.

`$hidden.states` a logical indicating whether hidden states were included in the model.

`$assume.cladogenetic` a logical indicating whether cladogenetic events were allowed at nodes.

`$condition.on.surivival` a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$phy` user-supplied tree

`$data` user-supplied dataset

`$trans.matrix` the user-supplied transition matrix

`$max.tol` relative optimization tolerance.

`$starting.vals` The starting values for the optimization.

`$upper.bounds` the vector of upper limits to the optimization search.

`$lower.bounds` the vector of lower limits to the optimization search.

`$ode.eps` The ode.eps value used for the estimation.

Author(s)

Jeremy M. Beaulieu

References

Caetano, D.S., B.C. O'Meara, and J.M. Beaulieu. 2018. Hidden state models improve state-dependent diversification approaches, including biogeographic models. Evolution, 72:2308-2324.

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., W.P. Maddison, and S.P. Otto. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., P.E. Midford, and S.P. Otto. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., R.M. May, and P.H. Harvey. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

GeoHiSSE.old *Original Hidden Geographic State Speciation and Extinction*

Description

Sets up and executes the original GeoHiSSE model (Hidden Geographic State Speciation and Extinction) on a phylogeny and character distribution.

Usage

```
GeoHiSSE.old(phy, data, f=c(1,1,1), speciation=c(1,2,3), extirpation=c(1,2),
hidden.areas=FALSE, trans.rate=NULL, assume.cladogenetic=TRUE,
condition.on.survival=TRUE, root.type="madfitz", root.p=NULL, sann=TRUE,
sann.its=1000, bounded.search=TRUE,  max.tol=.Machine$double.eps^.50,
mag.san.start=0.5, starting.vals=NULL, speciation.upper=1000, extirpation.upper=1000,
trans.upper=100, ode.eps=0)
```

Arguments

`phy` a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a matrix (or dataframe) with two columns containing species information. First column has the species names and second column has area codes. Values for the areas need to be 0, 1, or 2, where 0 is the widespread area ’01’, 1 is endemic area ’0’ and 2 is endemic area ’1’. See ’Details’.

`f` vector of length 3 with the estimated proportion of extant species in state 1 (area ’0’), state 2 (area ’1’), and state 0 (widespread area ’01’) that are included in the phylogeny. A value of c(0.25, 0.25, 0.5) means that 25 percent of species in areas ’0’ and ’1’ and 50 percent of species in area ’01’ are included in the phylogeny. By default all species are assumed to be sampled.

`speciation` a numeric vector of length equal to 3+(number of `hidden.areas` * 3). A GeoSSE model has 3 speciation parameters: s0, s1 and s01. A GeoHiSSE model with one hidden area has 6 speciation parameters: s0A, s1A, s01A, s0B, s1B, s01B. And so on. The length of the numeric vector needs to match the number of speciation parameters in the model. See ’Details’.

`extirpation` a numeric vector of length equal to 2+(number of `hidden.areas` * 2). A GeoSSE model has 2 extirpation parameters: x0 and x1. A GeoHiSSE model with one hidden area has 4 extirpation parameters: x0A, x1A, x0B, and x1B. And so on. The length of the numeric vector needs to match the number of extirpation parameters in the model. See ’Details’.

`hidden.areas` a logical indicating whether the model includes a hidden area. The default is `FALSE`.

`trans.rate` provides the transition rate model. See function `TransMatMakerGeoHiSSE.old`.

`assume.cladogenetic` assumes that cladogenetic events occur at nodes. The default is `TRUE`.

`condition.on.survival`
: a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type`
: indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

`root.p`
: a vector indicating fixed root state probabilities. The default is `NULL`. Order of the areas in the vector need to follow: root.p[1] = 1 (endemic area ’0’); root.p[2] = 2 (endemic area ’1’); root.p[3] = 0 (widespread area ’01’).

`sann`
: a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `TRUE`.

`sann.its`
: a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search`
: a logical indicating whether or not bounds should be enforced during optimization. The default is is `TRUE`.

`max.tol`
: supplies the relative optimization tolerance to `subplex`.

`mag.san.start`
: Sets the extinction fraction to estimate the starting values for the diversification parameters. The equation used is based on Magallon and Sanderson (2001), and follows the procedure used in the original GeoSSE implementation.

`starting.vals`
: a numeric vector of length 3 with starting values for the model for all areas and hidden states. Position [1] sets turnover, [2] sets extinction fraction, and [3] dispersal rates.

`speciation.upper`
: sets the upper bound for the speciation parameters.

`extirpation.upper`
: sets the upper bound for the extirpation parameters.

`trans.upper`
: sets the upper bound for the transition rate parameters.

`ode.eps`
: sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

Details

This function sets up and executes the original GeoHiSSE model. The model closely follows `diversitree`, although here we employ modified optimization procedures. As for data file format, `GeoHiSSE` expects a two column matrix or data frame, with the first column containing the species names and the second containing the are information. The area information need to be in the format of three numbers: 0 for area ’01’, 1 for area ’0’, and 2 for ’1’. Please note that the code for the areas here differ from the `make.geosse` function of package `diversitree`. The order of the data file and the names in the “phylo” object need not be in the same order; `hisse` deals with this internally. Also, the character information MUST be 0, 1, or 2, otherwise, the function will return an error message.

To setup a model, users input vectors containing values to indicate how many free parameters are to be estimated for each of the variables in the model. This is done using the `speciation` and `extirpation` parameters. One needs to specify a value for each of the parameters of the model, when two parameters show the same value, then the parameters are set to be linked during the estimation of the model. For example, a GeoHiSSE model with 1 hidden area and all free parameters has `speciation = 1:6`. The same model with speciation rates constrained to be the same for all hidden areas has `speciation = c(1,2,3,1,2,3)`. This same format applies to `extirpation`. Please note that GeoHiSSE currently works with up to 4 hidden areas. The most complex model would be `speciation = 1:15` and `extirpation = 1:10`.

Once the model is specified, the parameters can be estimated using the subplex routine (default), or use a two-step process (i.e., sann=TRUE) that first employs a stochastic simulated annealing procedure, which is later refined using the subplex routine.

The "trans.rate" input is the transition model and has an entirely different setup than speciation and extirpation rates. See `TransMatMakerGeoHiSSE.old` function for more details.

For user-specified "root.p", you should specify the probability for each area. If you are doing a hidden model, there will be six areas: 0A, 1A, 01A, 0B, 1B, 01B. So if you wanted to say the root had to be in area 0 (widespread distribution), you would specify "root.p = c(0.5, 0, 0, 0.5, 0, 0)". In other words, the root has a 50% chance to be in one of the areas 0A or 0B.

For the "root.type" option, we are currently maintaining the previous default of "madfitz". However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing "herr_als". It is unclear to us which is exactly correct, but it does seem that both "madfitz" and "herr_als" behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We've also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the "madfitz" conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

Also, note, that in the case of "root.type=user" and "root.type=equal" are no longer explicit "root.type" options. Instead, either "madfitz" or "herr_als" are specified and the "root.p" can be set to allow for custom root options.

Value

`GeoHiSSE` returns an object of class `geohisse.fit`. This is a list with elements:

`$loglik`	the maximum negative log-likelihood.
`$AIC`	Akaike information criterion.
`$AICc`	Akaike information criterion corrected for sample-size.
`$solution`	a matrix containing the maximum likelihood estimates of the model parameters.
`$index.par`	an index matrix of the parameters being estimated.
`$f`	user-supplied sampling frequencies.
`$hidden.areas`	a logical indicating whether hidden areas were included in the model.
`$assume.cladogenetic`	a logical indicating whether cladogenetic events were allowed at nodes.

`$condition.on.surivival` a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$timeslice` indicates whether the user-specified timeslice that split the tree.

`$phy` user-supplied tree

`$data` user-supplied dataset

`$trans.matrix` the user-supplied transition matrix

`$max.tol` relative optimization tolerance.

`$starting.vals` The starting values for the optimization.

`$upper.bounds` the vector of upper limits to the optimization search.

`$lower.bounds` the vector of lower limits to the optimization search.

`$ode.eps` The ode.eps value used for the estimation.

Author(s)

Jeremy M. Beaulieu

References

Caetano, D.S., B.C. O'Meara, and J.M. Beaulieu. 2018. Hidden state models improve state-dependent diversification approaches, including biogeographic models. Evolution, 72:2308-2324.

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., W.P. Maddison, and S.P. Otto. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Goldberg, E. E., L. T. Lancaster, and R. H. Ree. 2011. Phylogenetic Inference of Reciprocal Effects between Geographic Range Evolution and Diversification. Syst. Biol. 60:451-465.

Maddison W.P., P.E. Midford, and S.P. Otto. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., R.M. May, and P.H. Harvey. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`hisse` *Hidden State Speciation and Extinction*

Description

Sets up and executes a HiSSE model (Hidden State Speciation and Extinction) on a phylogeny and character distribution.

Usage

```
hisse(phy, data, f=c(1,1), turnover=c(1,2), eps=c(1,2),
hidden.states=FALSE, trans.rate=NULL, condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL, sann=TRUE, sann.its=1000,
bounded.search=TRUE,  max.tol=.Machine$double.eps^.50, starting.vals=NULL,
turnover.upper=10000, eps.upper=3, trans.upper=100, restart.obj=NULL,
ode.eps=0, dt.threads=1)
```

Arguments

`phy` a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a matrix (or dataframe) with three columns. The first column containing the species names and the second and third containing the binary character information. Character "0" is on column 2 and chracter "1" is on column 3. A value of 0 means character absent and a value of 1 character present. The input of data follows a Pagel model. See 'Details'.

`f` vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be sampled.

`turnover` a numeric vector indicating the number of free turnover parameters in the model.

`eps` a numeric vector indicating the number of free extinction fraction parameters in the model.

`hidden.states` a logical indicating whether the model includes a hidden states. The default is `FALSE`.

`trans.rate` provides the transition rate model. See function `TransMatMakerHiSSE`.

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, "madfitz" or Herrera-Alsina et al. 2018, "herr_als".

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`sann` a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `TRUE`.

`sann.its` a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search` a logical indicating whether or not bounds should be enforced during optimization. The default is is `TRUE`.

`max.tol` supplies the relative optimization tolerance to `subplex`.

`starting.vals` a numeric vector of length 3 with starting values for the model for all areas and hidden states. Position [1] sets turnover, [2] sets extinction fraction, and [3] transition rates.

`turnover.upper` sets the upper bound for the turnover parameters.

`eps.upper` sets the upper bound for the eps parameters.

`trans.upper` sets the upper bound for the transition rate parameters.

`restart.obj` an object of class that contains everything to restart an optimization.

`ode.eps` sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

`dt.threads` sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

This function sets up and executes a new and faster version of the HiSSE model. Note that the four-state character-independent model can be called from this command in addition to the two-state BiSSE model and the full character-dependent HiSSE model. See vignette on how to set this up.

The “trans.rate” input is the transition model and has an entirely different setup than turnover rates and extinction fraction. See `TransMatMakerHiSSE` function for more details.

For the “root.type” option, we are currently maintaining the previous default of “madfitz”. However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing “herr_als”. It is unclear to us which is exactly correct, but it does seem that both “madfitz” and “herr_als” behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We’ve also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the “madfitz” conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

For user-specified “root.p”, you should specify the probability for each state. If you are doing a hidden model, there will be four states: 0A, 1A, 0B, 1B. So if you wanted to say the root had to be state 0, you would specify “root.p = c(0.5, 0, 0.5, 0)”.

This code will completely replace the original hisse function in the next version.

Value

`hisse` returns an object of class `hisse.fit`. This is a list with elements:

`$loglik` the maximum negative log-likelihood.

`$AIC` Akaike information criterion.

`$AICc` Akaike information criterion corrected for sample-size.

`$solution` a matrix containing the maximum likelihood estimates of the model parameters.

`$index.par` an index matrix of the parameters being estimated.

`$f` user-supplied sampling frequencies.

`$hidden.states` a logical indicating whether hidden states were included in the model.

`$condition.on.surivival`
a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$phy` user-supplied tree

`$data` user-supplied dataset

`$trans.matrix` the user-supplied transition matrix

`$max.tol` relative optimization tolerance.

`$starting.vals` The starting values for the optimization.

`$upper.bounds` the vector of upper limits to the optimization search.

`$lower.bounds` the vector of lower limits to the optimization search.

`$ode.eps` The ode.eps value used for the estimation.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Herrera-Alsina, L., P. van Els, and R.S. Etienne. 2018. Detecting the dependence of diversification on multiples traits from phylogenetic trees and trait data. Systematic Biology, 68:317-328.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
library(diversitree)
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- tree.bisse(pars, max.t=30, x0=0)
sim.dat <- data.frame(names(phy$tip.state), phy$tip.state)

## Fit BiSSE equivalent:
trans.rates.bisse <-  TransMatMakerHiSSE(hidden.traits=0)
pp.bisse <- hisse(phy, sim.dat, hidden.states=FALSE, turnover=c(1,2),
eps=c(1,2), trans.rate=trans.rates.bisse)
```

```
## Now fit HiSSE equivalent with a hidden state for state 1:
trans.rates.hisse <- TransMatMakerHiSSE(hidden.traits=1)
pp.hisse <- hisse(phy, sim.dat, hidden.states=TRUE, turnover=c(1,2,1,2),
eps=c(1,2,1,2), trans.rate=trans.rates.hisse)
```

hisse.null4.old *Original Four state trait-independent Hidden State Speciation and Extinction*

Description

Sets up and executes the original four state trait-independent HiSSE model (Hidden State Speciation and Extinction) on a phylogeny and character set.

Usage

```
hisse.null4.old(phy, data, f=c(1,1), turnover.anc=rep(c(1,2,3,4),2),
eps.anc=rep(c(1,2,3,4),2), trans.type="equal", condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL,  output.type="turnover", sann=TRUE,
sann.its=1000, bounded.search=TRUE, max.tol=.Machine$double.eps^.50,
starting.vals=NULL, turnover.upper=10000, eps.upper=3, trans.upper=100,
ode.eps=0)
```

Arguments

`phy` a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a data matrix containing species information (see Details).

`f` vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be sampled.

`turnover.anc` a vector of length 8, indicating the free parameters associated with the net turnover rates. Default setting assumes character independent diversification (see Details).

`eps.anc` a vector of length 8, indicating the free parameters associated with the extinction fractions. Default setting assumes character independent diversification (see Details).

`trans.type` provides the type of transition rate model. Currently this model allows two types: “equal”, the default, which assumes all transitions are equal, and “three.rate”, that assumes three rates (see Details).

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`output.type` indicates whether the rates should be printed onscreen as the optimized variables, “turnover”, transformed to reflect net diversification, “net.div”, or transformed to reflect λ and μ, “raw”.

`sann` a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `TRUE`.

`sann.its` a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search` a logical indicating whether or not bounds should be enforced during optimization. The default is is `TRUE`.

`max.tol` supplies the relative optimization tolerance to `subplex`.

`starting.vals` a vector of starting values to be used instead of the default settings. These are just three values given in the following order: turnover (1), extinction fraction (2), and a single transition rate (3)

`turnover.upper` sets the upper bound for the turnover parameters. The default upper bound assumes an event occurs every 100 years.

`eps.upper` sets the upper bound for the extinction fraction parameters.

`trans.upper` sets the upper bound for the transition rate parameters.

`ode.eps` sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

Details

This function sets up and executes a four-state trait independent HiSSE model. The model closely follows `hisse.old`. However, note that this function is no longer necessary and can be called and evaluated directly using the new `hisse` function.

Like `hisse.old`, users input vectors containing values to indicate how many free parameters are to be estimated for each of the variables in the model. However, the null-four model assumes that “turnover.anc” and “eps.anc” are linked between the two observed states. Thus, users are unlikely to alter the inputs much, aside from perhaps fixing “turnover.anc” or “eps.anc” to be equal across the four hidden states, where the “turnover.anc” input vector is set as rep(c(1,1,1,1),2). For a Yule equivalent, the input vector for “eps.anc” would be rep(c(0,0,0,0),2). For how to setup a null-two model see the example code below.

For user-specified “root.p”, you should specify the probability for each state. See help for “hisse.old” for more on other parameters for this function.

For the “root.type” option, we are currently maintaining the previous default of “madfitz”. However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing “herr_als”. It is unclear to us which is exactly correct, but it does seem that both “madfitz” and “herr_als” behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We’ve also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the “madfitz” conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

Also, note, that in the case of “root.type=user” and “root.type=equal” are no longer explicit “root.type” options. Instead, either “madfitz” or “herr_als” are specified and the “root.p” can be set to allow for custom root options.

Value

`hisse.null4.old` returns an object of class `hisse.fit`. This is a list with elements:

`$loglik` the maximum negative log-likelihood.

`$AIC` Akaike information criterion.

`$AICc` Akaike information criterion corrected for sample-size.

`$solution` a matrix containing the maximum likelihood estimates of the model parameters.

`$index.par` an index matrix of the parameters being estimated.

`$f` user-supplied sampling frequencies.

`$condition.on.surivival`
a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$phy` user-supplied tree

`$data` user-supplied dataset

`$output.type` the user-specified output.type to be printed on the screen.

`$trans.type` the user-specified transition model.

`$trans.mat` the index matrix that specifies the free parameters in the transition model.

`$max.tol` relative optimization tolerance.

`$upper.bounds` the vector of upper limits to the optimization search.

`$lower.bounds` the vector of lower limits to the optimization search.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

Examples

```
## Not run
library(diversitree)
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- tree.bisse(pars, max.t=30, x0=0)
sim.dat <- data.frame(names(phy$tip.state), phy$tip.state)

## Fit null-four HiSSE:
pp.hisse.null <- hisse.null4.old(phy, sim.dat, turnover.anc=rep(c(1,2,3,4),2),
eps.anc=rep(c(1,2,3,4),2), trans.type="equal")

## Fit null-two HiSSE model:
trans.rates.hisse <- TransMatMaker.old(hidden.states=TRUE)
trans.rates.hisse <- ParDrop(trans.rates.hisse, c(3,5,8,10))
trans.rates.hisse[!is.na(trans.rates.hisse) & !trans.rates.hisse == 0] = 1
pp.hisse <- hisse.old(phy, sim.dat, hidden.states=TRUE, turnover.anc=c(1,1,2,2),
eps.anc=c(1,1,2,2), trans.rate=trans.rates.hisse)
```

---

hisse.old *Original Hidden State Speciation and Extinction*

Description

Sets up and executes the original HiSSE model (Hidden State Speciation and Extinction) on a phylogeny and character distribution.

Usage

```
hisse.old(phy, data, f=c(1,1), hidden.states=TRUE, turnover.anc=c(1,1,0,0),
eps.anc=c(1,1,0,0), trans.rate=NULL, turnover.beta=c(0,0,0,0),
eps.beta=c(0,0,0,0), timeslice=NULL, condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL, output.type="turnover", sann=TRUE,
sann.its=1000, bounded.search=TRUE, max.tol=.Machine$double.eps^.50,
starting.vals=NULL, turnover.upper=10000, eps.upper=3, trans.upper=100,
ode.eps=0)
```

Arguments

phy	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.
data	a data matrix containing species information (see Details).
f	vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be sampled.

`hidden.states` a logical indicating whether the model includes a hidden state. The default is `FALSE`.

`turnover.anc` a vector of length 4, indicating the free parameters associated with the net turnover rates. Default settings is a BiSSE model with fixed turnover rates for both observed states (see Details).

`eps.anc` a vector of length 4, indicating the free parameters associated with the extinction fractions. Default settings is a BiSSE model with fixed extinction fractions for both observed states (see Details).

`trans.rate` provides the transition rate model.

`turnover.beta` a vector of length 4, indicating the free parameters associated with time-varying net turnover rates (see Details).

`eps.beta` a vector of length 4, indicating the free parameters associated with time-varying extinction fractions (see Details).

`timeslice` a user-supplied time to split the tree.

`condition.on.survival`

a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`output.type` indicates whether the rates should be printed onscreen as the optimized variables, “turnover”, transformed to reflect net diversification, “net.div”, or transformed to reflect λ and μ, “raw”.

`sann` a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `TRUE`.

`sann.its` a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search` a logical indicating whether or not bounds should be enforced during optimization. The default is is `TRUE`.

`max.tol` supplies the relative optimization tolerance to `subplex`.

`starting.vals` a vector of starting values to be used instead of the default settings. These are just three values given in the following order: turnover (1), extinction fraction (2), and a single transition rate (3)

`turnover.upper` sets the upper bound for the turnover parameters. The default upper bound assumes an event occurs every 100 years.

`eps.upper` sets the upper bound for the extinction fraction parameters.

`trans.upper` sets the upper bound for the transition rate parameters.

`ode.eps` sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

Details

This function sets up and executes the original HiSSE model. The model closely follows `diversitree`, although here we employ modified optimization procedures. For example, rather than optimizing birth and death separately, hisse optimizes orthogonal transformations of these variables: we let tau = birth+death define "net turnover", and we let eps = death/birth define the “extinction fraction”. This reparameterization alleviates problems associated with overfitting when birth and death are highly correlated, but both matter in explaining the diversity pattern. As for data file format, `hisse` expects a two column matrix or data frame, with the first column containing the species names and the second containing the binary character information. Note that the order of the data file and the names in the “phylo” object need not be in the same order; `hisse` deals with this internally. Also, the character information should be binary and coded as 0 and 1, otherwise the function will misbehave. However, if the state for a species is unknown, a user can specify this with a 2, and the state will be considered maximally ambiguous.

To setup a model, users input vectors containing values to indicate how many free parameters are to be estimated for each of the variables in the model. For example, the “turnover.anc” input vector is set by default as c(1,1,0,0). This means for state 0 and state 1, we are allowing one free parameter to define the net turnover rate (birth+death) in the model. This is essentially a BiSSE model with fixed turnover rates. Now, say we want to include separate turnover rates for both states we would simply input c(1,2,0,0). The last two entries, which in the preceding example are set to zero, correspond to the hidden states; the third entry corresponds to a hidden state associated with observed state 0, such that 0A (hidden state absent) is the first entry, and 0B (hidden state present) is the third entry. So, to set up a model with three turnover rates, where we include a free parameter for a hidden state associated with state 0 we input c(1,2,3,0). A full model would thus be c(1,2,3,4), which corresponds to four separate net turnover rates, for states 0A (first entry), 1A (second entry), 0B (third entry), and 1B (fourth entry). Extinction fraction, or “eps.anc”, follows the same format, though including a zero for a state we want to include in the model corresponds to no extinction, which is the Yule equivalent. In general, we follow this format to make it easier to generate a large set of nested models. Once the model is specified, the parameters can be estimated using the subplex routine (default), or use a two-step process (i.e., sann=TRUE) that first employs a stochastic simulated annealing procedure, which is later refined using the subplex routine.

The “trans.rate” input is the transition model and has an entirely different setup than turnover and extinction rates. See `TransMatMaker` function for more details.

For user-specified “root.p”, you should specify the probability for each state. If you are doing a hidden model, there will be four states: 0A, 1A, 0B, 1B. So if you wanted to say the root had to be state 0, you would specify “root.p = c(0.5, 0, 0.5, 0)”.

For the “root.type” option, we are currently maintaining the previous default of “madfitz”. However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing “herr_als”. It is unclear to us which is exactly correct, but it does seem that both “madfitz” and “herr_als” behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We’ve also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the “madfitz” conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

Also, note, that in the case of “root.type=user” and “root.type=equal” are no longer explicit “root.type” options. Instead, either "madfitz" or "herr_als" are specified and the "root.p" can be set to allow for custom root options.

Finally, the options ".beta" and "timeslice" are included, but neither have been tested – needless to say, use at your own risk (but really, though, you should probably forget that these options exist for the time being). The ".beta" provides a means for testing for time-varying rates, whereas "timeslice" splits the tree to allow the process to vary before and after some user defined time period. These options will be further developed in due course.

Value

`hisse.old` returns an object of class `hisse.fit`. This is a list with elements:

`$loglik` the maximum negative log-likelihood.

`$AIC` Akaike information criterion.

`$AICc` Akaike information criterion corrected for sample-size.

`$solution` a matrix containing the maximum likelihood estimates of the model parameters.

`$index.par` an index matrix of the parameters being estimated.

`$f` user-supplied sampling frequencies.

`$hidden.states` a logical indicating whether a hidden state was included in the model.

`$condition.on.surivival`
a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$timeslice` indicates whether the user-specified timeslice that split the tree.

`$phy` user-supplied tree

`$data` user-supplied dataset

`$iterations` number of iterations of the likelihood search that were executed.

`$output.type` the user-specified output.type to be printed on the screen.

`$max.tol` relative optimization tolerance.

`$upper.bounds` the vector of upper limits to the optimization search.

`$lower.bounds` the vector of lower limits to the optimization search.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Herrera-Alsina, L., P. van Els, and R.S. Etienne. 2018. Detecting the dependence of diversification on multiples traits from phylogenetic trees and trait data. Systematic Biology, 68:317-328.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
library(diversitree)
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- tree.bisse(pars, max.t=30, x0=0)
sim.dat <- data.frame(names(phy$tip.state), phy$tip.state)
## Fit BiSSE equivalent:
trans.rates.bisse <- TransMatMaker.old(hidden.states=FALSE)
pp.bisse <- hisse.old(phy, sim.dat, hidden.states=FALSE, turnover.anc=c(1,2,0,0),
eps.anc=c(1,2,0,0), trans.rate=trans.rates.bisse)

## Now fit HiSSE equivalent with a hidden state for state 1:
trans.rates.hisse <- TransMatMaker.old(hidden.states=TRUE)
trans.rates.hisse <- ParDrop(trans.rates.hisse, c(2,3,5,7,8,9,10,12))
pp.hisse <- hisse.old(phy, sim.dat, hidden.states=TRUE, turnover.anc=c(1,2,0,3),
eps.anc=c(1,2,0,3), trans.rate=trans.rates.hisse)
```

---

makeGeoHiSSELikelihood

Export likelihood function for the GeoHiSSE model

Description

Exports a likelihood function conditioned on the data and a named vector with the parameters for each of the models.

Usage

```
makeGeoHiSSELikelihood(phy, data, hidden.areas=0, f=c(1,1,1), assume.cladogenetic=TRUE,
condition.on.survival=TRUE, root.type=”madfitz”, root.p=NULL, dt.threads=1, ode.eps = 0,
bad.likelihood = exp(-300))
```

Arguments

`phy` a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a data matrix containing species information (see Details).

`hidden.areas`	Number of hidden areas included in the model. When hidden.areas = 0, the model is equal to the GeoSSE model (i.e., no hidden states included).
`f`	vector of length 3 with the estimated proportion of extant species in areas 0, 1, and 2 (see fgeohisse function for more information) that are included in the phylogeny. By default all species are assumed to be sampled.
`assume.cladogenetic`	a logical indicating whether the likelihood should be conditioned on geographical speciation happening at the nodes. If set to FALSE then all range transitions are treated as anagenetic. The default is `TRUE`. See Caetano et al. (2018 - Evolution) for more information.
`condition.on.survival`	a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.
`root.type`	indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.
`root.p`	a vector indicating fixed root state probabilities. The default is `NULL`.
`dt.threads`	Number of core threads to be used by the data.table function. Default is 1.
`ode.eps`	sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.
`bad.likelihood`	Value returned when there is a problem in the computation of the likelihood value for the function.

Details

This function sets up and returns the likelihood for the GeoHiSSE model together with a vector of parameters. The likelihood function is conditioned on the observed data and will return a value of loglikelihood given a vector of parameter values. The length of the parameter vector as well as the order of the parameter vector cannot be changed. Please pay special attention to the length of the parameter vector and the names of the parameters provided by the “pars” element of the list.

Note that when the likelihood computation fails the function will return the value set as “bad.likelihood”.

Value

`makeGeoHiSSELikelihood` returns a list with elements:

`$loglik`	the likelihood function for the model. This has a single parameter ’p’.
`$pars`	a named vector for the likelihood function pupulated with 0 values.

Author(s)

Jeremy M. Beaulieu and Daniel S. Caetano

References

Caetano, D.S., B.C. O'Meara, and J.M. Beaulieu. 2018. Hidden state models improve state-dependent diversification approaches, including biogeographic models. Evolution, 72:2308-2324.

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
library(diversitree)
library(hisse)
## Generate data:
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- tree.bisse(pars, max.t=30, x0=0)
sim.dat <- data.frame(names(phy$tip.state), phy$tip.state)
## Get lik function:
lik.hisse <- makeHiSSELikelihood(phy = phy, data = sim.dat, hidden.states = FALSE)
likf <- lik.hisse$log.lik
pars <- lik.hisse$pars
## Set the parameter values. Note that we have turnover and eps, not speciation and extinction!
pars <- setNames(c(0.1+0.03,0.2+0.03,0.03/0.1,0.03/0.2,0.01,0.01), names(pars))
## Compute the log-likelihood for the model.
likf(pars)
```

---

makeHiSSELikelihood *Export likelihood function for the HiSSE model*

Description

Exports a likelihood function conditioned on the data and a named vector with the parameters for each of the models.

Usage

```
makeHiSSELikelihood(phy, data, hidden.states=TRUE, null4=FALSE,
f=c(1,1), condition.on.survival=TRUE, root.type="madfitz", root.p=NULL, ode.eps=0)
```

Arguments

`phy` a phylogenetic tree, in `ape` “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a data matrix containing species information (see Details).

`hidden.states` whether the model has hidden states. This is ignored if the option “null4” is selected.

`null4` used to select the null model for the full HiSSE model. This is a special model with 4 hidden states.

`f` vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be sampled.

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`ode.eps` sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

Details

This function sets up and returns the likelihood for the HiSSE model together with a vector of parameters. The likelihood function is conditioned on the observed data and will return a value of loglikelihood given a vector of parameter values. The length of the parameter vector as well as the order of the parameter vector cannot be changed. The parameter values are provided in natural form but are log-transformed for the likelihood evaluation. Please pay special attention to the length of the parameter vector and the names of the parameters provided by the “pars” element of the list.

When the option “null4” is set to TRUE, then the likelihood returned is for the HiSSE null model with 4 hidden states. The returned list will include an additional element named “trans.mat.guide” which can be used as a reference for the meaning of the 32 transition parameters. Note that the original model in “hisse.null4” usually sets all these transitions to a single value. This helps model estimation since information is limited to estimate distinct transition rates between parameters.

Value

`makeHiSSELikelihood` returns a list with elements:

`$loglik` the likelihood function for the model. This has a single parameter ’p’.

`$pars` a named vector for the likelihood function pupulated with 0 values.

`$trans.mat.guide`
: a reference matrix to understand the parameters returned by the "$pars" vector. Only present if the "null4" option is set to TRUE.

Author(s)

Jeremy M. Beaulieu and Daniel S. Caetano

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
library(diversitree)
library(hisse)
## Generate data:
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- tree.bisse(pars, max.t=30, x0=0)
sim.dat <- data.frame(names(phy$tip.state), phy$tip.state)
## Get lik function:
lik.hisse <- makeHiSSELikelihood(phy = phy, data = sim.dat, hidden.states = FALSE)
likf <- lik.hisse$log.lik
pars <- lik.hisse$pars
## Set the parameter values. Note that we have turnover and eps, not speciation and extinction!
pars <- setNames(c(0.1+0.03,0.2+0.03,0.03/0.1,0.03/0.2,0.01,0.01), names(pars))
## Compute the log-likelihood for the model.
likf(pars)
```

MarginRecon.old *Original Ancestral State Estimation based on Marginal Reconstruction*

Description

Estimates the likeliest states for both internal nodes and tips of a phylogeny using the marginal reconstruction algorithm.

Usage

```
MarginRecon.old(phy, data, f, pars, hidden.states=TRUE, four.state.null=FALSE,
timeslice=NULL, condition.on.survival=TRUE, root.type="madfitz", root.p=NULL,
AIC=NULL, verbose=TRUE, n.cores=NULL)
```

Arguments

`phy`	a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.
`data`	a data matrix containing species information (see Details).
`f`	vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be known.
`pars`	vector containing the MLE of the parameters.
`hidden.states`	a logical indicating whether the model includes a hidden state. The default is `FALSE`.
`four.state.null`	a logical indicating whether the model is the null-four model. The default is `FALSE`.
`timeslice`	a user-supplied time to split the tree.
`condition.on.survival`	a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.
`root.type`	indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, “madfitz”, assumed equal, “equal”, or set to user, “user”.
`root.p`	a vector indicating fixed root state probabilities. The default is `NULL`.
`AIC`	the AIC for the model being used for the reconstruction. This is used by the plotting function. The default is `NULL`.
`verbose`	a logical indicating whether progress should be printed to screen. The default is `TRUE`.
`n.cores`	specifies the number of independent processors to conduct the analysis.. The default is `NULL`.

Details

Conducts ancestral state reconstruction on the original `hisse.old` and `hisseNull4.old` functions. In this implementation the marginal probability of state i for a focal node is simply the overall likelihood of the tree and data when the state of the focal node is fixed in state i. Note that the likeliest tip states can also be estimated: we observe state 1, but the underlying state could either be state 1A or 1B. Thus, for any given node or tip we traverse the entire tree as many times as there are states in the model. As the size of the tree grows, however, these repeated tree traversals can slow the calculation down considerably. For this reason, we allow the marginal calculation to be conducted in parallel across any number of independent computer processors.

For user-specified "root.p", you should specify the probability for each state. If you are doing a hidden model, there will be four states: 0A, 1A, 0B, 1B. So if you wanted to say the root had to be state 0, you would specify "root.p = c(0.5, 0, 0.5, 0)".

See help for "hisse.old" for more on other parameters for this function.

Value

`MarginRecon` returns an object of class `hisse.states`. This is a list with elements:

`$node.mat`	the marginal probabilities calculated for each node. They are ordered based on the elements in the edge matrix in "phylo" format.
`$tip.mat`	the marginal probabilities calculated for each tip. They are ordered based on the order of tip labels in the tree.
`$rate.mat`	a matrix that details the rates for each state combination. This is used by the plotting function.
`$phy`	a phylogenetic tree in the "phylo" format that contains the states with the highest marginal probability at each internal node.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters' effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
## Not run
library(diversitree)
pars <- c(0.1, 0.2, 0.03, 0.03, 0.01, 0.01)
set.seed(4)
phy <- tree.bisse(pars, max.t=30, x0=0)
sim.dat <- data.frame(names(phy$tip.state), phy$tip.state)
## Now fit HiSSE equivalent with a hidden state for state 1:
trans.rates.hisse <- TransMatMaker.old(hidden.states=TRUE)
trans.rates.hisse <- ParDrop(trans.rates.hisse, c(2,3,5,7,8,9,10,12))
pp.hisse <- hisse(phy, sim.dat, hidden.states=TRUE, turnover.anc=c(1,2,0,3),
eps.anc=c(1,2,0,3), trans.rate=trans.rates.hisse)
##Now reconstruct the likeliest states under this model:
pp <- MarginRecon.old(phy, sim.dat, f=c(1,1), pars=pp.hisse$solution,
hidden.states=TRUE)
```

MarginReconGeoSSE *Ancestral State Estimation based on Marginal Reconstruction for the GeoSSE and GeoHiSSE models.*

Description

Estimates the likeliest states for both internal nodes and tips of a phylogeny using the marginal reconstruction algorithm.

Usage

```
MarginReconGeoSSE(phy, data, f, pars, hidden.states=1,
assume.cladogenetic=TRUE, condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL, AIC=NULL, get.tips.only=FALSE,
verbose=TRUE, n.cores=NULL, dt.threads=1)
```

Arguments

`phy`	a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.
`data`	a data matrix containing species information (see Details).
`f`	vector of length 3 with the estimated proportion of extant species in area 00, 11, and 01 that are included in the phylogeny. A value of c(0.25, 0.5, 0.25) means that 25 percent of species in state 00 and 01 and 50 percent of species in state 11 are included in the phylogeny. By default all species are assumed to be known.
`pars`	vector containing the MLE of the parameters.
`hidden.states`	a numeric indicating the number of shifts. The default is `1`.
`assume.cladogenetic`	a logical indicating whether the model assumes that changes occurr at the nodes. The default is `TRUE`. Setting this to `FALSE` will result in a different models from GeoSSE. Please check `hisse::GeoHiSSE` for more details.
`condition.on.survival`	a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.
`root.type`	indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, "madfitz", assumed equal, "equal", or set to user, "user".
`root.p`	a vector indicating fixed root state probabilities. The default is `NULL`.
`AIC`	the AIC for the model being used for the reconstruction. This is used by the plotting function. The default is `NULL`.
`get.tips.only`	a logical indicating whether just tip reconstructions should be output. The default is `FALSE`.

`verbose` a logical indicating whether progress should be printed to screen. The default is `TRUE`.

`n.cores` specifies the number of independent processors to conduct the analysis.. The default is `NULL`.

`dt.threads` sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

In this implementation the marginal probability of area i for a focal node is simply the overall likelihood of the tree and data when the area of the focal node is fixed in area i. Note that the likeliest tip areas can also be estimated: we observe area 11, but the underlying area could either be area 11A or 11B. Thus, for any given node or tip we traverse the entire tree as many times as there are areas in the model. As the size of the tree grows, however, these repeated tree traversals can slow the calculation down considerably. For this reason, we allow the marginal calculation to be conducted in parallel across any number of independent computer processors.

For user-specified "root.p", you should specify the probability for each area. If you are doing a hidden model, the number of parameters will depend on the number of hidden states included. For a two classes model there are six areas: 00A, 11A, 01A, 00B, 11B, 01B. So if you wanted to say the root had to be area A, you would specify "root.p = c(0.5, 0, 0, 0.5, 0, 0)". The root area is 00, but there is an equal chance for hidden states A or B.

See help for "GeoHiSSE" for more on other parameters for this function.

Value

`MarginReconGeoSSE` returns an object of class `geohisse.states`. This is a list with elements:

`$node.mat` the marginal probabilities calculated for each node. They are ordered based on the elements in the edge matrix in "phylo" format.

`$tip.mat` the marginal probabilities calculated for each tip. They are ordered based on the order of tip labels in the tree.

`$rate.mat` a matrix that details the rates for each state combination. This is used by the plotting function.

`$phy` a phylogenetic tree in the "phylo" format that contains the states with the highest marginal probability at each internal node.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Goldberg, E. E., L. T. Lancaster, and R. H. Ree. 2011. Phylogenetic Inference of Reciprocal Effects between Geographic Range Evolution and Diversification. Syst. Biol. 60:451-465.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters' effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`MarginReconGeoSSE.old` *Original Ancestral State Estimation based on Marginal Reconstruction for the GeoSSE and GeoHiSSE models.*

Description

Estimates the likeliest states for both internal nodes and tips of a phylogeny using the marginal reconstruction algorithm.

Usage

```
MarginReconGeoSSE.old(phy, data, f, pars, hidden.areas=TRUE,
assume.cladogenetic=TRUE, condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL, AIC=NULL, verbose=TRUE,
n.cores=NULL)
```

Arguments

`phy` a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a data matrix containing species information (see Details).

`f` vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be known.

`pars` vector containing the MLE of the parameters.

`hidden.areas` a logical indicating whether the model includes hidden areas. The default is `FALSE`.

`assume.cladogenetic` a logical indicating whether the model assumes that changes occurr at the nodes. The default is `TRUE`. Setting this to `FALSE` will result in a different models from GeoSSE. Please check `hisse::GeoHiSSE.old` for more details.

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, "madfitz", assumed equal, "equal", or set to user, "user".

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`AIC` the AIC for the model being used for the reconstruction. This is used by the plotting function. The default is `NULL`.

`verbose` a logical indicating whether progress should be printed to screen. The default is `TRUE`.

`n.cores` specifies the number of independent processors to conduct the analysis.. The default is `NULL`.

Details

In this implementation the marginal probability of area i for a focal node is simply the overall likelihood of the tree and data when the area of the focal node is fixed in area i. Note that the likeliest tip areas can also be estimated: we observe area 1, but the underlying area could either be area 1A or 1B. Thus, for any given node or tip we traverse the entire tree as many times as there are areas in the model. As the size of the tree grows, however, these repeated tree traversals can slow the calculation down considerably. For this reason, we allow the marginal calculation to be conducted in parallel across any number of independent computer processors.

For user-specified “root.p”, you should specify the probability for each area. If you are doing a hidden model, the number of parameters will depend on the number of hidden areas included. For a two classes model there are six areas: A0, B0, AB0, A1, B1, AB1. So if you wanted to say the root had to be area A, you would specify “root.p = c(0.5, 0, 0, 0.5, 0, 0)”. There is ’root area is A and there is an equal chance for hidden area A0 or A1’.

See help for “GeoHiSSE.old” for more on other parameters for this function.

Value

`MarginReconGeoSSE.old` returns an object of class `geohisse.states`. This is a list with elements:

`$node.mat` the marginal probabilities calculated for each node. They are ordered based on the elements in the edge matrix in “phylo” format.

`$tip.mat` the marginal probabilities calculated for each tip. They are ordered based on the order of tip labels in the tree.

`$rate.mat` a matrix that details the rates for each state combination. This is used by the plotting function.

`$phy` a phylogenetic tree in the “phylo” format that contains the states with the highest marginal probability at each internal node.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Goldberg, E. E., L. T. Lancaster, and R. H. Ree. 2011. Phylogenetic Inference of Reciprocal Effects between Geographic Range Evolution and Diversification. Syst. Biol. 60:451-465.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters' effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
## Need some examples here.
```

MarginReconHiSSE *Ancestral State Estimation based on Marginal Reconstruction for the HiSSE model.*

Description

Estimates the likeliest states for both internal nodes and tips of a phylogeny using the marginal reconstruction algorithm.

Usage

```
MarginReconHiSSE(phy, data, f, pars, hidden.states=1,
condition.on.survival=TRUE,  root.type="madfitz", root.p=NULL,
AIC=NULL, get.tips.only=FALSE, verbose=TRUE, n.cores=NULL, dt.threads=1)
```

Arguments

`phy` a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a matrix (or dataframe) with three columns. The first column containing the species names and the second and third containing the binary character information. Character "0" is on column 2 and chracter "1" is on column 3. A value of 0 means character absent and a value of 1 character present. The input of data follows a Pagel model. See 'Details'.

`f` vector of length 2 with the estimated proportion of extant species in state 0 and 1 that are included in the phylogeny. A value of c(0.25, 0.5) means that 25 percent of species in state 0 and 50 percent of species in state 1 are included in the phylogeny. By default all species are assumed to be sampled.

`pars` vector containing the MLE of the parameters.

`hidden.states` a numeric indicating the number of shifts. The default is `1`.

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

root.type indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, “madfitz”, assumed equal, “equal”, or set to user, “user”.

root.p a vector indicating fixed root state probabilities. The default is NULL.

AIC the AIC for the model being used for the reconstruction. This is used by the plotting function. The default is NULL.

get.tips.only a logical indicating whether just tip reconstructions should be output. The default is FALSE.

verbose a logical indicating whether progress should be printed to screen. The default is TRUE.

n.cores specifies the number of independent processors to conduct the analysis.. The default is NULL.

dt.threads sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

This is the marginal reconstruction algorithm for the newer, faster version of HiSSE. In this implementation the marginal probability of state i for a focal node is simply the overall likelihood of the tree and data when the state of the focal node is fixed in state i. Note that the likeliest tip states can also be estimated: we observe state 1, but the underlying state could either be state 1A or 1B. Thus, for any given node or tip we traverse the entire tree as many times as there are states in the model. As the size of the tree grows, however, these repeated tree traversals can slow the calculation down considerably. For this reason, we allow the marginal calculation to be conducted in parallel across any number of independent computer processors.

See help for “hisse” for more on other parameters for this function.

Value

MarginReconHiSSE returns an object of class hisse.states. This is a list with elements:

$node.mat the marginal probabilities calculated for each node. They are ordered based on the elements in the edge matrix in “phylo” format.

$tip.mat the marginal probabilities calculated for each tip. They are ordered based on the order of tip labels in the tree.

$rate.mat a matrix that details the rates for each state combination. This is used by the plotting function.

$phy a phylogenetic tree in the “phylo” format that contains the states with the highest marginal probability at each internal node.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters' effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`MarginReconMiSSE` *Ancestral State Estimation based on Marginal Reconstruction for the MiSSE models.*

Description

Estimates the likeliest states for both internal nodes and tips of a phylogeny using the marginal reconstruction algorithm.

Usage

```
MarginReconMiSSE(phy, f, pars, hidden.states=1, fixed.eps=NULL,
condition.on.survival=TRUE,  root.type="madfitz", root.p=NULL,
AIC=NULL, get.tips.only=FALSE, verbose=TRUE, n.cores=NULL, dt.threads=1)
```

Arguments

`phy` a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.

`f` the estimated proportion of extant species included in the phylogeny. A value of 0.50 means that 50 percent of species are contained in the. By default all species are assumed to be sampled.

`pars` vector containing the MLE of the parameters.

`hidden.states` a numeric indicating the number of shifts. The default is 2.

`fixed.eps` a value to be used to fix extinction fraction during search. Default is `NULL` meaning that it is freely estimated.

`condition.on.survival`
a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, "madfitz", assumed equal, "equal", or set to user, "user".

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`AIC` the AIC for the model being used for the reconstruction. This is used by the plotting function. The default is `NULL`.

`verbose` a logical indicating whether progress should be printed to screen. The default is `TRUE`.

`get.tips.only` a logical indicating whether just tip reconstructions should be output. The default is `FALSE`.

`n.cores` specifies the number of independent processors to conduct the analysis. The default is `NULL`.

`dt.threads` sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

In this implementation the marginal probability of state combination i for a focal node is simply the overall likelihood of the tree and data when the state of the focal node is fixed in state combination i.

See help for “MiSSE” for more on other parameters for this function.

Value

`MarginReconMiSSE` returns an object of class `misse.states`. This is a list with elements:

`$node.mat` the marginal probabilities calculated for each node. They are ordered based on the elements in the edge matrix in “phylo” format.

`$tip.mat` the marginal probabilities calculated for each tip. They are ordered based on the order of tip labels in the tree.

`$rate.mat` a matrix that details the rates for each state combination. This is used by the plotting function.

`$phy` a phylogenetic tree in the “phylo” format that contains the states with the highest marginal probability at each internal node.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters’ effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

MarginReconMuHiSSE *Ancestral State Estimation based on Marginal Reconstruction for the MuSSE and MuHiSSE models.*

Description

Estimates the likeliest states for both internal nodes and tips of a phylogeny using the marginal reconstruction algorithm.

Usage

```
MarginReconMuHiSSE(phy, data, f, pars, hidden.states=1,
condition.on.survival=TRUE,  root.type="madfitz", root.p=NULL,
AIC=NULL, get.tips.only=FALSE, verbose=TRUE, n.cores=NULL, dt.threads=1)
```

Arguments

`phy` a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a matrix (or dataframe) with three columns. The first column containing the species names and the second and third containing the binary character information. Character "0" is on column 2 and chracter "1" is on column 3. A value of 0 means character absent and a value of 1 character present. The input of data follows a Pagel model. See ’Details’.

`f` vector of length 4 with the estimated proportion of extant species in 00, 01, 10, and 11 that are included in the phylogeny. A value of c(0.50, 0.25, 0.125, 0.125) means that 50 percent of species in combination ’00’, 25 percent in ’01’ and 12.5 percent in ’10’ and ’11’. By default all species are assumed to be sampled.

`pars` vector containing the MLE of the parameters.

`hidden.states` a numeric indicating the number of shifts. The default is `2`.

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, “madfitz”, assumed equal, “equal”, or set to user, “user”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`AIC` the AIC for the model being used for the reconstruction. This is used by the plotting function. The default is `NULL`.

`get.tips.only` a logical indicating whether just tip reconstructions should be output. The default is `FALSE`.

`verbose` a logical indicating whether progress should be printed to screen. The default is `TRUE`.

`n.cores`	specifies the number of independent processors to conduct the analysis.. The default is `NULL`.
`dt.threads`	sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

In this implementation the marginal probability of state combination i for a focal node is simply the overall likelihood of the tree and data when the state of the focal node is fixed in state combination i. Note that the likeliest tip states can also be estimated: we observe state 00, but the underlying state could either be 00A or 00B. Thus, for any given node or tip we traverse the entire tree as many times as there are states in the model. As the size of the tree grows, however, these repeated tree traversals can slow the calculation down considerably. For this reason, we allow the marginal calculation to be conducted in parallel across any number of independent computer processors.

For user-specified “root.p”, you should specify the probability for each state combination. If you are doing a hidden model, the number of parameters will depend on the number of hidden states included. For a two classes model there are eight states: 00A, 01A, 10A, 11A, 00B, 01B, 10B, and 11B. So if you wanted to say the root had to be in state 00, you would specify “root.p = c(0.5, 0, 0, 0, 0.5, 0, 0, 0, 0)”. There is 50 percent chance the root state is 00 and there is an equal chance for hidden state A or B.

See help for “MuHiSSE” for more on other parameters for this function.

Value

`MarginRecon` returns an object of class `muhisse.states`. This is a list with elements:

`$node.mat`	the marginal probabilities calculated for each node. They are ordered based on the elements in the edge matrix in “phylo” format.
`$tip.mat`	the marginal probabilities calculated for each tip. They are ordered based on the order of tip labels in the tree.
`$rate.mat`	a matrix that details the rates for each state combination. This is used by the plotting function.
`$phy`	a phylogenetic tree in the “phylo” format that contains the states with the highest marginal probability at each internal node.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters’ effect on speciation and extinction. Syst. Biol. 56:701-710.

Nakov, T., Beaulieu, J.M., and Alverson, A.J. 2019. Diatoms diversify and turn over faster in freshwater than marine environments. Evolution, doi: https://doi.org/10.1111/evo.13832.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

MiSSE *Character-free State Speciation and Extinction*

Description

Sets up and executes a MiSSE model (Missing State Speciation and Extinction) on a phylogeny.

Usage

```
MiSSE(phy, f=1, turnover=c(1,2), eps=c(1,2), fixed.eps=NULL, condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL, sann=TRUE, sann.its=1000,
bounded.search=TRUE,  max.tol=.Machine$double.eps^.50, starting.vals=NULL,
turnover.upper=10000, eps.upper=3, trans.upper=100, restart.obj=NULL,
ode.eps=0, dt.threads=1,expand.mode=FALSE)
```

Arguments

`phy` a phylogenetic tree, in ape “phylo” format and with internal nodes labeled denoting the ancestral selective regimes.

`f` the estimated proportion of extant species included in the phylogeny. A value of 0.50 means that 50 percent of species are contained in the. By default all species are assumed to be sampled.

`turnover` a numeric vector of length equal to the number of suspected rates in turnover. See ’Details’.

`eps` a numeric vector of length equal to the number of suspected rates in extinction fraction. See ’Details’.

`fixed.eps` a value to be used to fix extinction fraction during search. Default is `NULL` meaning that it is freely estimated.

`condition.on.survival` a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`sann` a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `FALSE`.

`sann.its` a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search`	a logical indicating whether or not bounds should be enforced during optimization. The default is `TRUE`.
`max.tol`	supplies the relative optimization tolerance to `subplex`.
`starting.vals`	a numeric vector of length 3 with starting values for the model. Position [1] sets turnover, [2] sets extinction fraction, and [3] transition rates between distinct diversification rates.
`turnover.upper`	sets the upper bound for the turnover parameters.
`eps.upper`	sets the upper bound for the eps parameters.
`trans.upper`	sets the upper bound for the transition rate parameters.
`restart.obj`	an object of class that contains everything to restart an optimization.
`ode.eps`	sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.
`dt.threads`	sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.
`expand.mode`	allows passing in the number of free parameters for turnover and eps and creates vectors to permit this.

Details

One thing pointed out in the original HiSSE paper (Beaulieu & O'Meara, 2016) is that the trait-independent hisse model is basically a model for traits and a separate model for shifts in diversification parameters, much like BAMM (though without priors, discontinuous inheritance of extinction probability, or other mathematical foibles). The hidden states can drive different diversification processes, and the traits just evolve in a regular trait model. At that point, there is no harm in just dropping the trait (or analyzing separately) and just focusing on diversification driven by unknown factors. That is what this function does. It sets up and executes a completely trait-free version of a HiSSE model.

Thus, all that is required is a tree. The model allows up to 26 possible hidden states in diversification (denoted by A-Z). Transitions among hidden states are governed by a global transition rate, q. A "shift" in diversification denotes a lineage tracking some unobserved, hidden state. An interesting byproduct of this assumption is that distantly related clades can actually share the same discrete set of diversification parameters.

Note that "hidden state" is a shorthand. We do not mean that there is a single, discrete character that is solely driving diversification differences. There is some heritable "thing" that affects rates: but this could be a combination of body size, oxygen concentration, trophic level, and how many other species are competing in an area. This is true for HiSSE, but is especially important to grasp for MiSSE. It could be that there is some single discrete trait that drives everything; it's more likely that a whole range of factors play a role, and we just slice them up into discrete categories, the same way we slice up mammals into carnivore / omnivore / herbivore or plants into woody / herbaceous when the reality is more continuous.

As with `hisse`, we employ a modified optimization procedure. In other words, rather than optimizing birth and death separately, `MiSSE` optimizes orthogonal transformations of these variables:

we let tau = birth+death define "net turnover", and we let eps = death/birth define the “extinction fraction”. This reparameterization alleviates problems associated with overfitting when birth and death are highly correlated, but both matter in explaining the diversity pattern.

For the “root.type” option, we are currently maintaining the previous default of “madfitz”. However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing “herr_als”. It is unclear to us which is exactly correct, but it does seem that both “madfitz” and “herr_als” behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We’ve also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the “madfitz” conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

Also, note, that in the case of “root.type=user” and “root.type=equal” are no longer explicit “root.type” options. Instead, either “madfitz” or “herr_als” are specified and the “root.p” can be set to allow for custom root options.

Value

`MiSSE` returns an object of class `misse.fit`. This is a list with elements:

`$loglik` the maximum negative log-likelihood.

`$AIC` Akaike information criterion.

`$AICc` Akaike information criterion corrected for sample-size.

`$solution` a matrix containing the maximum likelihood estimates of the model parameters.

`$index.par` an index matrix of the parameters being estimated.

`$f` user-supplied sampling frequencies.

`$hidden.states` a logical indicating whether hidden states were included in the model.

`$condition.on.surivival`
a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$phy` user-supplied tree

`$max.tol` relative optimization tolerance.

`$starting.vals` The starting values for the optimization.

`$upper.bounds` the vector of upper limits to the optimization search.

`$lower.bounds` the vector of lower limits to the optimization search.

`$ode.eps` The ode.eps value used for the estimation.

`$turnover` The turnover vector used.

`$eps` The eps vector used.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Herrera-Alsina, L., P. van Els, and R.S. Etienne. 2018. Detecting the dependence of diversification on multiples traits from phylogenetic trees and trait data. Systematic Biology, 68:317-328.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`MiSSEGreedy` *Character-free State Speciation and Extinction Searching Greedily*

Description

Executes a set of MiSSE models (Missing State Speciation and Extinction) on a phylogeny, varying the number of parameters for turnover and extinction fraction and stopping when models stop being very good.

Usage

```
MiSSEGreedy(phy, f=1, possible.combos = generateMiSSEGreedyCombinations(),
stop.deltaAICc=10, save.file=NULL, n.cores=NULL, chunk.size=NULL,
condition.on.survival=TRUE, root.type="madfitz", root.p=NULL, sann=TRUE,
sann.its=10000, bounded.search=TRUE, max.tol=.Machine$double.eps^.50, starting.vals=NULL,
turnover.upper=10000, eps.upper=3, trans.upper=100, restart.obj=NULL, ode.eps=0)
```

Arguments

`phy` a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.

`f` the estimated proportion of extant species included in the phylogeny. A value of 0.50 means that 50 percent of species are contained in the. By default all species are assumed to be sampled.

`possible.combos`
data.frame of parameter combinations to try. See 'Details'.

`stop.deltaAICc` how bad compared to the best does a model have to be to far enough outside we stop looking at even more complex ones?

`save.file` file to use to save results while the code is running

`n.cores` how many cores to run this on in parallel

`chunk.size` how many models to run before checking to make sure there are no improvements. See 'Details'.

condition.on.survival
: a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is TRUE.

root.type
: indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, “madfitz” or Herrera-Alsina et al. 2018, “herr_als”.

root.p
: a vector indicating fixed root state probabilities. The default is NULL.

sann
: a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using subplex. The default is TRUE.

sann.its
: a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

bounded.search
: a logical indicating whether or not bounds should be enforced during optimization. The default is TRUE.

max.tol
: supplies the relative optimization tolerance to subplex.

starting.vals
: a numeric vector of length 3 with starting values for the model. Position [1] sets turnover, [2] sets extinction fraction, and [3] transition rates between distinct diversification rates.

turnover.upper
: sets the upper bound for the turnover parameters.

eps.upper
: sets the upper bound for the eps parameters.

trans.upper
: sets the upper bound for the transition rate parameters.

restart.obj
: an object of class that contains everything to restart an optimization.

ode.eps
: sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

Details

See the MiSSE function for description of the method overall. It requires a set number of hidden state categories, but finding the best number of categories can be hard. For example, one could 1 to 26 different turnover categories and 1 to 26 possible extinction fraction categories. For most cases, we suspect that it makes sense to have the number of extinction fraction categories either equal to the number of turnover categories or set to the same category over the tree, but there are actually a lot of possibilities: have turnover=c(1,2,3) and eps=c(1,2,1), for example, or turnover=c(1,1,1) and eps=c(1,2,3). This uses the generateMiSSEGreedyCombinations function to generate a very large set of these possible models, then runs them in increasing complexity. By default, it stops when the models stop getting being reasonable in AICc. This is NOT where the models stop being significant (if you’re looking for significance, note you don’t get that from AICc), but where they probably will not contribute much to the model averaged parameter estimates and so may not be worth the bother of searching further. However, there’s no guarantee that this is a wise decision: you could stop with 10 turnover rates, and 11 and 12 are far worse for AICc, but it could be that 13 turnover rates are much better than anything else (for that matter, the best number of turnover rates could be 42, even though MiSSe’s current code can only go up to 26). And of course, in reality, the truth is that there is an infinite set of hidden rate parameters: a passing cloud blocks a tiny bit of energy for photosynthesis, so for that moment the rate of extinction for plants underneath the shadow is a tiny bit higher. You should not be using this to test hypotheses about how many hidden factors there are, but rather as a way to get a good enough model to estimate rates on a tree.

You can change how quickly the function stops trying new models with stop.deltaAICc. Once it runs a chunk of models where the best gets a model that is at least stop.deltaAICc worse than the *current* best model, it stops running new models. Since this is based on current best AICc, and we start with simple models, there's an asymmetry: a terrible model with no rate variation is always included, but a slightly less terrible model with 26 turnover rates might never be run.

This works MUCH faster when run in parallel. To do so, set n.cores to the number of available cores on your machine (for example, for a modern Mac laptop, this would be 4). An easy way to do this automatically is to use n.cores=parallel::detectCores(). The default is one core; if running this function on a cluster, a default of parallel::detectCores() might take over an entire compute node when you're supposed to be using just one core, and get you in trouble. Since we run many models, the most natural approach is to run one model per core, see if at least one of the models are still ok, then send out the next models out to all the cores. Setting n.cores to the number of parallel jobs you want, and leaving chunk.size set to NULL, will do this. However, this is slightly inefficient – the odds are that some cores will finish earlier than others, and will be waiting until all finish. So a different approach is to set a chunk.size greater than n.cores – it will still use no more than n.cores at a time, but once one model in the set finishes it will send off the next until all models in the chunk are run. This keeps the computer even busier, but then it won't stop to check to make sure the models are still feasible as often, and it only saves intermediate results after each chunk of models finishes. Our recommendation is to use n.cores=parallel::detectCores() if you're on a machine where you can use all the cores and leave chunk.size unspecified (so it will also default to n.cores), but it's up to you.

After every chunk of models are done, this function will display the status: what models have been run, what the likelihoods and AICs are, etc. It will also predict how long future runs will check (based on a linear regression between the number of free parameters and log(minutes to run)). These are just estimates based on the runs so far, but it's a stochastic search and can take more or less time.

Saving output while this goes is highly recommended – you can see how things are going and salvage something if a run fails (then, start again, deleting the models that worked from possible.combos and USING A DIFFERENT FILE TO SAVE TO SO YOU DON'T OVERWRITE THE OLD ONE. You can so this by giving a file name (including path, if you want) as the save.file argument. This will save the list of finished models (misse.list) and the possible.combos data.frame with additional information.

Value

`MiSSEGreedy` returns a list of class `misse.fit` objects.

Author(s)

Brian C. O'Meara

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Herrera-Alsina, L., P. van Els, and R.S. Etienne. 2018. Detecting the dependence of diversification on multiples traits from phylogenetic trees and trait data. Systematic Biology, 68:317-328.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

Examples

```
library(ape)
phy <- rcoal(50)
potential.combos <- generateMiSSEGreedyCombinations(max.param=4, vary.both=TRUE)

#run on one core:
result <- MiSSEGreedy(phy, potential.combos, n.cores=1, f=1)

#run on all available cores: (commented out because CRAN does not allow this in examples)
#result <- MiSSEGreedy(phy, potential.combos, n.cores=parallel::detectCores(), f=1)



AICc <- unlist(lapply(result, "[[", "AICc"))
deltaAICc <- AICc-min(AICc)
print(length(result))

# Yes, [[ is a function you can lapply, and the name of elements within each list
# object can be arguments. Life is awesome.
```

`Model averaged rates` *Model average rates at tips and nodes*

Description

Summarizes reconstructions from a single model or a set of models and returns model averaged rates.

Usage

```
GetModelAveRates(x, AIC.weights=NULL, type=c("tips", "nodes", "both"),
bound.par.matrix=cbind(c(0,-1000,0,0,0),c(1000,1000,1000,3,1000)) )
```

Arguments

- `x` a `hisse.states` object, a `hisse.states.geosse` object, a `muhisse.states` or a list of such objects. A list of model can only include models of one type.
- `AIC.weights` a numeric vector with length equal to the number of models in the list 'x'. This is the AICw for each of the models to be averaged. If 'AIC.weights==NULL' (the default value) then function will compute the AICw for the models in the set from the $AIC value for each model.
- `type` one of "tips", "nodes", or "both" (the default). This controls whether only the averaged rates for the tips, nodes or both should be returned. If tips or nodes is chosen the output will be a matrix and if both are returned the output will be a list.
- `bound.par.matrix` A numeric matrix with exactly 5 rows and 2 columns to set the limits for the model parameters. First column is the minimum value and second column is the maximum value (both inclusive). The rows are the model parameter in this fixed order: turnover, net diversification, speciation, extinction fraction, and extinction.

Details

Provides a data frame model-averaged rates for all possible configurations of the model parameters (i.e., turnover, net.div, speciation, extinction, or extinction.fraction), either for all tips or for all nodes. As with the plotting function, if you give a single hisse.state object, it uses that, and the rates account for uncertainty in the marginal probabilities of the states; if you give it a list of them, it will model-average the results (it assumes the trees are the same) in addition to accounting for uncertainty in the marginal probabilities at tips and nodes. If 'AIC.weights==NULL', then the models in the set do not require the '$AIC' element to compute the AICw. Otherwise the function will return an error message.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

`MuHiSSE` *Multicharacter Hidden State Speciation and Extinction*

Description

Sets up and executes a MuHiSSE model (Multicharacter Hidden State Speciation and Extinction) on a phylogeny and character distribution.

Usage

```
MuHiSSE(phy, data, f=c(1,1,1,1), turnover=c(1,2,3,4), eps=c(1,2,3,4),
hidden.states=FALSE, trans.rate=NULL, condition.on.survival=TRUE,
root.type="madfitz", root.p=NULL, sann=TRUE, sann.its=1000,
bounded.search=TRUE,  max.tol=.Machine$double.eps^.50, starting.vals=NULL,
turnover.upper=10000, eps.upper=3, trans.upper=100, restart.obj=NULL,
ode.eps=0, dt.threads=1)
```

Arguments

`phy` a phylogenetic tree, in ape "phylo" format and with internal nodes labeled denoting the ancestral selective regimes.

`data` a matrix (or dataframe) with three columns. The first column containing the species names and the second and third containing the binary character information. Character "0" is on column 2 and chracter "1" is on column 3. A value of 0 means character absent and a value of 1 character present. The input of data follows a Pagel model. See 'Details'.

`f` vector of length 4 with the estimated proportion of extant species in 00, 01, 10, and 11 that are included in the phylogeny. A value of c(0.50, 0.25, 0.125, 0.125) means that 50 percent of species in combination '00', 25 percent in '01' and 12.5 percent in '10' and '11'. By default all species are assumed to be sampled.

`turnover` a numeric vector of length equal to 4+(number of `hidden.states * 4`). A MuSSE model has 4 speciation parameters: lambda00, lambda01, lambda10, and lambda11. A MuHiSSE model with one hidden states has 8 speciation parameters: lambda0A, s1A, s01A, s0B, s1B, s01B. And so on. The length of the numeric vector needs to match the number of speciation parameters in the model. See 'Details'.

`eps` a numeric vector of length equal to4+(number of `hidden.states * 4`). A MuSSE model has 4 extinction parameters: mu00, mu01, mu10, and mu11. A MuHiSSE model with one hidden state has 8 extinction parameters: mu00A, mu01A, mu10A, mu11A, mu00B, mu01B, mu10B, and mu11B. And so on. The length of the numeric vector needs to match the number of extinction parameters in the model. See 'Details'.

`hidden.states` a logical indicating whether the model includes a hidden states. The default is `FALSE`.

`trans.rate` provides the transition rate model. See function `TransMatMakerMuHiSSE`.

`condition.on.survival`
a logical indicating whether the likelihood should be conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root summarization follow the procedure described by FitzJohn et al. 2009, "madfitz" or Herrera-Alsina et al. 2018, "herr_als".

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`sann` a logical indicating whether a two-step optimization procedure is to be used. The first includes a simulate annealing approach, with the second involving a refinement using `subplex`. The default is `FALSE`.

`sann.its` a numeric indicating the number of times the simulated annealing algorithm should call the objective function.

`bounded.search` a logical indicating whether or not bounds should be enforced during optimization. The default is is `TRUE`.

`max.tol` supplies the relative optimization tolerance to `subplex`.

`starting.vals` a numeric vector of length 3 with starting values for the model for all areas and hidden states. Position [1] sets turnover, [2] sets extinction fraction, and [3] transition rates.

`turnover.upper` sets the upper bound for the turnover parameters.

`eps.upper` sets the upper bound for the eps parameters.

`trans.upper` sets the upper bound for the transition rate parameters.

`restart.obj` an object of class that contains everything to restart an optimization.

`ode.eps` sets the tolerance for the integration at the end of a branch. Essentially if the sum of compD is less than this tolerance, then it assumes the results are unstable and discards them. The default is set to zero, but in testing a value of 1e-8 can sometimes produce stable solutions for both easy and very difficult optimization problems.

`dt.threads` sets the number of threads available to data.table. In practice this need not change from the default of 1 thread, as we have not seen any speedup from allowing more threads.

Details

This function sets up and executes a multiple state HiSSE model. The model allows up to 8 hidden categories (hidden states A-H), and implements a more efficient means of carrying out the branch calculation. Specifically, we break up the tree into carry out all descendent branch calculations simultaneously, combine the probabilities based on their shared ancestry, then repeat for the next set of descendents. In testing, we've found that as the number of taxa increases, the calculation becomes much more efficient. In future versions, we will likely allow for multicore processing of these calculations to further improve speed. We also note that there is vignette tha describes more details for running this particular function.

As for data file format, `MuHiSSE` expects a three column matrix or data frame, with the first column containing the species names and the second and third containing the binary character information. Note that the order of the data file and the names in the “phylo” object need not be in the same order; `MuHiSSE` deals with this internally. Also, the character information must be coded as 0 and 1, otherwise, the function will misbehave. However, if the state for a species is unknown for either character, a user can specify this with a 2, and the state will be considered maximally ambiguous for all relevant character combinations. For example, if character 1 is in state 0, but character 2 is provided a 2, then the program provides a probability of 1 for 00 and a probability of for 01.

As with `hisse`, we employ a modified optimization procedure. In other words, rather than optimizing birth and death separately, `MuHisse` optimizes orthogonal transformations of these variables: we let tau = birth+death define "net turnover", and we let eps = death/birth define the “extinction fraction”. This reparameterization alleviates problems associated with overfitting when birth and death are highly correlated, but both matter in explaining the diversity pattern.

To setup a model, users input vectors containing values to indicate how many free parameters are to be estimated for each of the variables in the model. This is done using the `turnover` and `extinct.frac` parameters. One needs to specify a value for each of the parameters of the model, when two parameters show the same value, then the parameters are set to be linked during the estimation of the model. For example, a MuHiSSE model with 1 hidden state and all free parameters has `turnover = 1:8`. The same model with turnover rates constrained to be the same for all hidden states has `turnover = c(1,2,3,4,1,2,3,4)`. This same format applies to `extinct.frac`.

The “trans.rate” input is the transition model and has an entirely different setup than speciation and extinction rates. See `TransMatMakerMuHiSSE` function for more details.

For user-specified “root.p”, you should specify the probability for each state combination. If you are doing a hidden model, there will be eight state combinations: 00A, 01A, 10A, 11A, 00B, 01B, 10B, 11B. So if you wanted to say the root had to be in state 00, and since you do not know the hidden state, you would specify “root.p = c(0.5, 0, 0, 0, 0.5, 0, 0, 0)”. In other words, the root has a 50% chance to be in one of the states to be 00A or 00B.

For the “root.type” option, we are currently maintaining the previous default of “madfitz”. However, it was recently pointed out by Herrera-Alsina et al. (2018) that at the root, the individual likelihoods for each possible state should be conditioned prior to averaging the individual likelihoods across states. This can be set doing “herr_als”. It is unclear to us which is exactly correct, but it does seem that both “madfitz” and “herr_als” behave exactly as they should in the case of character-independent diversification (i.e., reduces to likelihood of tree + likelihood of trait model). We’ve also tested the behavior and the likelihood differences are very subtle and the parameter estimates in simulation are nearly indistinguishable from the “madfitz” conditioning scheme. We provide both options and encourage users to try both and let us know conditions in which the result vary dramatically under the two root implementations. We suspect they do not.

Also, note, that in the case of “root.type=user” and “root.type=equal” are no longer explicit “root.type” options. Instead, either “madfitz” or “herr_als” are specified and the “root.p” can be set to allow for custom root options.

Value

`MuHiSSE` returns an object of class `muhisse.fit`. This is a list with elements:

`$loglik` the maximum negative log-likelihood.

`$AIC` Akaike information criterion.

`$AICc` Akaike information criterion corrected for sample-size.

`$solution` a matrix containing the maximum likelihood estimates of the model parameters.

`$index.par` an index matrix of the parameters being estimated.

`$f` user-supplied sampling frequencies.

`$hidden.states` a logical indicating whether hidden states were included in the model.

`$condition.on.surivival`
a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them.

`$root.type` indicates the user-specified root prior assumption.

`$root.p` indicates whether the user-specified fixed root probabilities.

`$phy` user-supplied tree

`$data` user-supplied dataset

`$trans.matrix`	the user-supplied transition matrix
`$max.tol`	relative optimization tolerance.
`$starting.vals`	The starting values for the optimization.
`$upper.bounds`	the vector of upper limits to the optimization search.
`$lower.bounds`	the vector of lower limits to the optimization search.
`$ode.eps`	The ode.eps value used for the estimation.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Herrera-Alsina, L., P. van Els, and R.S. Etienne. 2018. Detecting the dependence of diversification on multiples traits from phylogenetic trees and trait data. Systematic Biology, 68:317-328.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nakov, T., Beaulieu, J.M., and Alverson, A.J. 2019. Diatoms diversify and turn over faster in freshwater than marine environments. Evolution, doi: https://doi.org/10.1111/evo.13832.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`plot.geohisse.states` *Plotting function geohisse.states objects*

Description

A plotting function for visualizing changes in states and rates over a phylogeny

Usage

```
## S3 method for class 'geohisse.states'
plot(x, rate.param = "net.div", type = "fan",
show.tip.label = TRUE, fsize = 1.0, legend = TRUE, ...)
```

Arguments

`x`	a `hisse.states` object or a list of such objects.
`rate.param`	indicates the type of rates to plot. Options include: “turnover”, “net.div”, “speciation”, “extinction”, “extinction.fraction”.
`type`	a character string specifying the type of phylogeny to be drawn. Options are "fan" (default) or "phylogram".
`show.tip.label`	a logical indicating whether tip names should be included. Default is TRUE
`fsize`	sets the font size for the tip labels.
`legend`	indicates if the legend is to be plotted. TRUE or FALSE.
`...`	Additional parameters to control the plot. See “Details”.

Details

Additional parameters can be defined using “...”:

“do.observed.only” is a logical indicating whether just the states should be plotted; for now, only TRUE works. “rate.colors” are user specified colors to be used for coloring rates. “state.colors” are user specified colors to be used for coloring states. A vector with the color for state 1 (endemic area ’0’), state 2 (endemic area ’1’) and state 0 (widespread area ’01’). Always in this order! “edge.width” is the width of the branches of the phylogeny. “width.factor” is the factor multiplying the “edge.width” to get the width of the branch for the states tree. Needs to be a numeric value between 0 and 1. “rate.range” is an optional two element vector. If present, specifies the range of rate values to use for plotting. “lims.percentage.correction” deals with cases where the limits are slightly smaller than the values due to imprecision issues. “legend.position” are the coordinates for placing the legend. “legend.cex” is the text size inside the legend. “legend.kernel” lets you specify the way the density plot or histogram is made for rates. A value of “auto” chooses what we think is the best option given your data, “hist” makes a histogram, “rectangular”, “gaussian”, and others make a density plot. See `?density` for all non-“hist” options. “legend.bg” sets the color for the legend background. “mar” sets the margins for the plot. See more information in ’help(par)’. “outline” is whether to plot an outline on the branches of the tree. Choose the color of the outline using the parameter outline.color. “outline.color” is the color for the outline. Defaults to "black". “swap.underscore” sets whether to substitute all "_" with " " on the labels for the tips.

This function is very similar to the `hisse::plot.hisse.states` function. See more details in help page for `hisse::plot.hisse.states` function.

Author(s)

Brian O’Meara and Daniel Caetano

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

`plot.hisse.states` *Plotting function hisse.states objects*

Description

A plotting function for visualizing changes in states and rates over a phylogeny

Usage

```
## S3 method for class 'hisse.states'
plot(x, rate.param = "net.div", type = "fan",
show.tip.label = TRUE, fsize = 1.0, legend = "tips", ...)
```

Arguments

`x` a `hisse.states` object or a list of such objects.

`rate.param` indicates the type of rates to plot. Options include: “turnover”, “net.div”, “speciation”, “extinction”, “extinction.fraction”. The default is “net.div”.

`type` a character string specifying the type of phylogeny to be drawn. Options are "fan" (default) or "phylogram".

`show.tip.label` a logical indicating whether tip names should be included.

`fsize` sets the font size for the tip labels.

`legend` indicates the type of legend. Options include: “none”, “traditional”, “tips”, “internal”, “all”.

`...` Additional parameters to control the plot. See “Details”.

Details

Provides phylogeny that shows a heat map of the diversification rate parameter you specify (which could be turnover, net.div, speciation, extinction, or extinction.fraction). The discrete state reconstruction appears as lines on top of the heat map. If you give a single hisse.state object, it uses that; if you give it a list of them, it will model-average the results (it assumes the trees are the same). Colors can be specified by sending a vector of colors to rate.colors or state.colors (the defaults are red to blue for rate and white to black for state). You can specify two or more colors: c("red", "gray", "blue") for example. By default the visualization uses the minimum rate on the tree for the minimum color, and the maximum rate for the maximum color, but you may want to use the same color scale across models, even if some of them have a smaller range than others. To do this, pass a vector with the minimum and maximum rate across all models to the visualization for all models and they will use the same scale. There are many options for adding a legend. A traditional legend showing what values a color corresponds to is “traditional”, like what plotSimmap will show in phytools. However, we can also use the legend to show the distribution of values, rather than just a key to color. “tips” shows a density plot of states or rates at tips, “internal” a distribution at internal nodes, and “all” at all nodes in the tree. For the density or histogram plots, you can let the package pick the best visualization or choose yourself whether to use a histogram or density plot, and if the latter, what kernel you want. The legend can be moved around the overall tree plot by using “legend.position”: this is a vector that specifies the “fig” argument to “par”: c(x1, x2, y1, y2), where the values are the starting and ending positions as a fraction of the overall plot. By default, the legend starts at the lower left corner and continues up 20 the rest of the plot (c(0, 0.2, 0, 0.2)): by changing values, you can make the legend larger or smaller and change its position. The heatmap code is modified slightly from Liam Revell’s phytools package.

Additional parameters to control the plot: “do.observed.only” is a logical indicating whether just the states should be plotted; for now, only TRUE works. “rate.colors” sets user specified colors to be used for coloring rates. “state.colors” sets user specified colors to be used for coloring states. “edge.width” sets the width of the branches of the phylogeny. “width.factor” is the factor multiplying the edge.width to get the width of the branch for the states tree. Needs to be a numeric value between 0 and 1. “rate.range” is an optional two element vector. If present, specifies the range of rate values to use for plotting. “lims.percentage.correction” deals with cases where the limits are slightly smaller than the values due to imprecision issues. “legend.position” are the coordinates for placing the legend. “legend.cex” is the text size inside the legend. “legend.kernel.rates” used for legend=tips, internal, or all, lets you specify the way the density plot or histogram is made for rates. “auto” chooses what we think is the best option given your data, “hist” makes a histogram, “rectangular”, “gaussian”, and others make a density plot. See `?density` for all non-“hist” options. “legend.kernel.states” is as above, for states. “legend.bg” sets the color for the legend background. “mar” sets the margins for the plot. See more information in ’help(par)’. “outline” sets whether to plot an outline on the branches of the tree. Choose the color of the outline using the parameter outline.color. “outline.color” sets the color for the outline. Defaults to "black". “swap.underscore” defines whether to substitute all "_" with " " on the labels for the tips.

Author(s)

Brian O’Meara and Daniel Caetano

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

`plot.misse.states` *Plotting function geohisse.states objects*

Description

A plotting function for visualizing changes in states and rates over a phylogeny

Usage

```
## S3 method for class 'misse.states'
plot(x, rate.param = "net.div", type = "fan",
show.tip.label = TRUE, fsize = 1.0, legend = "tips", ...)
```

Arguments

`x` a `misse.states` object or a list of such objects.

`rate.param` indicates the type of rates to plot. Options include: “turnover”, “net.div”, “speciation”, “extinction”, “extinction.fraction”.

`type` a character string specifying the type of phylogeny to be drawn. Options are "fan" (default) or "phylogram".

`show.tip.label` a logical indicating whether tip names should be included. Default is TRUE

`fsize` sets the font size for the tip labels.

`legend` indicates if the legend is to be plotted. TRUE or FALSE.

`...` Additional parameters to control the plot. See “Details”.

Details

Additional parameters can be defined using “...”:

“do.observed.only” is a logical indicating whether just the states should be plotted; for now, only TRUE works. “rate.colors” are user specified colors to be used for coloring rates. “state.colors” are user specified colors to be used for coloring states. A vector with the color for states 00, 01, 10, and 11 (in this order). “edge.width” is the width of the branches of the phylogeny. “width.factor” is the factor multiplying the “edge.width” to get the width of the branch for the states tree. Needs to be a numeric value between 0 and 1. “rate.range” is an optional two element vector. If present, specifies the range of rate values to use for plotting. “lims.percentage.correction” deals with cases where the limits are slightly smaller than the values due to imprecision issues. “legend.position” are the coordinates for placing the legend. “legend.cex” is the text size inside the legend. “legend.kernel” lets you specify the way the density plot or histogram is made for rates. A value of “auto” chooses what we think is the best option given your data, “hist” makes a histogram, “rectangular”, “gaussian”, and others make a density plot. See `?density` for all non-“hist” options. “legend.bg” sets the color for the legend background. “mar” sets the margins for the plot. See more information in ’help(par)’. “outline” is whether to plot an outline on the branches of the tree. Choose the color of the outline using the parameter outline.color. “outline.color” is the color for the outline. Defaults to "black". “swap.underscore” sets whether to substitute all "_" with " " on the labels for the tips.

This function is very similar to the `hisse::plot.hisse.states` function. See more details in help page for `hisse::plot.hisse.states` function.

Author(s)

Brian O’Meara and Daniel Caetano

References

Beaulieu, J.M, and B.C. O’Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

plot.muhisse.states *Plotting function muhisse.states objects*

Description

A plotting function for visualizing changes in states and rates over a phylogeny

Usage

```
## S3 method for class 'muhisse.states'
plot(x, rate.param = "net.div", type = "fan",
show.tip.label = TRUE, fsize = 1.0, legend = TRUE, ...)
```

Arguments

`x`	a `muhisse.states` object or a list of such objects.
`rate.param`	indicates the type of rates to plot. Options include: “turnover”, “net.div”, “speciation”, “extinction”, “extinction.fraction”.
`type`	a character string specifying the type of phylogeny to be drawn. Options are "fan" (default) or "phylogram".
`show.tip.label`	a logical indicating whether tip names should be included. Default is TRUE
`fsize`	sets the font size for the tip labels.
`legend`	indicates if the legend is to be plotted. TRUE or FALSE.
`...`	Additional parameters to control the plot. See “Details”.

Details

Additional parameters can be defined using “...”:

“do.observed.only” is a logical indicating whether just the states should be plotted; for now, only TRUE works. “rate.colors” are user specified colors to be used for coloring rates. “state.colors” are user specified colors to be used for coloring states. A vector with the color for states 00, 01, 10, and 11 (in this order). “edge.width” is the width of the branches of the phylogeny. “width.factor” is the factor multiplying the “edge.width” to get the width of the branch for the states tree. Needs to be a numeric value between 0 and 1. “rate.range” is an optional two element vector. If present, specifies the range of rate values to use for plotting. “lims.percentage.correction” deals with cases where the limits are slightly smaller than the values due to imprecision issues. “legend.position” are the coordinates for placing the legend. “legend.cex” is the text size inside the legend. “legend.kernel” lets you specify the way the density plot or histogram is made for rates. A value of “auto” chooses what we think is the best option given your data, “hist” makes a histogram, “rectangular”, “gaussian”, and others make a density plot. See `?density` for all non-“hist” options. “legend.bg” sets the color for the legend background. “mar” sets the margins for the plot. See more information in ’help(par)’. “outline” is whether to plot an outline on the branches of the tree. Choose the color of the outline using the parameter outline.color. “outline.color” is the color for the outline. Defaults to "black". “swap.underscore” sets whether to substitute all "_" with " " on the labels for the tips.

This function is very similar to the `hisse::plot.hisse.states` function. See more details in help page for `hisse::plot.hisse.states` function.

Author(s)

Brian O'Meara and Daniel Caetano

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

`PruneRedundantModels` *Prune Redundant Models*

Description

Removes models that fit equivalently well to models that are as simple or simpler.

Usage

```
PruneRedundantModels(..., precision=1e-5)
```

Arguments

`...`	A list of models (such as from MiSSEGreedy or just putting individual hisse runs in a list)
`precision`	How different the log likelihoods can be to assume they're equivalent models)

Details

Imagine two models: Model 1: parameters a and b allowed to vary Model 2: parameters a set to equal b

If the maximum likelihood estimate for a and b in model 1 are that they have the same value, it's essentially the same as including model 2 twice, and so other models have lower weight in consequence. This comes up with hidden state models where some hidden states are in extremely low frequencies throughout: they have more parameters than simpler models but the likelihood is essentially the same. Burnham and Anderson (2003, page 342ff) recommend removing redundant models in cases like this where two models are effectively identical.

This function does this by ordering models by the number of free parameters, then deleting models that have log likelihoods nearly exactly equal (to the precision set above) to a model that is as simple or simpler. This is not a perfect solution – it's possible that models could have very different parameters and have the same likelihood, for example. Also the "as simple" means that if there are two models with equal numbers of parameters, whichever is first gets saved and the second is removed.

Author(s)

Brian O'Meara

SimToPhylo *Convert simulated result to a tree*

Description

Converts the $results element of the return to a phylo object

Usage

```
SimToPhylo(results, include.extinct=FALSE, drop.stem=TRUE)
```

Arguments

`results`	dataframe of results from SimulateHisse.
`include.extinct`	include (TRUE) or delete (FALSE) extinct terminal taxa.
`drop.stem`	include the lineage leading to the first saved speciation even (FALSE) or cull it (TRUE).

Details

This takes the `$results` object from a `SimulateHisse` run and converts it to an ape `phylo` object. If there are no taxa on the final tree, it returns NA; if there is one taxon, it returns a one taxon tree. This is behavior different from diversitree's tree simulators, which returns NULL in both the zero and one taxon case. Extinct taxa can be pruned or not. For simulations starting with one taxon, and/or for simulations with some extinction, the final tree can have a time when there is a single lineage before it radiates into the crown group. This stem can be included or not. The tip states are stored in `$tip.state` in the returned tree.

Value

`phylo`	a phylo object in cladewise order.

Author(s)

Brian O'Meara

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

Examples

```
simulated.result <- SimulateHisse(c(.3, .1), c(.1, 0),
matrix(c(NA, 0.2, .3, NA), nrow=2), max.taxa=35, x0=1)
par(mfcol=c(1,2))
plot(SimToPhylo(simulated.result$results, include.extinct=TRUE))
plot(SimToPhylo(simulated.result$results, include.extinct=FALSE))
```

`SimulateGeoHiSSE` *Simulate under a GeoHiSSE model*

Description

Simulation function for the GeoHiSSE model

Usage

```
SimulateGeoHiSSE(pars, hidden.traits=1, x0="0A", max.taxa=Inf,
max.time=Inf, add.jumps=FALSE, add.extinction=FALSE,
include.extinct=FALSE, return.GeoHiSSE_pars=FALSE,
override.safeties=FALSE)
```

Arguments

`pars` an object of class "GeoHiSSE_pars" generated by `SimulateGeoHiSSE`. See 'Details'.

`hidden.traits` number of hidden areas in the model. Default is 1.

`x0` a character value for the starting state. Default is "01A".

`max.taxa` have the simulation stop at `max.taxa` surviving taxa.

`max.time` have the simulation stop at `max.time` height (including stem).

`return.GeoHiSSE_pars` whether to (only) return a "GeoHiSSE_pars" object in the correct format to use in this function. See 'Details'.

`add.jumps` whether to include jump dispersals in the model. Both for returning the parameters of the model and model estimation. Default is FALSE.

`add.extinction` whether to separate extirpation and extinction of endemic lineages. Both for returning the parameters of the model and model estimation. Default is FALSE.

`include.extinct` whether the simulation should keep extinct lineages. Default is FALSE.

`override.safeties` simulate even if there is no limit on number of taxa or depth of tree.

Details

This function uses diversitree::tree.classe function to simulate the tree under the GeoSSE model with hidden states. The parameter pars is a list of class "GeoHiSSE_pars" that can be generated using this same function. The best way is to first run SimulateGeoHiSSE with empty parameters and return.GeoHiSSE_pars=TRUE and with hidden.areas equal to the number of desired hidden states in the model. This will return an object in list format with a set of four matrices. Each matrix controls a different set of parameters of the model. All parameters are set to 0 and parameters not used by the model are set to NA. To prepare the simulation, edit these matrices and change the desired parameter values. Leaving a parameter equal to 0 is the same as reducing the model by eliminating this parameter of the model.

Value

This returns a list, with the following elements:

$phy	a phylogeny of class phylo.
$data	a data matrix with the species names as the first column and the ranges (in 0, 1, and 2 format) as the second column.
$hidden.states	a named vector with the 'true' simulated ranges (include the hidden state information).
$sim.attempts	number of times that the simulation tried before reaching desired termination condition.
$pars	the parameters used for the simulation.
$classe.pars	the parameters in 'make.classe' format for the 'diversitree' package.

Author(s)

Daniel Caetano

Examples

```
## Not run:
## Get the a list with the correct parameters to run the simulation:
pars <- SimulateGeoHiSSE(hidden.areas=1, return.GeoHiSSE_pars=TRUE)
## Edit the parameter values:
pars$model.pars[1:3,] <- 0.1
pars$model.pars[4:5,] <- 0.05
pars$model.pars[6:7,] <- 0.01
pars$q.01[1,2] <- pars$q.01[2,1] <- 0.01
pars$q.1[1,2] <- pars$q.1[2,1] <- 0.02
pars$q.2[1,2] <- pars$q.2[2,1] <- 0.02
## Run the simulation:
sim <- SimulateGeoHiSSE(pars=pars, hidden.areas=1, x0="01A", max.taxa=100)

## End(Not run)
```

`SimulateHisse` *Simulate under a HiSSE model*

Description

Flexible simulation function allowing checkpointing

Usage

```
SimulateHisse(turnover.rates, eps.values, transition.rates, max.taxa=Inf, max.t=Inf,
max.wall.time=Inf, x0, nstart=1, checkpoint.file=NULL, checkpoint.frequency=100,
checkpoint.start.object=NULL, override.safeties=FALSE, mass.extinction.heights=c(),
mass.extinction.magnitudes=c())
```

Arguments

`turnover.rates`	a vector of turnover rates.
`eps.values`	a vector of extinction fractions.
`transition.rates`	a matrix of transition rates.
`max.taxa`	have the simulation stop at `max.taxa` surviving taxa.
`max.t`	have the simulation stop at `max.t` height (including stem).
`max.wall.time`	have the simulation stop at this many seconds of running.
`x0`	the starting state.
`nstart`	how many taxa you want to start from. Usual values are 1 (start with single lineage) or 2 (crown group).
`checkpoint.file`	if you want to save progress (so you can restart from last saved point) include the name of a file.
`checkpoint.frequency`	if there is a checkpoint file, frequency (in terms of number of steps: births, deaths, or transitions) at which this is saved.
`checkpoint.start.object`	if you are starting from a checkpointed.file, the object loaded from that file. Typically called `checkpoint.result`.
`override.safeties`	simulate even if there is no limit on number of taxa, depth of tree, or wall time.
`mass.extinction.heights`	vector of times above the root [NOT from the present] to have a mass extinction.
`mass.extinction.magnitudes`	vector of per species extinction probability at mass extinctions.

Details

Note that currently, the simulator assumes `turnover.rates`, `eps.values`, and `transition rates` are in the same state order. Remember that `turnover.rates` are birth + death rates, and `eps.values` are death / birth rates.

We strongly advise putting some limits to stop the run. These can be any combination of `max.taxa`, `max.t`, and `max.wall.time`. For example, you could set up the run to stop when you hit 1000 taxa or 3600 seconds (one hour) of running on your computer, whichever comes first. If you want to run with no limits, you have to specify `override.safeties=TRUE`, and you should know what you're doing here (the runs might never finish). There is only one starting state (`x0`), which should be the index into the rate vectors, starting with 0: that is, if your states are 0A, 0B, 1A, and 1B, x0=0 would start in 0A, x0=3 would start in 1B. `nstart` lets you choose how many taxa to start with: it could be one lineage to start from a single taxon, in which case you'll have to wait some time for the first speciation event, or you could start with two, to start simulation with a clade of this size (though, if extinction is nonzero, you are not guaranteed to have the final crown group include both of these descendants). You can choose numbers higher than two, to start with, say, a 50-species polytomy. We're not sure why you would. You can add checkpointing: runs take a long time, and if it crashes, this will let you start from the same point in the last saved file (though with a different seed, unless you specify this yourself). If you want to save checkpoints as you go along, specify a file in `checkpoint.file`. Every `checkpoint.frequency` events, where an event is a speciation, extinction, or transition event, it will save current progress to a file. The smaller this value, the more frequently the simulation will be saved: fewer lost steps if you have to re-simulate, but slower during the run because it spends time writing to disk. If you want to start from a checkpointed analysis, load the file into R and then specify the object within R [not the file] containing the checkpointed results (by default, saved in an object called `checkpoint.result`) and the simulation will continue from that point.

Results are returned in a list, containing a dataframe with the outcome of the simulation (one row per edge in the tree, and containing information on the lengths, ancestor, tipward height from the original root, and state at the tipward end of the edge (you can get state at the rootward end by getting the tipward state of the ancestral edge). It may be of interest to know how many events of each type occurred, which is saved in `$birth.counts`, `$death.counts`, `$transition.counts`. The total number of surviving taxa is stored in `$n.surviving`.

The output can be returned as an ape `phylo` tree by passing the `$results` element of the final output to `SimToPhylo()`.

Value

This returns a list, with the following elements:

`$results` a dataframe containing the simulation output: states at nodes and tips, ancestor-descendant pairs, branch lengths.

`$birth.counts` a vector, in the same state order as turnover.rates et al., that includes the counts for the number of times species in each state speciated.

`$death.counts` same as birth.counts, but counting extinction events in each state outside of mass extinctions.

`$mass.extinction.death.counts`
total number of species that went extinct during a mass extinction event (which is assumed to be trait-independent).

`$transition.counts`
: matrix of counts of transitions from one state to another.

`$n.surviving` the number of taxa alive at the end of the simulation.

Author(s)

Brian O'Meara

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

Examples

```
simulated.result <- SimulateHisse(c(.3, .1), c(.1, 0),
matrix(c(NA, 0.2, .3, NA), nrow=2), max.taxa=35, x0=1)
par(mfcol=c(1,2))
plot(SimToPhylo(simulated.result$results, include.extinct=TRUE))
plot(SimToPhylo(simulated.result$results, include.extinct=FALSE))
```

SupportRegion.old *Adaptive Sampling of the Likelihood Surface*

Description

Adaptively samples points for each parameter to obtain an estimate of the confidence intervals.

Usage

```
SupportRegion.old(hisse.old.obj, n.points=1000, scale.int=0.1, desired.delta=2,
min.number.points=10, output.type="turnover", hidden.states=TRUE,
condition.on.survival=TRUE, root.type="madfitz", root.p=NULL, verbose=TRUE)
```

Arguments

`hisse.old.obj` an object of class `hisse.old.fit` that contains the MLE from a model run.

`n.points` indicates the number of points to sample.

`scale.int` the scaling multiplier that defines interval to randomly sample. By default the value is set to 0.1, meaning that values are drawn at random along an interval that encompasses 10 percent above and below the MLE.

`desired.delta` defines the number lnL units away from the MLE to include. By default the value is set to 2.

`min.number.points`
: sets the minimum number of points that can be returned. By default the value is set to 10.

`output.type` indicates whether the rates should be printed onscreen as the optimized variables, “turnover”, transformed to reflect net diversification, “net.div”, or transformed to reflect λ and μ, “raw”.

`hidden.states` a logical indicating whether the model includes a hidden state. The default is `TRUE`.

`condition.on.survival`
a logical indicating whether the likelihood was conditioned on the survival of two lineages and the speciation event subtending them (Nee et al. 1994). The default is `TRUE`.

`root.type` indicates whether root prior assumption should based the procedure described by FitzJohn et al. 2009, “madfitz”, assumed equal, “equal”, or set to user, “user”.

`root.p` a vector indicating fixed root state probabilities. The default is `NULL`.

`verbose` a logical indicating whether progress should be printed to the screen. The default is `TRUE`.

Details

This function provides a means for sampling the likelihood surface quickly to estimate confidence intervals that reflect the uncertainty in the MLE. The function starts with the MLE from the hisse run. It then uses a scaling multiplier to generate an interval by which to randomly alter each parameter. However, the algorithm was designed to “feel” the boundaries of the random search. In other words, when the algorithm begins to sample the hinterlands of the surface, it will know to restrict the boundary to allow sampling of more reasonable values based on the currently sampled set. The goal of this sampling process is to find points within some desired distance from the MLE; by default we assume this distance is 2 lnLik. The confidence interval can be estimated directly from these points. The full set of points tried are also provided and can be used to generate contour plots (though, it is not entirely straightforward to do so – but certainly doable).

Note the scale.int option. This roughly sets the variance for sampling points. If this seems to take a long while to find enough points within the desired likelihood region consider reducing scale.int to either 0.05 or, in some cases, 0.01.

Value

`SupportRegion.old` returns an object of class `hisse.old.support`. This is a list with elements:

`$ci` the sampled confidence interval.

`$points.within.region`
the sampled points that within 2lnL units from the MLE.

`$all.points` all points sampled by the adaptive sampler.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`SupportRegionGeoSSE` *Adaptive Sampling of the Likelihood Surface under the new version of GeoHiSSE model*

Description

Adaptively samples points for each parameter to obtain an estimate of the confidence intervals.

Usage

```
SupportRegionGeoSSE(geohisse.obj, n.points=1000, scale.int=0.1, desired.delta=2,
min.number.points=10, verbose=TRUE)
```

Arguments

`geohisse.obj`	an object of class `geohisse.fit` that contains the MLE from a model run.
`n.points`	indicates the number of points to sample.
`scale.int`	the scaling multiplier that defines interval to randomly sample. By default the value is set to 0.1, meaning that values are drawn at random along an interval that encompasses 10 percent above and below the MLE.
`desired.delta`	defines the number lnL units away from the MLE to include. By default the value is set to 2.
`min.number.points`	sets the minimum number of points that can be returned. By default the value is set to 10.
`verbose`	a logical indicating whether progress should be printed to the screen. The default is `TRUE`.

Details

This function provides a means for sampling the likelihood surface quickly to estimate confidence intervals that reflect the uncertainty in the MLE. The function starts with the MLE from the hisse run. It then uses a scaling multiplier to generate an interval by which to randomly alter each parameter. However, the algorithm was designed to "feel" the boundaries of the random search. In other words, when the algorithm begins to sample the hinterlands of the surface, it will know to restrict the boundary to allow sampling of more reasonable values based on the currently sampled set. The goal of this sampling process is to find points within some desired distance from the MLE; by default we assume this distance is 2 lnLik. The confidence interval can be estimated directly from these points. The full set of points tried are also provided and can be used to generate contour plots (though, it is not entirely straightforward to do so – but certainly doable).

Note the scale.int option. This roughly sets the variance for sampling points. If this seems to take a long while to find enough points within the desired likelihood region consider reducing scale.int to either 0.05 or, in some cases, 0.01.

Value

`SupportRegionGeoSSE.old` returns an object of class `geohisse.old.support`. This is a list with elements:

`$ci` the sampled confidence interval.

`$points.within.region`
the sampled points that within 2lnL units from the MLE.

`$all.points` all points sampled by the adaptive sampler.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`SupportRegionGeoSSE.old`

Adaptive Sampling of the Likelihood Surface under the original GeoHiSSE model

---

Description

Adaptively samples points for each parameter to obtain an estimate of the confidence intervals.

Usage

```
SupportRegionGeoSSE.old(geohisse.old.obj, n.points=1000, scale.int=0.1, desired.delta=2,
min.number.points=10, verbose=TRUE)
```

Arguments

`geohisse.old.obj`
: an object of class `geohisse.old.fit` that contains the MLE from a model run.

`n.points`
: indicates the number of points to sample.

`scale.int`
: the scaling multiplier that defines interval to randomly sample. By default the value is set to 0.1, meaning that values are drawn at random along an interval that encompasses 10 percent above and below the MLE.

`desired.delta`
: defines the number lnL units away from the MLE to include. By default the value is set to 2.

`min.number.points`
: sets the minimum number of points that can be returned. By default the value is set to 10.

`verbose`
: a logical indicating whether progress should be printed to the screen. The default is `TRUE`.

Details

This function provides a means for sampling the likelihood surface quickly to estimate confidence intervals that reflect the uncertainty in the MLE. The function starts with the MLE from the hisse run. It then uses a scaling multiplier to generate an interval by which to randomly alter each parameter. However, the algorithm was designed to “feel” the boundaries of the random search. In other words, when the algorithm begins to sample the hinterlands of the surface, it will know to restrict the boundary to allow sampling of more reasonable values based on the currently sampled set. The goal of this sampling process is to find points within some desired distance from the MLE; by default we assume this distance is 2 lnLik. The confidence interval can be estimated directly from these points. The full set of points tried are also provided and can be used to generate contour plots (though, it is not entirely straightforward to do so – but certainly doable).

Note the scale.int option. This roughly sets the variance for sampling points. If this seems to take a long while to find enough points within the desired likelihood region consider reducing scale.int to either 0.05 or, in some cases, 0.01.

Value

`SupportRegionGeoSSE.old` returns an object of class `geohisse.old.support`. This is a list with elements:

`$ci`
: the sampled confidence interval.

`$points.within.region`
: the sampled points that within 2lnL units from the MLE.

`$all.points`
: all points sampled by the adaptive sampler.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`SupportRegionHiSSE` *Adaptive Sampling of the Likelihood Surface under the new version of HiSSE*

Description

Adaptively samples points for each parameter to obtain an estimate of the confidence intervals.

Usage

```
SupportRegionHiSSE(hisse.obj, n.points=1000, scale.int=0.1, desired.delta=2,
min.number.points=10, verbose=TRUE)
```

Arguments

`hisse.obj`	an object of class `hisse.fit` that contains the MLE from a model run.
`n.points`	indicates the number of points to sample.
`scale.int`	the scaling multiplier that defines interval to randomly sample. By default the value is set to 0.1, meaning that values are drawn at random along an interval that encompasses 10 percent above and below the MLE.
`desired.delta`	defines the number lnL units away from the MLE to include. By default the value is set to 2.
`min.number.points`	sets the minimum number of points that can be returned. By default the value is set to 10.
`verbose`	a logical indicating whether progress should be printed to the screen. The default is `TRUE`.

Details

This is the support region estimator for the new version of `hisse`.

Note the scale.int option. This roughly sets the variance for sampling points. If this seems to take a long while to find enough points within the desired likelihood region consider reducing scale.int to either 0.05 or, in some cases, 0.01.

Value

`SupportRegionHiSSE` returns an object of class `hisse.support`. This is a list with elements:

`$ci` the sampled confidence interval.

`$points.within.region` the sampled points that within 2lnL units from the MLE.

`$all.points` all points sampled by the adaptive sampler.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

`SupportRegionMiSSE` *Adaptive Sampling of the Likelihood Surface under MiSSE*

Description

Adaptively samples points for each parameter to obtain an estimate of the confidence intervals.

Usage

```
SupportRegionMiSSE(misse.obj, n.points=1000, scale.int=0.1, desired.delta=2,
min.number.points=10, verbose=TRUE)
```

Arguments

`misse.obj` an object of class `misse.fit` that contains the MLE from a model run.

`n.points` indicates the number of points to sample.

`scale.int` the scaling multiplier that defines interval to randomly sample. By default the value is set to 0.1, meaning that values are drawn at random along an interval that encompasses 10 percent above and below the MLE.

`desired.delta` defines the number lnL units away from the MLE to include. By default the value is set to 2.

`min.number.points`
: sets the minimum number of points that can be returned. By default the value is set to 10.

`verbose`
: a logical indicating whether progress should be printed to the screen. The default is `TRUE`.

Details

This function provides a means for sampling the likelihood surface quickly to estimate confidence intervals that reflect the uncertainty in the MLE. The function starts with the MLE from the hisse run. It then uses a scaling multiplier to generate an interval by which to randomly alter each parameter. However, the algorithm was designed to "feel" the boundaries of the random search. In other words, when the algorithm begins to sample the hinterlands of the surface, it will know to restrict the boundary to allow sampling of more reasonable values based on the currently sampled set. The goal of this sampling process is to find points within some desired distance from the MLE; by default we assume this distance is 2 lnLik. The confidence interval can be estimated directly from these points. The full set of points tried are also provided and can be used to generate contour plots (though, it is not entirely straightforward to do so – but certainly doable).

Note the scale.int option. This roughly sets the variance for sampling points. If this seems to take a long while to find enough points within the desired likelihood region consider reducing scale.int to either 0.05 or, in some cases, 0.01.

Value

`SupportRegionMiSSE` returns an object of class `misse.support`. This is a list with elements:

`$ci`
: the sampled confidence interval.

`$points.within.region`
: the sampled points that within 2lnL units from the MLE.

`$all.points`
: all points sampled by the adaptive sampler.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

SupportRegionMuHiSSE *Adaptive Sampling of the Likelihood Surface under MuHiSSE*

Description

Adaptively samples points for each parameter to obtain an estimate of the confidence intervals.

Usage

```
SupportRegionMuHiSSE(muhisse.obj, n.points=1000, scale.int=0.1, desired.delta=2,
min.number.points=10, verbose=TRUE)
```

Arguments

`muhisse.obj`	an object of class `muhisse.fit` that contains the MLE from a model run.
`n.points`	indicates the number of points to sample.
`scale.int`	the scaling multiplier that defines interval to randomly sample. By default the value is set to 0.1, meaning that values are drawn at random along an interval that encompasses 10 percent above and below the MLE.
`desired.delta`	defines the number lnL units away from the MLE to include. By default the value is set to 2.
`min.number.points`	sets the minimum number of points that can be returned. By default the value is set to 10.
`verbose`	a logical indicating whether progress should be printed to the screen. The default is `TRUE`.

Details

This function provides a means for sampling the likelihood surface quickly to estimate confidence intervals that reflect the uncertainty in the MLE. The function starts with the MLE from the hisse run. It then uses a scaling multiplier to generate an interval by which to randomly alter each parameter. However, the algorithm was designed to “feel” the boundaries of the random search. In other words, when the algorithm begins to sample the hinterlands of the surface, it will know to restrict the boundary to allow sampling of more reasonable values based on the currently sampled set. The goal of this sampling process is to find points within some desired distance from the MLE; by default we assume this distance is 2 lnLik. The confidence interval can be estimated directly from these points. The full set of points tried are also provided and can be used to generate contour plots (though, it is not entirely straightforward to do so – but certainly doable).

Note the scale.int option. This roughly sets the variance for sampling points. If this seems to take a long while to find enough points within the desired likelihood region consider reducing scale.int to either 0.05 or, in some cases, 0.01.

Value

SupportRegionMuHiSSE returns an object of class muhisse.support. This is a list with elements:

$ci	the sampled confidence interval.
$points.within.region	the sampled points that within 2lnL units from the MLE.
$all.points	all points sampled by the adaptive sampler.

Author(s)

Jeremy M. Beaulieu

References

Beaulieu, J.M, and B.C. O'Meara. 2016. Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. Syst. Biol. 65:583-601.

FitzJohn R.G., Maddison W.P., and Otto S.P. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. Syst. Biol. 58:595-611.

Maddison W.P., Midford P.E., and Otto S.P. 2007. Estimating a binary characters effect on speciation and extinction. Syst. Biol. 56:701-710.

Nakov, T., Beaulieu, J.M., and Alverson, A.J. 2019. Diatoms diversify and turn over faster in freshwater than marine environments. Evolution, doi: https://doi.org/10.1111/evo.13832.

Nee S., May R.M., and Harvey P.H. 1994. The reconstructed evolutionary process. Philos. Trans. R. Soc. Lond. B Biol. Sci. 344:305-311.

TranslateParsMakerGeoHiSSE

Key to translate GeoHiSSE into ClaSSE

Description

Simulate key of parnames to translate between GeoHiSSE and ClaSSE models

Usage

```
TranslateParsMakerGeoHiSSE(k, add.extinction, add.jumps)
```

Arguments

k	number of hidden states in the model. Minimum is 0 for a model without hidden states (i.e., GeoSSE).
add.extinction	if the extinction parameter should be separated from extirpation.
add.jumps	if jumps between endemic areas should be added.

Details

Function will only return the parameters that are relevant for a GeoHiSSE model. The ClaSSE model will have many more parameters. The extra parameters of the ClaSSE model need to be all set to 0. The number of parameters of the correspondent ClaSSE model given the parameter `k` is given by:

(k+1)*3

Value

Function returns a matrix of characters names. Column 'classe.pars' have the parameter names for a ClaSSE model and column 'geohisse.pars' have the parameter names for the matching GeoHiSSE model.

Author(s)

Daniel Caetano

`TransMatMaker.old` *Transition Rate matrix generator*

Description

Generates and manipulates the index of the rate parameters to be optimized

Usage

```
TransMatMaker.old(hidden.states=FALSE)
ParDrop(rate.mat.index=NULL, drop.par=NULL)
ParEqual(rate.mat.index=NULL, eq.par=NULL)
```

Arguments

`hidden.states`	a logical indicating whether the underlying model includes hidden states. The default is `FALSE`.
`rate.mat.index`	A user-supplied rate matrix index to be manipulated.
`drop.par`	a vector of transitions to be dropped from the model.
`eq.par`	a vector of transitions pairs to be set equal.

Details

Outputs the full index of the rate parameters that are to be optimized. The intention is that a user might want to see how the matrix is designed prior to an analysis and perhaps drop a few parameters beforehand due to some hypothesis that he or she might have. The resulting matrix is to be plugged directly into `hisse.old`.

Value

Returns a rate matrix index

TransMatMakerGeoHiSSE *Transition Rate matrix generator for the GeoHiSSE model*

Description

Generates and manipulates the index of the transition rate parameters to be optimized.

Usage

```
TransMatMakerGeoHiSSE(hidden.traits=0, make.null=FALSE, include.jumps=FALSE,
separate.extirpation=FALSE)
```

Arguments

`hidden.traits`	a numeric value with the number of hidden trait in the model. The cannonical GeoSSE model has no hidden areas, so `'hidden.traits=0'`. The default value is 0.
`make.null`	Sets the transition matrix to the null model such that (1) transition rates from endemic to widespread areas (A -> AB and B -> AB) are the same across all hidden areas and (2) the transition rates between hidden areas (such as 0A <-> 1A) are the same for all areas.
`include.jumps`	allows for 0->1 and 1->0 transitions.
`separate.extirpation`	allows for 01->0 and 01->1 transitions.

Details

Outputs the full index of the rate parameters that are to be optimized.

The intention is that a user might want to see how the matrix is designed prior to an analysis and perhaps drop a few parameters beforehand due to some hypothesis that she might have. The resulting matrix is to be plugged directly into `GeoHiSSE`.

Value

Returns a rate matrix index.

TransMatMakerHiSSE *Transition Rate matrix generator for the HiSSE model*

Description

Generates and manipulates the index of the transition rate parameters to be optimized.

Usage

```
TransMatMakerHiSSE(hidden.traits=0, make.null=FALSE, cat.trans.vary=FALSE)
```

Arguments

`hidden.traits`	a numeric value with the number of hidden states in the model. The canonical BiSSE model has no hidden states, so `'hidden.traits=0'`. The default value is 0.
`make.null`	Sets the transition matrix to the null model such that (1) transition rates are the same across all hidden states and (2) the transition rates between hidden states (such as 0A <-> 1A) are the same.
`cat.trans.vary`	sets transition among hidden categories to vary.

Details

Outputs the full index of the rate parameters that are to be optimized.

The intention is that a user might want to see how the matrix is designed prior to an analysis and perhaps drop a few parameters beforehand due to some hypothesis that she might have. The resulting matrix is to be plugged directly into `hisse`.

Value

Returns a rate matrix index.

TransMatMakerMuHiSSE *Transition Rate matrix generator for the MuHiSSE model*

Description

Generates and manipulates the index of the transition rate parameters to be optimized.

Usage

```
TransMatMakerMuHiSSE(hidden.traits=0, make.null=FALSE, include.diagonals=FALSE,
cat.trans.vary=FALSE)
```

Arguments

`hidden.traits` a numeric value with the number of hidden states in the model. The canonical MuSSE model has no hidden states, so `'hidden.traits=0'`. The default value is 0.

`make.null` Sets the transition matrix to the null model such that (1) transition rates are the same across all hidden states and (2) the transition rates between hidden states (such as 00A <-> 01A) are the same.

`include.diagonals` allows for dual transitions, e.g., 00->11 and 11->00 transitions.

`cat.trans.vary` sets transitions among hidden categories to vary.

Details

Outputs the full index of the rate parameters that are to be optimized.

The intention is that a user might want to see how the matrix is designed prior to an analysis and perhaps drop a few parameters beforehand due to some hypothesis that she might have. The resulting matrix is to be plugged directly into `MuHiSSE`.

Value

Returns a rate matrix index.

Index