

Package ‘BMhyb’

August 3, 2019

Version 2.1.5

Date 2019-08-02

Type Package

Title Comparative Methods for Phylogenetic Networks

Imports corpcor, numDeriv, geiger, ape, TreeSim, mvtnorm, methods, lhs, viridis, Matrix, DEoptim, igraph, MASS, grDevices, ggplot2, cowplot, graphics, stats, metR, parallel, phytools, plyr

Depends R (>= 3.5.0)

Suggests testthat

Description Analyze the phenotypic evolution of species of hybrid origin on a phylogenetic network. This can detect a burst of variation at the formation of a hybrid as well as an increase or decrease in trait value at a hybridization event. Parameters are estimated by maximum likelihood, and model averaging can be done automatically. Users need to enter a comparative data set and a phylogenetic network.

License GPL (>= 2)

URL <http://github.com/bomeara/BMhyb>

BugReports <https://github.com/bomeara/BMhyb/issues>

RoxygenNote 6.1.1.9000

Encoding UTF-8

NeedsCompilation no

Author Dwueng-Chwuan Jhwueng [aut, cre],
Brian C. O'Meara [aut]

Maintainer Dwueng-Chwuan Jhwueng <djhwueng@umail.iu.edu>

Repository CRAN

Date/Publication 2019-08-02 22:10:10 UTC

R topics documented:

AddHybridization	2
BMhyb	3
BMhybExhaustive	5
cichlid	6
ComputeLikelihood	7
ComputeVCV	8
ConvertEvonetToIgraphWithNodeNumbers	9
CreateHybridlessEvonet	9
GetConvexHull	10
MergeExhaustiveForPlotting	10
nicotiana	11
plot.BMhybExhaustiveResult	11
plot.BMhybResult	12
print.BMhybExhaustiveResult	13
print.BMhybResult	13
SimulateNetwork	14
SimulateTips	15
summary.BMhybExhaustiveResult	16
summary.BMhybResult	16
Index	17

AddHybridization	<i>Add hybrid events to a phy.graph</i>
------------------	---

Description

Given an evonet object, and info on where the gene flow is from and to, and when this occurs, add a hybridization event. The edges things move from and to are specified by the list of descendant taxa of those edges (basically the edge is the subtending branch for the clade). You do not have to list all taxa, only those spanning the node at the end of the edge. You can enter a single taxon to have gene flow to or from a terminal branch. You also ideally will specify when the gene flow happens. This can be given as time from the root of the tree to when the event starts or time from the tip of the tree back to when the gene flow starts (but you must give one of these). If gene flow goes through an unsampled ghost intermediate, you can enter the length of time it spends there. If you do not specify any of these, flow is assumed to directly from the source to the recipient, with the time set at the start of the recent of the two branches (i.e., if flow goes from taxon A to taxon D, if D is younger the flow is assumed to happen partway up the terminal branch of A to directly connect to the start of D.

Usage

```
AddHybridization(phy.graph, from.clade, to.clade, time.from.root = NULL,
  time.from.tip = NULL, ghost.length = 0)
```

Arguments

<code>phy.graph</code>	An ape::evonet object (a phylogeny stored in phylo format that also includes a reticulation matrix)
<code>from.clade</code>	A vector of names specifying taxa spanning the node descended from the focal edge for the start of the hybridization event
<code>to.clade</code>	A vector of names specifying taxa spanning the node descended from the focal edge for the start of the hybridization event
<code>time.from.root</code>	When the hybridization event starts, as measured from the root of the tree
<code>time.from.tip</code>	When the hybridization event starts, as measured from the tips of the tree (assumed to be coeval)
<code>ghost.length</code>	How long the hybrid genes spend in an unsampled species before arriving in their recipient

Value

An evonet object with the new hybridization event

 BMhyb

Optimize model

Description

Fits a BMhyb model to your data.

Usage

```
BMhyb(phy.graph, traits, free.parameter.names = c("sigma.sq", "mu", "SE",
  "bt", "vh"), confidence.points = 5000, measurement.error = 0,
  gamma = 0.5, do.Higham.correction = FALSE,
  do.Brissette.correction = FALSE, verbose = TRUE,
  likelihood.precision = 0.01, max.steps = 10, confidence.lnl = 2,
  control = list(reltol = 0.001))
```

Arguments

<code>phy.graph</code>	An ape::evonet object (a phylogeny stored in phylo format that also includes a reticulation matrix)
<code>traits</code>	A vector of trait values, with names equal to the names of taxa on the phylogeny
<code>free.parameter.names</code>	What parameters you want to optimize rather than use defaults; options are sigma.sq, mu, SE, bt, and vh
<code>confidence.points</code>	How many points to use to estimate parameter uncertainty

<code>measurement.error</code>	How much uncertainty there is in tip values; a single number is applied to all taxa, a vector is applied to the corresponding taxa
<code>gamma</code>	In a hybridization event, what proportion of the trait comes from the donating parent. 0.5 means half comes from each parent
<code>do.Higham.correction</code>	Variance-covariance matrices for this model are sometimes poorly conditioned; this is a hack to reduce the impact of that
<code>do.Brissette.correction</code>	Applies method of Brissette et al. 2007 to also try to fix matrix condition
<code>verbose</code>	If TRUE, BMhyb will chat about its progress
<code>likelihood.precision</code>	When optimizing, how much of a lnL improvement is required to restart optimization between starts
<code>max.steps</code>	The number of restarts without improvement it will attempt
<code>confidence.lnl</code>	For figuring out the confidence interval, how wide you want the confidence region to be in lnL space
<code>control</code>	List of options to pass to optim. ?optim for help.

Details

This takes an `ape::evonet` object. If all you have is a tree (an `ape::phylo` object), you can use `CreateHybridlessEvoNet()` to convert the tree to an `evonet` object. You can then use the `AddHybridization()` function to add hybrid events to this object. Note that networks created in this way can, by chance, result in orders of nodes in the internal edge matrix that cause `ape`'s `reorder.phylo` function to crash, which is called in many of the plot and write functions. You can still use the plot functions in this package, however.

Value

Returns an object of class `BMhybResult` which contains `best` (a `data.frame` of the solution), `good.region` (`data.frame` of the points making up those in the `confidence.lnl` region), `bad.region` (all the other points sampled), `phy.graph` (same as what you put in), `traits` (same as what you put in), and `free.parameter.names`.

Examples

```
## Not run:
utils::data("cichlid")
result <- BMhyb(phy.graph=cichlid$phy.graph, traits=cichlid$trait,
  free.parameter.names=c("sigma.sq", "mu"))

## End(Not run)
```

BMhybExhaustive	<i>Exhaustively evaluate models</i>
-----------------	-------------------------------------

Description

Fits all possible BMhyb models to your data.

Usage

```
BMhybExhaustive(phy.graph, traits, measurement.error = 0,
  ncores = max(c(1, parallel::detectCores() - 1), na.rm = TRUE), ...)
```

Arguments

phy.graph	An ape::evonet object (a phylogeny stored in phylo format that also includes a reticulation matrix)
traits	A vector of trait values, with names equal to the names of taxa on the phylogeny
measurement.error	How much uncertainty there is in tip values; a single number is applied to all taxa, a vector is applied to the corresponding taxa
ncores	Number of cores to use. By default, uses parallel package to detect what's available and uses all but one.
...	All other parameters to pass to BMhyb (see ?BMhyb)

Details

This takes an ape::evonet object. If all you have is a tree (an ape::phylo object), you can use CreateHybridlessEvoNet() to convert the tree to an evonet object. You can then use the AddHybridization() function to add hybrid events to this object. Note that networks created in this way can, by chance, result in orders of nodes in the internal edge matrix that cause ape's reorder.phylo function to crash, which is called in many of the plot and write functions. You can still use the plot functions in this package, however.

This will return a list with one model result per element: you can plot these individually (see ?hybResult). By default, these results will include the information about uncertainty. We also compute a summary table so you can see the point estimates for each model and the likelihoods. It is often advisable to average across models, weighting each by its AICc weight, so this is also done automatically. We also return the single best model as an object for convenience, though for most users, we would suggest using the model average and looking at a set of fairly good models rather than look only at the single best one: there are often others that are nearly as good.

We do not expect large AIC difference between models unless you have a really large tree, and so you may get a warning if this happens. It is likely something has gone wrong with optimization. Look at all the models and examine for outliers. This issue can come up with certain combinations of networks and parameters (even, very rarely, in Brownian motion with no hybridization), where a step in the likelihood (inverting a matrix) does not yield a numerically stable result (the matrix is poorly conditioned). The 'likelihoods' in such cases are wrong, and they can look too good or too

bad. Neither is ideal, but you should especially beware cases where the 'best' model has likelihoods much below some of the other models – you will often see bad parameter estimates, too. If you get this, do not believe the results – perhaps look at models with better condition.

To try to help with this, if one or more of the models has poor condition at the maximum likelihood estimate, we report this as it having an obvious problem. It is still returned in the results and the `original.summary.df` objects, but it is excluded from model averaging, the `summary.df`, and the `best.model` return (though note the `ModelNumber` column in `summary.df` allowing you to get the matching model in the results list). A model not having an obvious problem does *not* mean it worked well, just that it does not exhibit one particular problematic issue. Essentially we're saying, "This model does not have a lion eating its foot" – which suggest it's not unhealthy in that way, but doesn't mean there's not a crocodile eating its hand. User beware. Plotting the confidence using the plot functions can help.

Value

Returns a list of objects of class `BMhybResult` (`results`), a summary data frame (`summary.df`) with parameter estimates and weights for all models where we do not see obvious problems, a summary data frame of all the models, whether or no they seemed to fail (`original.summary.df`), the model averaged result weighted by AICc weights of the unproblematic models (`model.average`), and the best unproblematic model (`best.model`).

Examples

```
## Not run:
utils::data("cichlid")
traits.only <- cichlid$traits_and_SE$trait
names(traits.only) <- rownames(cichlid$traits_and_SE)
all.models <- BMhybExhaustive(phy.graph=cichlid$phy.graph, traits=traits.only)
print(all.models$summary.df)

## End(Not run)
```

cichlid

Cichlid dataset

Description

A dataset containing a phylogenetic network and trait data for cichlid species

Format

A list with two items:

phy.graph the phylogenetic network in `ape::evonet` format

trait a vector of trait data

final.se a vector of standard error

Details

The tree is made by doing a tree search with mitochondrial data from Kobmuller, S., N. Duftner, K. M. Sefc, M. Aibara, M. Stipacek, M. Blanc, B. Egger, and C. Sturmbauer. 2007. Reticulate phylogeny of gastropod-shell-breeding cichlids from Lake Tanganyika: the result of repeated introgressive hybridization. *BMC Evolutionary Biology* 7:7.

<https://bmcevolbiol.biomedcentral.com/articles/10.1186/1471-2148-7-7>

We then added hybridization events based on their cartoon Fig. 4: <https://media.springernature.com/full/springer-static/image/art>

Hybridization events with solid lines (coeval events) were modeled as going from the later of the source or descendant nodes.

Hybridization events with dotted lines, indicating ghost lineages, went from the MRCA of the source clade to the MRCA of the recipient taxon.

Trait data comes from fishbase.

ComputeLikelihood	<i>Compute the likelihood for a set of parameters</i>
-------------------	---

Description

Computes likelihood for a given network, set of traits, and parameters.

Usage

```
ComputeLikelihood(parameters, phy.graph, traits, measurement.error = 0,
  gamma = 0.5, do.Higham.correction = FALSE,
  do.Brissette.correction = FALSE, do.DE.correction = FALSE)
```

Arguments

parameters	Named vector of parameter values; expected names are sigma.sq, mu, SE, bt, and vh
phy.graph	An ape::evonet object (a phylogeny stored in phylo format that also includes a reticulation matrix)
traits	A vector of trait values, with names equal to the names of taxa on the phylogeny
measurement.error	How much uncertainty there is in tip values; a single number is applied to all taxa, a vector is applied to the corresponding taxa
gamma	In a hybridization event, what proportion of the trait comes from the donating parent. 0.5 means half comes from each parent
do.Higham.correction	Variance-covariance matrices for this model are sometimes poorly conditioned; this is a hack to reduce the impact of that
do.Brissette.correction	Applies method of Brissette et al. 2007 to also try to fix matrix condition

do.DE.correction

Inspired by Mishra, Sudhanshu K. "The nearest correlation matrix problem: Solution by differential evolution method of global optimization." (2007)

Details

This takes an `ape::evonet` object. If all you have is a tree (an `ape::phylo` object), you can use `CreateHybridlessEvoNet()` to convert the tree to an `evonet` object. You can then use the `AddHybridization()` function to add hybrid events to this object. Note that networks created in this way can, by chance, result in orders of nodes in the internal edge matrix that cause `ape::reorder.phylo` function to crash, which is called in many of the plot and write functions. You can still use the plot functions in this package, however.

Value

Returns the negative log likelihood

ComputeVCV

Compute the variance-covariance matrix

Description

Creates a variance-covariance matrix for a network and parameters.

Usage

```
ComputeVCV(phy.graph, sigma.sq = 1, mu = 0, bt = 1, vh = 0,
           SE = 0, measurement.error = 0, gamma = 0.5)
```

Arguments

<code>phy.graph</code>	An <code>ape::evonet</code> object (a phylogeny stored in phylo format that also includes a reticulation matrix)
<code>sigma.sq</code>	Value for sigma squared
<code>mu</code>	Value for state at the root
<code>bt</code>	Value for beta parameter
<code>vh</code>	Value for V_h , the variance that comes from a hybridization event
<code>SE</code>	Standard error
<code>measurement.error</code>	How much uncertainty there is in tip values; a single number is applied to all taxa, a vector is applied to the corresponding taxa
<code>gamma</code>	In a hybridization event, what proportion of the trait comes from the donating parent. 0.5 means half comes from each parent

Value

Returns the variance-covariance matrix

`ConvertEvonetToIgraphWithNodeNumbers`*Convert an evonet object into igraph*

Description

ape can already convert from evonet to igraph; the advantage of this function is that it uses the node ids from the evonet object for labels in igraph.

Usage

```
ConvertEvonetToIgraphWithNodeNumbers(phy.graph)
```

Arguments

<code>phy.graph</code>	An <code>ape::evonet</code> object (a phylogeny stored in phylo format that also includes a reticulation matrix)
------------------------	--

Value

An igraph network

Examples

```
phy <- ape::rcoal(5)
phy.evo <- ape::evonet(phy, from=1, to=2)
plot(phy.evo) # this is the ape plot
phy.igraph <- ConvertEvonetToIgraphWithNodeNumbers(phy.evo)
plot(phy.igraph)
```

`CreateHybridlessEvoNet`*Convert phylo object to evonet*

Description

`ape::evonet` converts a phylo object to evonet, but requires having at least one hybridization event. This lets you convert to evonet without having a hybridization event.

Usage

```
CreateHybridlessEvoNet(phy)
```

Arguments

<code>phy</code>	A phylo object (ape's basic tree format)
------------------	--

Value

An evonet object, suitable for passing as phy.graph into many of BMhyb's functions

GetConvexHull	<i>Get convex hull at a given threshold</i>
---------------	---

Description

For a given delta lnL, get the convex hull (blob encircling the points) for two dimensions

Usage

```
GetConvexHull(threshold = 2, df, height, x, y)
```

Arguments

threshold	What value to exclude numbers worse than
df	The data.frame
height	The variable name to use for the height threshold
x	The first variable to look at for the hull
y	The second variable to look at for the hull

MergeExhaustiveForPlotting	<i>Function to merge adaptive sampling sims for plotting</i>
----------------------------	--

Description

Function to merge adaptive sampling sims for plotting

Usage

```
MergeExhaustiveForPlotting(exhaustive.object)
```

Arguments

exhaustive.object	Return of BMhybExhaustive
-------------------	---------------------------

Value

Returns a single BMhyb object with results from all models merged (use for plotting)

 nicotiana

Nicotiana dataset

Description

A dataset containing a phylogenetic network and trait data for *Nicotiana* species

Format

A list with two items:

phy.graph the phylogenetic network in ape::evonet format

trait a vector of trait data

Details

The tree and data come from

Chase M.W., Knapp S., Cox A.V., Clarkson J.J., Butsko Y., Joseph J., Savolainen V., and Parokony A.S. 2003. Molecular systematics, GISH and the origin of hybrid taxa in *Nicotiana*(Solanaceae). *Annals of Botany* 92: 107-127.

Clarkson J.J., Lim K.Y., Kovarik A., Chase M.W., Knapp S. and Leitch A.R. 2005. Long-term genome diploidization I allopolyploid *Nicotiana* section *Repandae*(Solanaceae). *New Phytologist* 168:241-252.

Komori T., Myers P.N., Yamada S., Kubo T., and Imaseki H. 2000. Comparative study of the *Nicotiana* species with respect to water deficit tolerance during early growth. *Euphytica* 116:121-130.

 plot.BMhybExhaustiveResult

Plot BMhybExhaustive result

Description

Note this aggregates the info from all the model runs and plots the MLE across all of them and the contours from all the sims

Usage

```
## S3 method for class 'BMhybExhaustiveResult'
plot(x, ...)
```

Arguments

x A BMhybExhaustive object (result of a BMhybExhaustive() call)
 ... Other parameters to pass to hybResult

plot.BMhybResult *Plot BMhyb result*

Description

Shows the plot of confidence regions with MLEs indicated (red dots) or a plot of pairs of traits together. Note that for the latter plot, it converts the sampled points to an even grid with interpolation; it sets any points with likelihood worse than ten units to just ten units worse so that you can see the colors near the area of the optimum.

Usage

```
## S3 method for class 'BMhybResult'
plot(x, style = "univariate",
     focal.color = "red", inregion.color = "black",
     outregion.color = "gray", gradientworst.color = "black",
     gradientbest.color = "white", contour.color = "red",
     contour.threshold = 2, nrow = NULL, ...)
```

Arguments

x	A BMhyb object (result of a BMhyb() call)
style	Either univariate or contour
focal.color	Color for the point showing the maximum likelihood estimate
inregion.color	Color for univariate plot, points in the good region
outregion.color	Color for univariate plot, points in the bad region
gradientworst.color	Color for contour plot, color of the worst contour region
gradientbest.color	Color for contour plot, color of the best contour region
contour.color	Color showing the contour line for the best threshold
contour.threshold	What delta log likelihood to use for the best/worst threshold for the contour plot
nrow	The number of rows to plot in the grid for contour (will set it automatically if NULL)
...	Other arguments to pass to plot (for univariate only; the contour plot uses ggplot2)

```
print.BMhybExhaustiveResult  
    Print BMhybExhaustive result
```

Description

Print BMhybExhaustive result

Usage

```
## S3 method for class 'BMhybExhaustiveResult'  
print(x, ...)
```

Arguments

x A BMhybExhaustive object (result of a BMhybExhaustive() call)
... Other arguments to pass to this function

```
print.BMhybResult      Print BMhyb result
```

Description

Print BMhyb result

Usage

```
## S3 method for class 'BMhybResult'  
print(x, ...)
```

Arguments

x A BMhyb object (result of a BMhyb() call)
... Other arguments to pass to this function

SimulateNetwork *Simulate a phylogenetic network*

Description

This uses a birth death process (`TreeSim::sim.bd.taxa.age`) to make a tree, then randomly adds hybridization events. The events are placed uniformly with time (not with numbers of taxa). If you use the `phy.graph` argument, you can pass in an existing phylogenetic network and it will add hybridization events to that; if you use a `phy` argument, it will add hybridization events to that. Note that currently there is no checking for multiple events between the same two branches. While hybridization events happen between taxa alive at the same instant of time, it is possible that the donor taxon later goes extinct with no descendants (other than the taxa of hybrid origin). These are basically ghost lineages, and this process (which then looks like gene flow going forward in time) is permitted if `allow.ghost` is `TRUE`.

Usage

```
SimulateNetwork(ntax = 100, nhybridizations = 10, birth = 1,
  death = 1, sample.f = 0.5, tree.height = 1, allow.ghost = FALSE,
  phy.graph = NULL, phy = NULL)
```

Arguments

<code>ntax</code>	How many surviving taxa to have on the tree (extinct taxa are pruned, with the exception of donors if <code>allow.ghost=TRUE</code>)
<code>nhybridizations</code>	How many hybridization events to have
<code>birth</code>	Birth rate (instantaneous rate)
<code>death</code>	Extinction rate (instantaneous)
<code>sample.f</code>	What fraction of taxa alive at the present to sample (resulting in the final <code>ntax</code>)
<code>tree.height</code>	Root to tip height of the final tree
<code>allow.ghost</code>	Allow a hybridization events from an unsampled ancestor
<code>phy.graph</code>	If not <code>NULL</code> , uses this network and adds hybridizations to it to reach <code>nhybridizations</code> in total
<code>phy</code>	If not <code>NULL</code> , uses this tree to create hybridization events on

Value

A `phy.graph` object with hybridizations

Examples

```
p <- SimulateNetwork(ntax=10 ,nhybridizations=2)
plot(p)
```

 SimulateTips

Simulate trait data

Description

For a given phylogenetic network generate tip data. Any values not specified use default values

Usage

```
SimulateTips(phy.graph, sigma.sq = 1, mu = 0, bt = 1, vh = 0,
  SE = 0, measurement.error = 0, gamma = 0.5,
  exclude.donors.recipients = TRUE)
```

Arguments

phy.graph	An ape::evonet object (a phylogeny stored in phylo format that also includes a reticulation matrix)
sigma.sq	The Brownian motion wiggle rate
mu	The population mean (in the absence of hybridization)
bt	The beta value (multiplier on expected value for each hybridization event)
vh	The burst of variance that comes from a hybridization event
SE	Uniform uncertainty at the tips
measurement.error	Uncertainty at the tips, especially if it varies between species
gamma	In a hybridization event, what proportion of the trait comes from the donating parent. 0.5 means half comes from each parent
exclude.donors.recipients	If TRUE, do not generate for any donors or recipient placeholder taxa

Value

A vector of trait values

Examples

```
network <- SimulateNetwork(ntax=5, nhybridizations=2)
tips <- SimulateTips(network, mu=1.1, bt=3, vh=1.1, SE=1)
```

summary.BMhybExhaustiveResult
Summarize BMhybExhaustive result

Description

Summarize BMhybExhaustive result

Usage

```
## S3 method for class 'BMhybExhaustiveResult'  
summary(object, ...)
```

Arguments

object A BMhybExhaustive object (result of a BMhybExhaustive() call)
... Other arguments to pass to this function

Value

A data.frame with summarized results

summary.BMhybResult *Summarize BMhyb result*

Description

Summarize BMhyb result

Usage

```
## S3 method for class 'BMhybResult'  
summary(object, ...)
```

Arguments

object A BMhyb object (result of a BMhyb() call)
... Other arguments to pass to this function

Value

A data.frame with summarized results

Index

*Topic **datasets**

cichlid, [6](#)

nicotiana, [11](#)

AddHybridization, [2](#)

BMhyb, [3](#)

BMhybExhaustive, [5](#)

cichlid, [6](#)

ComputeLikelihood, [7](#)

ComputeVCV, [8](#)

ConvertEvonetToIgraphWithNodeNumbers,
[9](#)

CreateHybridlessEvonet, [9](#)

GetConvexHull, [10](#)

MergeExhaustiveForPlotting, [10](#)

nicotiana, [11](#)

plot.BMhybExhaustiveResult, [11](#)

plot.BMhybResult, [12](#)

print.BMhybExhaustiveResult, [13](#)

print.BMhybResult, [13](#)

SimulateNetwork, [14](#)

SimulateTips, [15](#)

summary.BMhybExhaustiveResult, [16](#)

summary.BMhybResult, [16](#)