Package ‘RPANDA’

August 4, 2018

Version 1.4
Date 2018-08-01
Type Package
Title Phylogenetic ANalyses of DiversificAtion
Depends R (>= 2.14.2), picante, methods
Suggests testthat
Imports ape, pspline, deSolve, igraph, TESS, cluster, fpc, pvclust, corpcor, phytools, mvMORPH, stats, graphics, grDevices, utils, BioGeoBEARS, geiger, mvtnorm, glassoFast
Encoding UTF-8
Author Hélène Morlon [aut, cre, cph], Eric Lewitus [aut, cph], Fabien Condamine [aut, cph], Marc Manceau [aut, cph], Julien Clavel [aut, cph], Jonathan Drury [aut, cph]
Maintainer Hélène Morlon <morlon@biologie.ens.fr>
License GPL-2
URL https://github.com/hmorlon/PANDA
Repository CRAN
RoxygenNote 5.0.1
R topics documented:

RPANDA-package .................................................. 4
ancestral ......................................................... 5
Anolis.data .......................................................... 6
Balaenopteridae ..................................................... 7
BGB.examples ....................................................... 8
BICompare .......................................................... 9
Calomys ............................................................. 10
Cetacea ............................................................. 10
c02 ................................................................. 11
c02_res ............................................................. 12
coccolithophore ................................................... 13
CreateClassObject .................................................. 13
CreateGeobyClassObject .......................................... 15
CreateGeoObject ................................................... 17
CreateGeoObject_BioGeoBEARS ................................. 18
createModel ......................................................... 20
createModelCoevolution ........................................... 21
d13c ................................................................. 22
fitTipData ........................................................... 23
fitTipData-methods ................................................ 24
fit_bd ............................................................... 25
fit_coal_cst ........................................................ 28
fit_coal_var ........................................................ 30
fit_env .............................................................. 32
fit_sgd .............................................................. 35
fit_t_comp .......................................................... 36
fit_t_comp_subgroup ............................................... 38
fit_t_env ............................................................ 41
fit_t_pl ............................................................. 45
foraminifera ....................................................... 47
getDataLikelihood .................................................. 48
dataGetLikelihood-methods ....................................... 49
dataGetLikelihood-methods ....................................... 49
dataGetLikelihood-methods ....................................... 49
GIC.fit_plRpanda .................................................. 51
gic_criterion ...................................................... 52
greenalgae .......................................................... 54
InfTemp ............................................................. 55
JSDtree ............................................................. 56
JSDtree_cluster ..................................................... 57
landplant ........................................................... 58
likelihood_bd ...................................................... 59
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>likelihood_coal_cst</td>
<td>60</td>
</tr>
<tr>
<td>likelihood_coal_var</td>
<td>62</td>
</tr>
<tr>
<td>likelihood_sgd</td>
<td>63</td>
</tr>
<tr>
<td>likelihood_subgroup_model</td>
<td>64</td>
</tr>
<tr>
<td>likelihood_t_DD</td>
<td>66</td>
</tr>
<tr>
<td>likelihood_t_DD_geog</td>
<td>67</td>
</tr>
<tr>
<td>likelihood_t_env</td>
<td>69</td>
</tr>
<tr>
<td>likelihood_t_MC</td>
<td>71</td>
</tr>
<tr>
<td>likelihood_t_MC_geog</td>
<td>72</td>
</tr>
<tr>
<td>lines.fit_t_env</td>
<td>73</td>
</tr>
<tr>
<td>modelSelection</td>
<td>75</td>
</tr>
<tr>
<td>modelSelection-methods</td>
<td>75</td>
</tr>
<tr>
<td>ostracoda</td>
<td>76</td>
</tr>
<tr>
<td>PhenotypicACDC-class</td>
<td>76</td>
</tr>
<tr>
<td>PhenotypicADiag-class</td>
<td>78</td>
</tr>
<tr>
<td>PhenotypicBM-class</td>
<td>79</td>
</tr>
<tr>
<td>PhenotypicDD-class</td>
<td>80</td>
</tr>
<tr>
<td>PhenotypicGMM-class</td>
<td>81</td>
</tr>
<tr>
<td>PhenotypicModel-class</td>
<td>82</td>
</tr>
<tr>
<td>PhenotypicOU-class</td>
<td>83</td>
</tr>
<tr>
<td>PhenotypicPM-class</td>
<td>85</td>
</tr>
<tr>
<td>Phocoenidae</td>
<td>86</td>
</tr>
<tr>
<td>phyl.pca_pl</td>
<td>86</td>
</tr>
<tr>
<td>Phyllostomidae</td>
<td>88</td>
</tr>
<tr>
<td>Phyllostomidae.genera</td>
<td>89</td>
</tr>
<tr>
<td>plot.fit_t_env</td>
<td>89</td>
</tr>
<tr>
<td>plot_BICompare</td>
<td>91</td>
</tr>
<tr>
<td>plot_dtt</td>
<td>92</td>
</tr>
<tr>
<td>plot_fit_bd</td>
<td>93</td>
</tr>
<tr>
<td>plot_fit_env</td>
<td>94</td>
</tr>
<tr>
<td>plot_spectR</td>
<td>95</td>
</tr>
<tr>
<td>Posdef</td>
<td>96</td>
</tr>
<tr>
<td>radiolaria</td>
<td>97</td>
</tr>
<tr>
<td>redalgae</td>
<td>98</td>
</tr>
<tr>
<td>sealevel</td>
<td>98</td>
</tr>
<tr>
<td>silica</td>
<td>99</td>
</tr>
<tr>
<td>sim.convergence.geo</td>
<td>100</td>
</tr>
<tr>
<td>sim.divergence.geo</td>
<td>101</td>
</tr>
<tr>
<td>simulateTipData</td>
<td>102</td>
</tr>
<tr>
<td>simulateTipData-methods</td>
<td>103</td>
</tr>
<tr>
<td>sim_env_bd</td>
<td>104</td>
</tr>
<tr>
<td>sim_sgd</td>
<td>106</td>
</tr>
<tr>
<td>sim_t.comp</td>
<td>107</td>
</tr>
<tr>
<td>sim_t_env</td>
<td>108</td>
</tr>
<tr>
<td>spectR</td>
<td>110</td>
</tr>
<tr>
<td>Index</td>
<td>112</td>
</tr>
</tbody>
</table>
Description

Implements macroevolutionary analyses on phylogenetic trees

Details

<table>
<thead>
<tr>
<th>Package:</th>
<th>RPANDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type:</td>
<td>Package</td>
</tr>
<tr>
<td>Version:</td>
<td>1.3</td>
</tr>
<tr>
<td>Date:</td>
<td>2017-04-07</td>
</tr>
<tr>
<td>License:</td>
<td>GPL (&gt;= 2)</td>
</tr>
</tbody>
</table>

Author(s)

Hélène Morlon <morlon@biologie.ens.fr>
Julien Clavel <clavel@biologie.ens.fr>
Fabien Condamine <fabien.condamine@gmail.com>
Jonathan Drury <jonathan.p.druy@gmail.com>
Eric Lewitus <lewitus@biologie.ens.fr>
Marc Manceau <manceau@biologie.ens.fr>

References

**Description**

Reconstruct the ancestral states at the root (and possibly for each node) of a phylogenetic tree from models fit obtained using the `fit_t_xx` functions.

**Usage**

```r
ancestral(object)
```

**Arguments**

- `object`: A model fit object obtained by the `fit_t_xx` class of functions.

**Details**

`ancestral` is an S3 method that reconstruct the ancestral states at the root and possibly for each node of a phylogenetic tree from the models fit obtained by the `fit_t_xx` class of functions (e.g., `fit_t_pl`, `fit_t_comp` and `fit_t_env`). Ancestral states are estimated using generalized least squares (GLS; Martins & Hansen 1997, Cunningham et al. 1998).

**Value**

A list with the following components:

- `root`: the reconstructed ancestral states at the root
- `nodes`: the reconstructed ancestral states at each node (not yet implemented for all the methods)

**Note**

The function is used internally in `phyl.pca_pl`.

**Author(s)**

J. Clavel
References


See Also

fit_t_pl, fit_t_env, phyl.pca_pl, GIC, gic_criterion

Examples

set.seed(1)
n <- 32 # number of species
p <- 31 # number of traits
tree <- pbtree(n=n) # phylogenetic tree
R <- Posdef(p) # a random symmetric matrix (covariance)

# simulate a dataset
Y <- mvSIM(tree, model="BM1", nsim=1, param=list(sigma=R))

# fit a multivariate BM with Penalized likelihood
fit <- fit_t_pl(Y, tree=model="BM", method="RidgeAlt")

# Perform the ancestral states reconstruction
anc <- ancestral(fit)

# retrieve the scores
head(anc$nodes)

---

**Anolis.data**

**Anolis dataset**

Description

Phylogeny, trait data, and geography.object for a subclade of Greater Antillean *Anolis* lizards.

Usage

data(Anolis.data)
Balaenopteridae

Details

Illustrative phylogeny trimmed from the maximum clade credibility tree of Mahler et al. 2013, corresponding phylogenetic principal component data from Mahler et al. 2013, and biogeography data from Mahler & Ingram 2014 (in the form of a geography object, as detailed in the CreateGeoObject help file).

References


See Also

CreateGeoObject

Examples

data(HanolisNdata)
plot(HanolisNdata$phylo)
print(HanolisNdata$data)
print(HanolisNdata$geography.object)

Balaenopteridae

Balaenopteridae phylogeny

Description

Ultrametric phylogenetic tree of the 9 extant Balaenopteridae species

Usage

data(Balaenopteridae)

Details

This phylogeny was extracted from Steeman et al. Syst Bio 2009 cetacean phylogeny

References


Description

Phylogenies and example stochastic maps for Canidae (from an unstratified BioGeoBEARS analysis) and Ochotonidae (from a stratified BioGeoBEARS analysis)

Usage

data(BGB.examples)

References


See Also

CreateGeoObject_BioGeoBEARS

Examples

data(BGB.examples)
par(mfrow=c(1,2))
plot(BGB.examples$Canidae.phylo)
plot(BGB.examples$Ochotonidae.phylo)
**Description**

Computes the BIC values for a specified number of modalities in the distance matrix of a phylogenetic tree and that of randomly bifurcating trees; identifies these modalities using k-means clustering.

**Usage**

```r
BICompare(phylo, t, meth=c("ultrametric"))
```

**Arguments**

- `phylo`: an object of type 'phylo' (see ape documentation)
- `t`: the number of modalities to be tested
- `meth`: whether the randomly bifurcating "control" tree should be ultrametric or non-ultrametric

**Value**

A list with the following components:

- `bic_test`: BIC values for finding `t` modalities in the distance matrix of a tree and the lowest five percent of 1000 random ("control") trees
- `clusters`: a vector specifying which nodes in the tree belong to each of `t` modalities
- `BSS/TSS`: the ratio of between-cluster sum of squares over total sum of squares

**Author(s)**

E Lewitus

**References**

Lewitus, E., Morlon, H., Characterizing and comparing phylogenies from their Laplacian spectrum, bioRxiv doi: http://dx.doi.org/10.1101/026476

**See Also**

- `plot_BICompare`, `spectR`, `JSDtree`

**Examples**

```r
data(Cetacea)
#BICompare(Cetacea, 5)
```
Calomys  

*Calomys phylogeny*

---

**Description**

Ultrametric phylogenetic tree of 11 of the 13 extant Calomys species

**Usage**

```r
data(Calomys)
```

**Details**

This phylogeny is from Pigot et al. PloS Biol 2012

**References**


Manceau, M., Lambert, A., Morlon, H. (submitted)

**Examples**

```r
data(Calomys)
print(Calomys)
plot(Calomys)
```

---

Cetacea  

*Cetacean phylogeny*

---

**Description**

Ultrametric phylogenetic tree for 87 of the 89 extant cetacean species

**Usage**

```r
data(Cetacea)
```

**Details**

This phylogeny was constructed by Bayesian phylogenetic inference from six mitochondrial and nine nuclear genes. It was calibrated using seven paleontological age constraints and a relaxed molecular clock approach. See Steeman et al. (2009) for details.
Source

Steeman ME et al. (2009) Radiation of extant cetaceans driven by restructuring of the oceans, Syst Biol 58:573-585

References

Steeman ME et al. (2009) Radiation of extant cetaceans driven by restructuring of the oceans Syst Biol 58:573-585


Examples

data(Cetacea)
print(Cetacea)
plot(Cetacea)

co2

c2 data since the Jurassic

description

Atmospheric co2 data since the Jurassic

Usage

data(co2)

Details

Atmospheric co2 data since the Jurassic taken from Mayhew et al., (2008, 2012) and derived from the GeoCarb-III model (Berner and Kothavala, 2001). The data are reported as the ratio of the mass of co2 at time t to that at present. The format is a dataframe with the two following variables:

age a numeric vector corresponding to the geological age, in Myrs before the present
co2 a numeric vector corresponding to the estimated co2 at that age

References


Examples
data(co2)
plot(co2)

co2_res  

co2 data since the beginning of the Cenozoic

Description

Atmospheric co2 data since the beginning of the Cenozoic

Usage
data(co2_res)

Details

Implied co2 data since the beginning of the Cenozoic taken from Hansen et al., (2013). The data are the amount of co2 in ppm required to yield observed global temperature throughout the Cenozoic:

age  a numeric vector corresponding to the geological age, in Myrs before the present
co2  a numeric vector corresponding to the estimated co2 at that age

Source


References


Examples
data(Cetacea)
print(Cetacea)
plot(Cetacea)
**Description**

Coccolithophore fossil diversity since the Jurassic

**Usage**

data(coccolithophore)

**Details**

Coccolithophore fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the two following variables:

- age: a numeric vector corresponding to the geological age, in Myrs before the present
- coccolithophore: a numeric vector corresponding to the estimated coccolithophore change at that age

**References**


**Examples**

data(coccolithophore)
plot(coccolithophore)

---

**CreateClassObject**

*Create class object*

**Description**

This function returns names of internode intervals, named descendants of each node, and a class object formatted in a way that can be passed to *CreateGeobyClassObject*

**Usage**

CreateClassObject(simmap, rnd=5)
Arguments

- `simmap`: stochastic map from `make.simmap` in phytools
- `rnd`: integer indicating the number of decimal places to which times should be rounded (default value is 5) (see `round`)

Details

This function formats the class object so that it can be correctly passed to the numerical integration performed in `fit_t_comp_subgroup`.

Value

A list with the following components:

- `class.object`: a list of matrices specifying the state of each branch during each internode interval (see Details)
- `times`: a vector containing the time since the root of the tree at which nodes or changes in biogeography occur (used internally in other functions)
- `spans`: a vector specifying the distances between times (used internally in other functions)

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References


See Also

`fit_t_comp_subgroup`, `CreateGeobyClassObject`

Examples

```r
data(Anolis.data)

# Create a make.simmap object
require(phytools)
geo<-c(rep("cuba",7),rep("hispamiana",9),"puerto_rico")
names(geo)<-Anolis.data$phylo$tip.label
stochastic.map<-make.simmap(Anolis.data$phylo,geo, model="ER", nsim=1)
CreateClassObject(stochastic.map)
```
CreateGeobyClassObject

Create merged biogeography-by-class object

Description

Create a merged biogeography-by-class object to be passed to fit_t_comp_subgroup using a stochastic map created from any model in BioGeoBEARS (see documentation in BioGeoBEARS package) and a simmap object from phytools (see documentation in phytools package).

Usage

CreateGeobyClassObject(phylo, simmap, trim.class, ana.events, clado.events, stratified=FALSE, rnd=5)

Arguments

phylo the object of type 'phylo' (see ape documentation) used to build ancestral range stochastic maps in BioGeoBEARS
simmap a phylo object created using make.simmap in phytools
trim.class category in the simmap object that represents the subgroup of interest (see Details and Examples)
ana.events the "ana.events" table produced in BioGeoBEARS that lists anagenetic events in the stochastic map
clado.events the "clado.events" table produced in BioGeoBEARS that lists cladogenetic events in the stochastic map
stratified logical indicating whether the ancestral biogeography stochastic map was built from a stratified analysis in BioGeoBEARS
rnd an integer value indicating the number of decimals to which values should be rounded in order to reconcile class and geo.objects (default is 5)

Details

This function merges a class object (which reconstructs group membership through time) and a stochastic map of ancestral biogeography (to reconstruct sympatry through time), such that lineages can only interact when they belong to the same subgroup AND are sympatric.

This allows fitting models of competition where only sympatric members of a subgroup can compete (e.g., all lineages that share similar diets or habitats).

This function should be used to format the geography object so that it can be correctly passed to the numerical integration performed in fit_t_comp_subgroup.
CreateGeobyClassObject

Value

Returns a list with the following components:

- **map**: a simmap object with phylogeny trimmed to subgroup of interest (including all branches determined to belong to that subgroup)

- **geography.object**: a list with the following components:
  - **geography.matrix**: a list of matrices specifying both sympatry & group membership (==1) or allopatry and/or non-membership in the focal subgroup (==0) for each species pair for each internode interval (see Details)
  - **times**: a vector containing the time since the root of the tree at which nodes or changes in biogeographyXsubgroup membership occur (used internally in other functions)
  - **spans**: a vector specifying the distances between times (used internally in other functions)

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References


See Also

- `fit_t_comp_subgroup`, `CreateGeoObject_BioGeoBEARS`, `CreateClassObject`

Examples

data(BGB.examples)

Canidae.phylo<-BGB.examples$Canidae.phylo
names(dummy.group)<-Canidae.phylo$tip.label

Canidae.simmap<-make.simmap(Canidae.phylo,dummy.group)

#build GeobyClass object with "A" as the focal group

Canidae.geobyclass.object<-CreateGeobyClassObject(phylo=Canidae.phylo,simmap=Canidae.simmap, trim.class="A",ana.events=BGB.examples$Canidae.ana.events,
CreateGeoObject

Create biogeography object

Description

This function returns names of internode intervals, named descendants of each node, and a geography object formatted in a way that can be passed to `fit_t_comp`.

Usage

CreateGeoObject(phylo, map)

Arguments

- **phylo**: an object of type 'phylo' (see ape documentation)
- **map**: either a matrix modified from `phylo$edge` or a phylo object created using `make.simmap` (see Details and Examples)

Details

This function should be used to format the geography object so that it can be correctly passed to the numerical integration performed in `fit_t_comp`.

The map can either be a matrix formed by specifying the region in which each branch specified by `phylo$edge` existed, or a stochastic map stored as a phylo object output from `make.simmap` (see Examples).

Value

A list with the following components:

- **geography.object**: a list of matrices specifying sympatry (1) or allopatry (0) for each species pair for each internode interval (see Details)
- **times**: a vector containing the time since the root of the tree at which nodes or changes in biogeography occur (used internally in other functions)
- **spans**: a vector specifying the distances between times (used internally in other functions)
Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References


See Also

fit_t_comp

Examples

data(Anolis.data)
# Create a geography object with a modified edge matrix
# First, specify which region each branch belonged to:
Anolis.regions<-c(rep("cuba",14),rep("hispaniola",17),"puerto_rico")
Anolis.map<-cbind(Anolis.data$phylo$edge,Anolis.regions)
CreateGeoObject(Anolis.data$phylo,map=Anolis.map)

# Create a geography object with a make.simmap object
# First, specify which region each branch belonged to:
require(phytools)
geo<-c(rep("cuba",7),rep("hispaniola",9),"puerto_rico")
names(geo)<-Anolis.data$phylo$tip.label
stochastic.map<-make.simmap(Anolis.data$phylo,geo, model="ER", nsim=1)
CreateGeoObject(Anolis.data$phylo,map=stochastic.map)
Arguments

- **full.phylo**: the object of type 'phylo' (see ape documentation) that was used to construct the stochastic map in BioGeoBEARS.
- **trimmed.phylo**: if the desired biogeography object excludes some species that were initially included in the stochastic map, this specifies a phylo object for the trimmed set of species.
- **ana.events**: the "ana.events" table produced in BioGeoBEARS that lists anagenetic events in the stochastic map.
- **clado.events**: the "clado.events" table produced in BioGeoBEARS that lists cladogenetic events in the stochastic map.
- **stratified**: logical indicating whether the stochastic map was built from a stratified analysis in BioGeoBEARS.

Value

A list with the following components:

- **geography.object**: a list of matrices specifying sympatry (1) or allopatry (0) for each species pair for each internode interval (see Details).
- **times**: a vector containing the time since the root of the tree at which nodes or changes in biogeography occur (used internally in other functions).
- **spans**: a vector specifying the distances between times (used internally in other functions).

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References


See Also

- fit_t_comp
- CreateGeoObject

Examples

data(BGB.examples)
createModel

## Example with a non-stratified tree

```r
Canidae.geography.object<-CreateGeoObject_BioGeoBEARS(full.phylo=BG3.example$Canidae.phylo, ana.events=BG3.example$Canidae.ana.events, clado.events=BG3.example$Canidae.clado.events)

# on a subclade
Canidae.trimmed<-drop.tip(BG3.example$Canidae.phylo, BG3.example$Canidae.phylo$tip.label[1:9])

Canidae.trimmed.geography.object<-CreateGeoObject_BioGeoBEARS(full.phylo=BG3.example$Canidae.phylo, trimmed.phylo=Canidae.trimmed, ana.events=BG3.example$Canidae.ana.events, clado.events=BG3.example$Canidae.clado.events)
```

## Example with a stratified tree

```r
Ochotonidae.geography.object<-CreateGeoObject_BioGeoBEARS(full.phylo=BG3.example$Ochotonidae.phylo, ana.events=BG3.example$Ochotonidae.ana.events, clado.events=BG3.example$Ochotonidae.clado.events, stratified = TRUE)

# on a subclade
Ochotonidae.trimmed<-drop.tip(BG3.example$Ochotonidae.phylo, BG3.example$Ochotonidae.phylo$tip.label[1:9])

Ochotonidae.trimmed.geography.object<-CreateGeoObject_BioGeoBEARS(full.phylo=BG3.example$Ochotonidae.phylo, trimmed.phylo=Ochotonidae.trimmed, ana.events=BG3.example$Ochotonidae.ana.events, clado.events=BG3.example$Ochotonidae.clado.events, stratified=TRUE)
```

---

**createModel** *Creation of a Phenotypic Model*

**Description**

Creates an object of class PhenotypicModel, intended to represent a model of trait evolution on a specific tree. Distinct keywords correspond to different models, using one phylogenetic tree.

**Usage**

```r
createModel(tree, keyword)
```

**Arguments**

- **tree**
  
an object of class `'phylo'` as defined in the R package `ape`.

- **keyword**
  
a string specifying the model. Available models include "BM", "BM_from0", "BM_from0_driftless", "OU", "OU_from0", "ACDC", "DD", "PM", "PM_OUless".
Value

the object of class "PhenotypicModel".

Author(s)

M Manceau

References


Examples

#Loading an example tree
tree <- read.tree(text=newick)

#Creating the models
modelBM <- createModel(tree, 'BM')
modelOU <- createModel(tree, 'OU')

#Printing basic or full informations on the model definitions
show(modelBM)
print(modelOU)

createModelCoevolution

Creation of a PhenotypicGMM

Description

Creates an object of class PhenotypicGMM, a subclass of the class PhenotypicModel intended to represent the Generalist Matching Mutualism model of trait evolution on two specific trees.

Usage

createModelCoevolution(tree1, tree2, keyword)

Arguments

tree1 an object of class 'phylo' as defined in the R package 'ape'.
tree2 an object of class 'phylo' as defined in the R package 'ape'.
keyword a string object. Default value "GMM" returns an object of class PhenotypicGMM, which takes advantage of faster distribution computation. Otherwise, a "PhenotypicModel" is returned, and the computation of the tip distribution will take much longer.
Value

an object of class "PhenotypicModel" or "PhenotypicGMM".

Author(s)

M Manceau

References


Examples

#Loading example trees
tree1 <- read.tree(text=newick1)
newick2 <- "((X:1.5,Y:1.5):3,Z:4.5);"
tree2 <- read.tree(text=newick2)

#Creating the model
modelGMM <- createModelCoevolution(tree1, tree2)

#Printing basic or full informations on the model definitions
show(modelGMM)
print(modelGMM)

#Simulates tip trait data
dataGMM <- simulateTipData(modelGMM, c(0,0,5,-5, 1, 1), method=2)

---

d13c data since the Jurassic

d13c

Description

Benthic d13c weathering ratio since the Jurassic

Usage

data(d13c)

Details

Ratio of stable carbon isotopes since the Jurassic calculated by Hannisdal and Peters (2011) and Lazarus et al. (2014) from marine carbonates. The format is a dataframe with the two following variables:

age  a numeric vector corresponding to the geological age, in Myrs before the present
d13c  a numeric vector corresponding to the estimated d13c at that age
References


Examples

data(d13c)
plot(d13c)

---

**fitTipData**

Maximum likelihood estimators of a model's parameters

Description

Finds the maximum likelihood estimators of the parameters, returns the likelihood and the inferred parameters.

Usage

`fitTipData(object, data, params0, GLSstyle, v)`

Arguments

- **object**: an object of class 'PhenotypicModel'.
- **data**: vector of tip trait data.
- **params0**: vector of parameters used to initialize the optimization algorithm. Default value is NULL, in which case the optimization procedure starts with the vector 'params0' specified within the 'model' object.
- **GLSstyle**: boolean specifying the way the mean trait value at the root is estimated. Default value is FALSE in which case the mean at the root is considered as any other parameter. If TRUE, the mean value at the root is estimated with the GLS method, as explained, e.g. in Hansen 1997.
- **v**: boolean specifying the verbose mode. Default value : FALSE.

Details

Warning : This function uses the standard R optimizer "optim". It may not always converge well. Please double check the convergence by trying distinct parameter sets for the initialisation.
Value

value A numerical value: the lowest \(-\log(\text{likelihood})\) value found during the optimization procedure.

inferredParams The maximum likelihood estimators of the model’s parameters.

convergence An integer code specifying the convergence of the optim function. Please refer to the optim function help files.

Author(s)

M Manceau

References


Examples

#Loading an example tree
tree <- read.tree(text=newick)

#Creating the models
modelBM <- createModel(tree, 'BM')

#Simulating tip traits under the model:
dataBM <- simulateTipData(modelBM, c(0,0,0,1))

#Fitting the model to the data
fitTipData(modelBM, dataBM, v=TRUE)

Description

~~ Methods for function fitTipData ~~

Methods

signature(object = "PhenotypicModel") This is the only method available for this function. Same behaviour for any PhenotypicModel.
fit_bd

Maximum likelihood fit of the general birth-death model

Description

Fits the birth-death model with potentially time-varying rates and potentially missing extant species to a phylogeny, by maximum likelihood. Notations follow Morlon et al. PNAS 2011.

Usage

```r
fit_bd(phylo, tot_time, f.lamb, f.mu, lamb_par, mu_par, f = 1,
       meth = "Nelder-Mead", cst.lamb = FALSE, cst.mu = FALSE,
       expo.lamb = FALSE, expo.mu = FALSE, fix.mu = FALSE,
       dt=0, cond = "crown")
```

Arguments

- **phylo**: an object of type 'phylo' (see ape documentation)
- **tot_time**: the age of the phylogeny (crown age, or stem age if known). If working with crown ages, tot_time is given by max(node.age(phylo)$ages).
- **f.lamb**: a function specifying the hypothesized functional form (e.g. constant, linear, exponential, etc.) of the variation of the speciation rate \( \lambda \) with time. Any functional form may be used. This function has two arguments: the first argument is time; the second argument is a numeric vector of the parameters of the time-variation (to be estimated).
- **f.mu**: a function specifying the hypothesized functional form (e.g. constant, linear, exponential, etc.) of the variation of the extinction rate \( \mu \) with time. Any functional form may be used. This function has two arguments: the first argument is time; the second argument is a numeric vector of the parameters of the time-variation (to be estimated).
- **lamb_par**: a numeric vector of initial values for the parameters of f.lamb to be estimated (these values are used by the optimization algorithm). The length of this vector is used to compute the total number of parameters in the model, so to fit a model with constant speciation rate (for example), lamb_par should be a vector of length 1. Otherwise aic values will be wrong.
- **mu_par**: a numeric vector of initial values for the parameters of f.mu to be estimated (these values are used by the optimization algorithm). The length of this vector is used to compute the total number of parameters in the model, so to fit a model without extinction (for example), mu_par should be empty (vector of length 0). Otherwise aic values will be wrong.
- **f**: the fraction of extant species included in the phylogeny
- **meth**: optimization to use to maximize the likelihood function, see optim for more details.
cst.lamb logical: should be set to TRUE only if f.lamb is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.

cst.m & logical: should be set to TRUE only if f.mu is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.

expo.lamb logical: should be set to TRUE only if f.lamb is exponential to use analytical instead of numerical computation in order to reduce computation time.

expo.m & logical: should be set to TRUE only if f.mu is exponential to use analytical instead of numerical computation in order to reduce computation time.

fix.m & logical: if set to TRUE, the extinction rate $\mu$ is fixed and will not be optimized.

dt the default value is 0. In this case, integrals in the likelihood are computed using R "integrate" function, which can be quite slow. If a positive dt is given as argument, integrals are computed using a piece-wise constant approximation, and dt represents the length of the intervals on which functions are assumed to be constant. For an exponential dependency of the speciation rate with time, we found that dt=1e-3 gives a good trade-off between precision and computation time.

cond conditioning to use to fit the model:
- FALSE: no conditioning (not recommended);
- "stem": conditioning on the survival of the stem lineage (use when the stem age is known, in this case tot_time should be the stem age);
- "crown" (default): conditioning on a speciation event at the crown age and survival of the 2 daughter lineages (use when the stem age is not known, in this case tot_time should be the crown age).

Details

The lengths of lamb_par and mu_par are used to compute the total number of parameters in the model, so to fit a model with constant speciation rate (for example), lamb_par should be a vector of length 1. Otherwise aic values will be wrong. In the f.lamb and f.mu functions, the first argument (time) runs from the present to the past. Hence, if the parameter controlling the variation of $\lambda$ with time is estimated to be positive (for example), this means that the speciation rate decreases from past to present.

Value

a list with the following components

- **model** the name of the fitted model
- **LH** the maximum log-likelihood value
- **aicc** the second order Akaike's Information Criterion
- **lamb_par** a numeric vector of estimated f.lamb parameters, in the same order as defined in f.lamb
- **mu_par** a numeric vector of estimated f.mu parameters, in the same order as defined in f.mu (if fix.mu is FALSE)
**Author(s)**
H Morlon

**References**

**See Also**
plot_fit_bd, plot_dtt, likelihood_bd

**Examples**

```r
# Some examples may take a little bit of time. Be patient!

data(Cetacea)
tot_time<-max(node.age(Cetacea)$ages)

# Fit the pure birth model (no extinction) with a constant speciation rate
f.lamb<-function(t,y)(y[1])
f.mu<-function(t,y)(0)
lamb_par<-c(0.09)
mu_par<-c()
#result_cst <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,
#                      f=87/89,cst.lamb=TRUE,fix.mu=TRUE,dt=1e-3)
#result_cst$model <- "pure birth with constant speciation rate"

# Fit the pure birth model (no extinction) with exponential variation
# of the speciation rate with time
f.mu<-function(t,y)(0)
lamb_par<-c(0.05, 0.01)
mu_par<-c()
#result_exp <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,
#                      f=87/89,expo.lamb=TRUE,fix.mu=TRUE,dt=1e-3)
#result_exp$model <- "pure birth with exponential variation in speciation rate"

# Fit the a pure birth model (no extinction) with linear variation of
# the speciation rate with time
f.mu<-function(t,y)(0)
lamb_par<-c(0.09, 0.001)
mu_par<-c()
#result_lin <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,f=87/89,fix.mu=TRUE,dt=1e-3)
#result_lin$model <- "pure birth with linear variation in speciation rate"

# Fit a birth-death model with exponential variation of the speciation
# rate with time and constant extinction
```
fit_coal_cst

Maximum likelihood fit of the equilibrium model

Description

Fits the equilibrium diversity model with potentially time-varying turnover rate and potentially missing extant species to a phylogeny, by maximum likelihood. The implementation allows only exponential time variation of the turnover rate, although this could be modified using expressions in Morlon et al. PLoSB 2010. Notations follow Morlon et al. PLoSB 2010.

Usage

fit_coal_cst(phylo, tau0 = 1e-2, gamma = 1, cst.rate = FALSE, meth = "Nelder-Mead", N0 = 0)

Arguments

phylo an object of type 'phylo' (see ape documentation)
tau0 initial value of the turnover rate at present (used by the optimization algorithm)
gamma initial value of the parameter controlling the exponential variation in turnover rate (used by the optimization algorithm)
cst.rate logical: should be set to TRUE to fit an equilibrium diversity model with time-constant turnover rate (know as the Hey model, model 1 in Morlon et al. PLoSB 2010). By default, a model with exponential time-varying rate exponential is fitted (model 2 in Morlon et al. PLoSB 2010).
meth optimization to use to maximize the likelihood function, see optim for more details.
N0 Number of extant species. With default value(0), N0 is set to the number of tips in the phylogeny. That is, the phylogeny is assumed to be 100% complete.
**Details**

This function fits models 1 (when cst.rate=TRUE) and 2 (when cst.rate=FALSE) from the PloSB 2010 paper. Likelihoods arising from these models are directly comparable to likelihoods from the fit_coal_var function, thus allowing to test support for equilibrium versus expanding diversity scenarios. Time runs from the present to the past. Hence, if gamma is estimated to be positive (for example), this means that the speciation rate decreases from past to present.

**Value**

A list with the following components:

- `model` the name of the fitted model
- `LH` the maximum log-likelihood value
- `aicc` the second order Akaike’s Information Criterion
- `tauP` the estimated turnover rate at present
- `gamma` the estimated parameter controlling the exponential variation in turnover rate (if cst.rate is FALSE)

**Author(s)**

H Morlon

**References**


**See Also**

likelihood_coal_cst, fit_coal_var

**Examples**

```r
data(Cetacea)
result <- fit_coal_cst(Cetacea, tauP=1.e-3, gamma=-1, cst.rate=FALSE, NQ=89)
print(result)
```
### Description

Fits the expanding diversity model with potentially time-varying rates and potentially missing extant species to a phylogeny, by maximum likelihood. The implementation allows only exponential time variation of the speciation and extinction rates, although this could be modified using expressions in Morlon et al. PloSB 2010. Notations follow Morlon et al. PLoSB 2010.

### Usage

```r
fit_coal_var(phylo, lamb0 = 0.1, alpha = 1, mu0 = 0.01, beta = 0,
              meth = "Nelder-Mead", NO = 0, cst.lamb = FALSE, cst.mu = FALSE,
              fix.eps = FALSE, mu.0 = FALSE, pos = TRUE)
```

### Arguments

- **phylo**: an object of type 'phylo' (see ape documentation)
- **lamb0**: initial value of the speciation rate at present (used by the optimization algorithm)
- **alpha**: initial value of the parameter controlling the exponential variation in speciation rate (used by the optimization algorithm)
- **mu0**: initial value of the extinction rate at present (used by the optimization algorithm)
- **beta**: initial value of the parameter controlling the exponential variation in extinction rate.
- **meth**: optimization to use to maximize the likelihood function, see `optim` for more details.
- **NO**: Number of extant species. With default value(0), N0 is set to the number of tips in the phylogeny. That is, the phylogeny is assumed to be 100% complete.
- **cst.lamb**: logical: should be set to TRUE only if f.lamb is constant (i.e. does not depend on time, models 3, 4b & 5 in Morlon et al. PloSB 2010) to use analytical instead of numerical computation in order to reduce computation time.
- **cst.mu**: logical: should be set to TRUE only if f.mu is constant (i.e. does not depend on time, models 3 & 4a in Morlon et al. PloSB 2010) to use analytical instead of numerical computation in order to reduce computation time.
- **fix.eps**: logical: should be set to TRUE only if the extinction fraction is constant (i.e. does not depend on time, model 4c in Morlon et al. PloSB 2010)
- **mu.0**: logical: should be set to TRUE to force the extinction rate to 0 (models 5 & 6 in Morlon et al. PloSB 2010)
- **pos**: logical: should be set to FALSE only to not enforce positive speciation and extinction rates
Details

The function fits models 3 to 6 from the PloSB 2010 paper. Likelihoods arising from these models are computed using the coalescent approximation and are directly comparable to likelihoods from the `fit_coal_cst` function, thus allowing to test support for equilibrium versus expanding diversity scenarios.

These models can be fitted using the options specified below:

- model 3:
  with `cst.lamb=TRUE` & `cst.mu=TRUE`
- model 4a:
  with `cst.lamb=FALSE` & `cst.mu=TRUE`
- model 4b:
  with `cst.lamb=TRUE` & `cst.mu=FALSE`
- model 4c:
  with `cst.lamb=FALSE`, `cst.mu=FALSE` & `fix.eps=TRUE`
- model 4d:
  with `cst.lamb=FALSE`, `cst.mu=FALSE` & `fix.eps=FALSE`
- model 5:
  with `cst.lamb=TRUE` & `mu.0=TRUE`
- model 6:
  with `cst.lamb=FALSE` & `mu.0=TRUE`

Time runs from the present to the past. Hence, if alpha is estimated to be positive (for example), this means that the speciation rate decreases from past to present.

Value

A list with the following components:

- `model` the name of the fitted model
- `lh` the maximum log-likelihood value
- `aicc` the second order Akaike’s Information Criterion
- `model.parameters` the estimated parameters

Author(s)

H Morlon

References

See Also

likelihood_coal_var, fit_coal_cst

Examples

data(Cetacea)
result <- fit_coal_var(Cetacea, lamb0=0.01, alpha=-0.001, mu0=0.0, beta=0, N0=89)
print(result)

---

fit_env

Maximum likelihood fit of the environmental birth-death model

Description

Fits the environmental birth-death model with potentially missing extant species to a phylogeny, by maximum likelihood. Notations follow Morlon et al. PNAS 2011 and Condamine et al. ELE 2013.

Usage

fit_env(phylo, env_data, tot_time, f.lamb, f.mu, lamb_par, mu_par, df= NULL, f = 1, 
meth = "Nelder-Mead", cst.lamb = FALSE, cst.mu = FALSE, 
expo.lamb = FALSE, expo.mu = FALSE, fix.mu = FALSE, 
dt=0, cond = "crown")

Arguments

phylo an object of type ’phylo’ (see ape documentation)
env_data environmental data, given as a data frame with two columns. The first column is time, the second column is the environmental data (temperature for instance).
tot_time the age of the phylogeny (crown age, or stem age if known). If working with crown ages, tot_time is given by max(node.age(phylo)$ages).
f.lamb a function specifying the hypothesized functional form of the variation of the speciation rate \( \lambda \) with time and the environmental variable. Any functional form may be used. This function has three arguments: the first argument is time; the second argument is the environmental variable; the third argument is a numeric vector of the parameters controlling the time and environmental variation (to be estimated).
f.mu a function specifying the hypothesized functional form of the variation of the extinction rate \( \mu \) with time and the environmental variable. Any functional form may be used. This function has three arguments: the first argument is time; the second argument is the environmental variable; the second argument is a numeric vector of the parameters controlling the time and environmental variation (to be estimated).
**fit_env**

- **lamb_par**: a numeric vector of initial values for the parameters of \( f.lamb \) to be estimated (these values are used by the optimization algorithm). The length of this vector is used to compute the total number of parameters in the model, so to fit a model with constant speciation rate (for example), `lamb_par` should be a vector of length 1. Otherwise `aic` values will be wrong.

- **mu_par**: a numeric vector of initial values for the parameters of \( f.mu \) to be estimated (these values are used by the optimization algorithm). The length of this vector is used to compute the total number of parameters in the model, so to fit a model without extinction (for example), `mu_par` should be empty (vector of length 0). Otherwise `aic` values will be wrong.

- **df**: the degree of freedom to use to define the spline. As a default, `smooth.spline(env_data[,1], env_data[,2])$df` is used. See `sm.spline` for details.

- **f**: the fraction of extant species included in the phylogeny

- **meth**: optimization to use to maximize the likelihood function, see `optim` for more details.

- **cst.lamb**: logical: should be set to `TRUE` only if \( f.lamb \) is constant (i.e. does not depend on time or the environmental variable) to use analytical instead of numerical computation in order to reduce computation time.

- **cst.mu**: logical: should be set to `TRUE` only if \( f.mu \) is constant (i.e. does not depend on time or the environmental variable) to use analytical instead of numerical computation in order to reduce computation time.

- **expo.lamb**: logical: should be set to `TRUE` only if \( f.lamb \) is an exponential function of time (and does not depend on the environmental variable) to use analytical instead of numerical computation in order to reduce computation time.

- **expo.mu**: logical: should be set to `TRUE` only if \( f.mu \) is an exponential function of time (and does not depend on the environmental variable) to use analytical instead of numerical computation in order to reduce computation time.

- **fix.mu**: logical: if set to `TRUE`, the extinction rate \( \mu \) is fixed and will not be optimized.

- **dt**: the default value is 0. In this case, integrals in the likelihood are computed using R "integrate" function, which can be quite slow. If a positive `dt` is given as argument, integrals are computed using a piece-wise constant approximation, and `dt` represents the length of the intervals on which functions are assumed to be constant. We found that `1e-3` generally provides a good trade-off between precision and computation time.

- **cond**: conditioning to use to fit the model:
  - `FALSE`: no conditioning (not recommended);
  - "stem": conditioning on the survival of the stem lineage (use when the stem age is known, in this case `tot_time` should be the stem age);
  - "crown" (default): conditioning on a speciation event at the crown age and survival of the 2 daughter lineages (use when the stem age is not known, in this case `tot_time` should be the crown age).

**Details**

The lengths of `lamb_par` and `mu_par` are used to compute the total number of parameters in the model, so to fit a model with constant speciation rate (for example), `lamb_par` should be a vector of
length 1. Otherwise aic values will be wrong. In the f.lamb and f.mu functions, time runs from the present to the past.

Value

a list with the following components

- **model** the name of the fitted model
- **lh** the maximum log-likelihood value
- **aicc** the second order Akaike’s Information Criterion
- **lamb_par** a numeric vector of estimated f.lamb parameters, in the same order as defined in f.lamb
- **mu_par** a numeric vector of estimated f.mu parameters, in the same order as defined in f.mu (if fix.mu is FALSE)

Note

The speed of convergence of the fit might depend on the degree of freedom chosen to define the spline.

Author(s)

H Morlon and F Condamine

References


See Also

- `plot_fit_env`, `fit_bd`, `likelihood_bd`

Examples

```r
data(Cetacea)
tot_time<-max(node.age(Cetacea)$ages)
data(InfTemp)
dof<-smooth.spline(InfTemp[,1], InfTemp[,2])$df

# Fits a model with lambda varying as an exponential function of temperature
# and mu fixed to 0 (no extinction). Here t stands for time and x for temperature.
f.lamb <-function(t,x,y){y[1] * exp(y[2] * x)}
f.mu<-function(t,x,y){0}
lamb_par<-c(0.10, 0.01)
mu_par<-c()
```
fit_sgd

Maximum likelihood fit of the SGD model

Description

Fits the SGD model with exponential growth of the metacommunity, by maximum likelihood. Notations follow Manceau et al. (2015)

Usage

fit_sgd(phylo, tot_time, par, f=1, meth = "Nelder-Mead")

Arguments

- **phylo**: an object of type 'phylo' (see ape documentation)
- **tot_time**: the age of the phylogeny (crown age, or stem age if known). If working with crown ages, tot_time is given by max(node.age(phylo)$ages)
- **par**: a numeric vector of initial values for the parameters (b,d,nu) to be estimated (these values are used by the optimization algorithm)
- **f**: the fraction of extant species included in the phylogeny
- **meth**: optimization to use to maximize the likelihood function, see *optim* for more details.

Value

a list with the following components

- **model**: the name of the fitted model
- **LH**: the maximum log-likelihood value
- **aicc**: the second order Akaike’s Information Criterion
- **par**: a numeric vector of estimated values of b (birth), b-d (growth) and nu (mutation)

Note

While b-d and nu can in general be well estimated, the likelihood surface is quite flat with respect to b, such that the estimated b can vary a lot depending on the choice of the initial parameter values. Estimates of b should not be trusted.

Author(s)

M Manceau
References


See Also

likelihood_sgd

Examples

# Some examples may take a little bit of time. Be patient!
data(Calomys)
tot_time <- max(node.age(Calomys)$ages)
par_init <- c(1e7,1e7-0.5,1)
#fit_sgd(Calomys, tot_time, par_init, f=11/13)

fit_t_comp

Fits models of trait evolution incorporating competitive interactions

Description

Fits matching competition (MC), diversity dependent linear (DDlin), or diversity dependent exponential (DDexp) models of trait evolution to a given dataset and phylogeny.

Usage

fit_t_comp(phylol, dataL model=c("MC","DDexp","DDlin"), pars=NULL, geography.object=NULL)

Arguments

phylo an object of type 'phylo' (see ape documentation)
data a named vector of trait values with names matching phylo$tip.label
model model chosen to fit trait data, "MC" is the matching competition model of Nuismer & Harmon 2014, "DDlin" is the diversity-dependent linear model, and "DDexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.

geography.object a list of sympatry through time created using CreateGeoObject

pars vector specifying starting parameter values for maximum likelihood optimization. If unspecified, default values are used (see Details)
**Details**

If unspecified, `par` takes the default values of `var(data)/max(nodeHeights(phylo))` for `sig2` and 0 for either `S` for the matching competition model, `b` for the linear diversity dependence model, or `r` for the exponential diversity dependence model. Values can be manually entered as a vector with the first element equal to the desired starting value for `sig2` and the second value equal to the desired starting value for either `S`, `b`, or `r`. Note: since likelihood optimization uses `sig` rather than `sig2`, and since the starting value for is exponentiated to stabilize the likelihood search, if you input a `par` value, the first value specifying `sig2` should be the `log(sqrt())` of the desired `sig2` starting value.

**Value**

A list with the following elements:

- `LH`: maximum log-likelihood value
- `aic`: Akaike Information Criterion value
- `aicc`: AIC value corrected for small sample size
- `free.parameters`: number of free parameters from the model
- `sig2`: maximum-likelihood estimate of `sig2` parameter
- `S`: maximum-likelihood estimate of `S` parameter of matching competition model (see Note)
- `b`: maximum-likelihood estimate of `b` parameter of linear diversity dependence model
- `r`: maximum-likelihood estimate of `r` parameter of exponential diversity dependence model
- `z0`: maximum-likelihood estimate of `z0`, the value at the root of the tree
- `convergence`: convergence diagnostics from `optim` function (see `optim` documentation)

**Note**

In current version, the `S` parameter is restricted to take on negative values in MC + geography ML optimization.

**Author(s)**

Jonathan Drury jonathan.p.drury@gmail.com
Julien Clavel

**References**


fit_t_comp_subgroup

Fits models of trait evolution incorporating competitive interactions, restricting competition to occur only between members of a subgroup.

**Description**

Fits matching competition (MC), diversity dependent linear (DDlin), or diversity dependent exponential (DDexp) models of trait evolution to a given dataset, phylogeny, and stochastic maps of both subgroup membership and biogeography.

**Usage**

```r
fit_t_comp_subgroup(full.phylo, ana.events, clado.events, stratified=FALSE, map, data, trim.class, model=c("MC","DDexp","DDlin"), par=NULL, method="Nelder-Mead", bounds=NULL)
```

**Arguments**

- `full.phylo`: an object of type 'phylo' (see ape documentation) containing all of the tips used to estimate ancestral biogeography in BioGeoBEARS
- `ana.events`: the "ana.events" table produced in BioGeoBEARS that lists anagenetic events in the stochastic map

**Examples**

```r
data(Anolis.data)
geography.object<-Anolis.data$geography.object
pPC1<-Anolis.data$data
phylo<-Anolis.data$phylo

#Fit three models without biogeography to pPC1 data
MC.fit<-fit_t_comp(phylo, pPC1, model="MC")
DDlin.fit<-fit_t_comp(phylo, pPC1, model="DDlin")
DDexp.fit<-fit_t_comp(phylo, pPC1, model="DDexp")

#Now fit models that incorporate biogeography, NOTE these models take longer to fit
MC.geo.fit<-fit_t_comp(phylo, pPC1, model="MC", geography.object=geography.object)
DDlin.geo.fit<-fit_t_comp(phylo, pPC1, model="DDlin", geography.object=geography.object)
DDexp.geo.fit<-fit_t_comp(phylo, pPC1, model="DDexp", geography.object=geography.object)
```
clamp.events  the "clamp.events" table produced in BioGeoBEARS that lists cladogenetic events in the stochastic map
stratified  logical indicating whether the stochastic map was built from a stratified analysis in BioGeoBEARS
map  a phylo object created using `make.simmap` in phytools that contains reconstructed subgroup membership
data  a named vector of trait values for subgroup members with names matching `full.phylo$tip.label`
trim.class  subgroup whose members are competing
model  model chosen to fit trait data, "MC" is the matching competition model of Nuismer & Harmon 2014, "DDlin" is the diversity-dependent linear model, and "DDexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.
par  vector specifying starting parameter values for maximum likelihood optimization. If unspecified, default values are used (see Details)
method  optimization algorithm to use (see `optim`)
bounds  (optional) list of bounds to pass to optimization algorithm (see details at `optim`)

Details

If unspecified, par takes the default values of \( \text{var(data)/max(nodeHeights(phylo))} \) for sig2 and 0 for either S for the matching competition model, b for the linear diversity dependence model, or r for the exponential diversity dependence model. Values can be manually entered as a vector with the first element equal to the desired starting value for sig2 and the second value equal to the desired starting value for either S, b, or r. Note: since likelihood optimization uses sig rather than sig2, and since the starting value for is exponentiated to stabilize the likelihood search, if you input a par value, the first value specifying sig2 should be the log(sqrt()) of the desired sig2 starting value.

Value

a list with the following elements:

- LH  maximum log-likelihood value
- aic  Akaike Information Criterion value
- aicc  AIC value corrected for small sample size
- free.parameters  number of free parameters from the model
- sig2  maximum-likelihood estimate of sig2 parameter
- S  maximum-likelihood estimate of S parameter of matching competition model (see Note)
- b  maximum-likelihood estimate of b parameter of linear diversity dependence model
- r  maximum-likelihood estimate of r parameter of exponential diversity dependence model
- z0  maximum-likelihood estimate of z0, the value at the root of the tree
- convergence  convergence diagnostics from `optim` function (see optim documentation)
Note

In current version, the S parameter is restricted to take on negative values in MC + geography ML optimization.

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References


See Also

likelihood_subgroup_model CreateGeobyClassObject

Examples

data(BGB.examples)

Canidae.phylo<-BGB.examples$Canidae.phylo

names(dummy.group)<-Canidae.phylo$tip.label

Canidae.simmap<-make.simmap(Canidae.phylo,dummy.group)

set.seed(123)
Canidae.data<-rnorm(length(Canidae.phylo$tip.label))

names(Canidae.data)<-Canidae.phylo$tip.label
Canidae.A<-Canidae.data[which(dummy.group=="A")]

fitA<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,ana.events=BGB.examples$Canidae.ana.events, clado.events=BGB.examples$Canidae.clado.events,stratified=FALSE,map=Canidae.simmap, data=Canidae.A,trim.class="A",model="DDexp")
**fit_t_env**

*Maximum likelihood fit of the environmental model of trait evolution*

**Description**

Fits model of trait evolution for which evolutionary rates depends on an environmental function, or more generally a time varying function.

**Usage**

```r
fit_t_env(phylo, data, env_data, error=NULL, model=c("EnvExp", "EnvLin"), method="Nelder-Mead", control=list(maxit=20000), ...)
```

**Arguments**

- `phylo` An object of class 'phylo' (see ape documentation)
- `data` A named vector of phenotypic trait values.
- `env_data` Environmental data, given as a time continuous function (see, e.g. splinefun) or a data frame with two columns. The first column is time, the second column is the environmental data (temperature for instance).
- `error` A named vector with standard error (SE) for each species. Default is NULL, if NA, then the SE is estimated from the data.
- `model` The model describing the functional form of variation of the evolutionary rate $\sigma^2$ with time and the environmental variable. Default models are "EnvExp" and "EnvLin" (see details). An user defined function of any functional form may be used (forward in time). This function has three arguments: the first argument is time; the second argument is the environmental variable; the third argument is a numeric vector of the parameters controlling the time and environmental variation (to be estimated). See the example below.
- `method` Methods used by the optimization routine (see ?optim for details).
- `control` Max. bound for the number of iteration of the optimizer; other options can be fixed on the list (see ?optim).
- `...` Arguments to be passed to the function. See details.

**Details**

`fit_t_env` allows fitting environmental models of trait evolution. The default models EnvExp and EnvLin represents models for which the evolutionary rates are changing as a function of environmental changes though times as defined below.

EnvExp:

$$\sigma^2(t) = \sigma_0^2 e^{(\beta T(t))}$$

EnvLin:

$$\sigma^2(t) = \sigma_0^2 + \beta T(t)$$
Users defined models should have the following form (see also examples below):

```r
fun <- function(t, env, param){ param*env(t)}
```

$t$: is the time parameter.

$env$: is a time function of an environmental variable. See for instance object created by `splinefun` when interpolating coordinate of points.

$param$: is a vector of parameters to estimate.

For instance, the `EnvExp` function can be coded as:

```r
fun <- function(t, env, param){ param[1]*exp(param[2]*env(t))}
```

where `param[1]` is the $\sigma^2$ parameter and `param[2]` is the $\beta$ parameter. Note that in this later case, two starting values should be provided in the `param` argument.

e.g.:

```r
sigma=0.1
beta=0
```

```r
fit_t_env(tree, data, env_data=InfTemp, model=fun, param=c(sigma,beta))
```

The various options are passed through "...".

- `param`: The starting values used for the model. Must match the total number of parameters of the specified models. If "error=NA", a starting value for the SE to be estimated must be provided with user-defined models.
- `scale`: scale the amplitude of the environmental curve between 0 and 1. This may improve the parameters search in some situations.
- `df`: the degree of freedom to use for defining the spline. As a default, `smooth.spline(env_data[,1], env_data[,2])$df` is used. See `sm.spline` for details.
- `upper`: the upper bound for the parameter search when the "L-BFGS-B" method is used. See `optim` for details.
- `lower`: the lower bound for the parameter search when the "L-BFGS-B" method is used. See `optim` for details.
- `sig2`: can be used instead of `param` to define the starting sigma value only
- `beta`: can be used instead of `param` to define the beta starting value only
- `maxdiff`: difference in time between tips and present day for phylogenetic trees with no contemporaneous species (default is 0)

**Value**

A list with the following components:

- `LH`: the maximum log-likelihood value
- `aic`: the Akaike’s Information Criterion
- `aicc`: the second order Akaike’s Information Criterion
- `free.parameters`: the number of estimated parameters
param a numeric vector of estimated parameters, sigma and beta respectively for the defaults models. In the same order as defined by the user if a customized model is provided

root the estimated root value

convergence convergence status of the optimizing function; "0" indicates convergence (See ?optim for details)

hess.value reliability of the likelihood estimates calculated through the eigen-decomposition of the hessian matrix. "0" means that a reliable estimate has been reached

env.func the environmental function

tot.time the root age of the tree

model the fitted model (default models or user specified)

SE the estimated SE for species mean when "error=NA"

Note

The users defined function is evaluated forward in time i.e.: from the root to the tips (time = 0 at the (present) tips). The speed of convergence of the fit might depend on the degree of freedom chosen to define the spline.

Author(s)

J. Clavel

References


See Also

plot.fit_t.env, likelihood_t_env

Examples

data(Cetacea)
data(InfTemp)

# Simulate a trait with temperature dependence on the Cetacean tree
set.seed(123)

trait <- sim_t_env(Cetacea, param=c(0.1,-0.2), env_data=InfTemp, model="EnvExp",
root.value=0, step=0.001, plot=TRUE)

## Fit the Environmental-exponential model
# Fit the environmental model
result1=fit_t_env(Cetacea, trait, env_data=InfTemp, scale=TRUE)
plot(result1)

# Add to the plot the results from different smoothing of the temperature curve
result2<-fit_t_env(Cetacea, trait, env_data=InfTemp, df=10, scale=TRUE)
lines(result2, col="red")

result3<-fit_t_env(Cetacea, trait, env_data=InfTemp, df=50, scale=TRUE)
lines(result3, col="blue")

## Fit the environmental linear model

fit_t_env(Cetacea, trait, env_data=InfTemp, model="EnvLin", df=50, scale=TRUE)

## Fit user defined model (note that several other environmental variables
can be simultaneously encapsulated in this function through the env argument)

# We define the function for the model
my_fun<-function(t, env_cont, param){
  param[1]*exp(param[2]*env_cont(t))
}

res<-fit_t_env(Cetacea, trait, env_data=InfTemp, model=my_fun,
  param=c(0.1,0), scale=TRUE)
# Retrieve the parameters and compare to 'result1'
res
plot(res, col="red")

## Fit user defined environmental function

require(pspline)
spline_result <- sm.spline(x=InfTemp[,1],y=InfTemp[,2], df=50)
env_func <- function(t)(predict(spline_result,t))
t<-unique(InfTemp[,1])

# We build the interpolated smoothing spline function
env_data<-splinefun(t,env_func(t))

# We then fit the model
fit_t_env(Cetacea, trait, env_data=env_data)

## Various parameterization (box constraints, df, scaling of the curve...) example
fit_t_env(Cetacea, trait, env_data=InfTemp, model="EnvLin", method="L-BFGS-B",
  scale=TRUE, lower=-30, upper=20, df=10)

## A very general model...

# We define the function for the Early-Burst/AC model:
maxtime = max(branching.times(Cetacea))

# sigma^2 * e^(rt)
my_fun_ebac <- function(t, env_cont, param){
  time = (maxtime - t)
  param[1]*exp(param[2]*time)
}
fit_t_pl

res <- fit_t_env(Cetacea, trait, env_data=InfTemp, model=my_fun_ebac,
                 param=c(0.1,0), scale=TRUE)
res # note that "r" is positive: it's the AC model (~OU model on ultrametric tree)

---

High-dimensional phylogenetic models of trait evolution

Description

Fits high-dimensional model of trait evolution on trees through penalized likelihood. A phylogenetic Leave-One-Out Cross-Validated log-likelihood (LOOCV) is used to estimate model parameters.

Usage

fit_t_pl(Y, tree, model=c("BM", "OU", "EB", "lambda"),
          method=c("RidgeAlt", "RidgeArch", "RidgeAltapprox",
                   "LASSO", "LASSOapprox"),
          targM=c("null", "Variance", "unitVariance"),
          REML=TRUE, up=NULL, low=NULL,
          tol=NULL, starting=NULL, SE=NULL,
          scale.height=TRUE, ...)

Arguments

Y A matrix of phenotypic traits values (the variables are represented as columns)
tree An object of class 'phylo' (see ape documentation)
model The evolutionary model. "BM" is Brownian Motion, "OU" is Ornstein-Uhlenbeck,
       "EB" is Early Burst, and "lambda" is Pagel's lambda transformation.
method The penalty method. "RidgeArch": Archetype (linear) Ridge penalty, "RidgeAlt":
         Quadratic Ridge penalty, "LASSO": Least Absolute Selection and Shrinkage
         Operator. "RidgeAltapprox" and "LASSOapprox" are fast approximations of
         the LOOCV for the Ridge quadratic and LASSO penalties.
targM The target matrix used for the Ridge regularizations. "null" is a null target,
        "Variance" for a diagonal unequal variance target, "unitVariance" for an equal
        diagonal target. Only works with "RidgeArch", "RidgeAlt", and "RidgeAltap-
        prox" methods.
REML Use REML (default) or ML for estimating the parameters.
up Upper bound for the parameter search of the evolutionary model (optional).
low Lower bound for the parameter search of the evolutionary model (optional).
tol minimum value for the regularization parameter. Singularities can occur with a zero value in high-dimensional cases. (default is NULL)

starting Starting values for the parameter search (optional).

SE Standard errors associated with values in Y. If TRUE, SE will be estimated.

scale.height Whether the tree should be scaled to unit length or not. (default is TRUE)

... Options to be passed through. (e.g., echo=FALSE to stop printing messages)

Details

fit_t_pl allows fitting various multivariate evolutionary models to high-dimensional datasets (where the number of variables \( p \) is larger than \( n \)). Models estimates are more accurate than maximum likelihood methods. Models fit can be compared using the GIC criterion (see ?GIC). Details about the methods are described in Clavel et al. (2018).

Value

a list with the following components

loocv the (negative) cross-validated penalized likelihood

model.par the evolutionary model parameter estimates

gamma the regularization/tuning parameter of the penalized likelihood

corrstruct a list with the transformed variables and the phylogenetic tree with branch length stretched to the model estimated parameters

model the evolutionary model

method the penalization method

p the number of traits

n the number of species

targM the target used for Ridge Penalization

R a list with the estimated evolutionary covariance matrix and it’s inverse

REML logical indicating if the REML (TRUE) or ML (FALSE) method has been used

variables Y is the input dataset and tree is the input phylogenetic tree

SE the estimated standard error

Note

The LASSO is computationally intensive. Please wait! For highly-dimensional datasets you should favor the "RidgeArch" method to speed up the computations. The Ridge penalties with "null" or "unitVariance" targets are rotation invariants.

Author(s)

J. Clavel
foraminifera

References


See Also

ancestral, phyl.pca_pl, GIC.fit_pl_rpanda, gic_criterion mvgl

Examples

```
set.seed(1)
n <- 32 # number of species
p <- 31 # number of traits
tree <- pbtree(n=n) # phylogenetic tree
R <- Posdef(matrix(p)) # a random symmetric matrix (variance)

# simulate a dataset
Y <- mvSIM(tree, model="BM", nsim=1, param=list(sigma=R))

# fit the model
fit_t_pl(Y, tree, model="BM", method="Bridgealt")

# try on rotated axis (using pca)
trans <- prcomp(Y, center=FALSE)
fit_t_pl(trans$x, tree, model="BM", method="Bridgealt")

# Estimate the SE (similar to Pagel's lambda for BM).
# Advised with empirical datasets
fit_t_pl(Y, tree, model="BM", method="Bridgealt", SE=TRUE)
```

foraminifera

Foraminifera diversity since the Jurassic

Description

Foraminifera fossil diversity since the Jurassic

Usage

data(foraminifera)
getDatosLikelihood

Details

Foraminifera fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the two following variables:

- age: a numeric vector corresponding to the geological age, in Myrs before the present
- foraminifera: a numeric vector corresponding to the estimated foraminifera change at that age

References


Examples

data(foraminifera)
plot(foraminifera)

getDatosLikelihood Likelihood of tip trait values.

Description

Computes -log( likelihood ) of tip trait data under a given set of parameters, and for a specified model of trait evolution.

Usage

getDatosLikelihood(object, data, params, v)

Arguments

- object: an object of class 'PhenotypicModel'.
- data: vector of tip trait data.
- params: vector of parameters, given in the same order as in the 'model' object.
- v: boolean specifying the verbose mode. Default value : FALSE.

Value

A numerical value : -log( likelihood ) of the model.

Author(s)

M Manceau
References

Examples

#Loading an example tree

#Creating the models
modelBM <- createModel(tree, 'BM')

dataBM <- simulateTipData(modelBM, c(0,0,0,1))

#Likelihood of the data:
getDataLikelihood(modelBM, dataBM, c(0,0,0,1))

---

Description

getTipDistribution Distribution of tip trait values.

Usage

getTipDistribution(object, params, v)
Arguments

object an object of class 'PhenotypicModel'
params vector of parameters, given in the same order as in the 'model' object.
\( v \) boolean specifying the verbose mode. Default value: FALSE.

Value

mean Expectation vector of the tip trait distribution.
Sigma Variance-covariance matrix of the tip trait distribution.

Author(s)

M Manceau

References


Examples

# Loading an example tree
tree <- read.tree(text=newick)

# Creating a BM model
modelBM <- createModel(tree, 'BM')

# Tip trait distribution under the model:
getTipDistribution(modelBM, c(0,0,0,1))

getTipDistribution-methods

Distribution of tip trait values.

Description

Computes the mean and variance of the tip trait distribution under a specified model of trait evolution.

Methods

signature(object = "PhenotypicModel") In the most general case, this function computes the expectation vector and the variance-covariance matrix using a numerical integration procedure that may take time.

signature(object = "PhenotypicACDC") The function has been optimized for this subclass.
signature(object = "PhenotypicADiag") The function has been optimized for this subclass.
signature(object = "PhenotypicBM") The function has been optimized for this subclass.
signature(object = "PhenotypicDD") The function has been optimized for this subclass.
signature(object = "PhenotypicGMM") The function has been optimized for this subclass.
signature(object = "PhenotypicOU") The function has been optimized for this subclass.
signature(object = "PhenotypicPM") The function has been optimized for this subclass.

References


---

**GIC.fit_pl.rpanda**

Generalized Information Criterion (GIC) to compare models fit by Maximum Likelihood (ML) or Penalized Likelihood (PL).

**Description**

The GIC allows comparing models fit by Maximum Likelihood (ML) or Penalized Likelihood (PL).

**Usage**

```r
## S3 method for class 'fit_pl.rpanda'
GIC(object, ...)
```

**Arguments**

- `object` An object of class "fit_pl.rpanda". See ?fit_t_pl
- `...` Options to be passed through.

**Details**

GIC allows comparing the fit of various models estimated by Penalized Likelihood (see ?fit_t_pl).

It's a wrapper to the `gic_criterion` function.

**Value**

A list with the following components:

- `LogLikelihood` the log-likelihood estimated for the model with estimated parameters
- `GIC` the GIC criterion
- `bias` the value of the bias term estimated to compute the GIC
Author(s)

J. Clavel

References


See Also

gic_criterion, fit_t_pl, mvglS

Examples

```r
set.seed(1)
n <- 32 # number of species
p <- 40 # number of traits
tree <- pbtree(n=n) # phylogenetic tree
R <- Posdef(p) # a random symmetric matrix (covariance)
# simulate a dataset
Y <- mvSIM(tree, model="BM1", nsim=1, param=list(sigma=R))

fit1 <- fit_t_pl(Y, tree, model="BM", method="RidgeAlt")
fit2 <- fit_t_pl(Y, tree, model="OU", method="RidgeAlt")

GIC(fit1); GIC(fit2)
```

---

gic_criterion

`Generalized Information Criterion (GIC) to compare models fit by Maximum Likelihood (ML) or Penalized Likelihood (PL).`

Description

The GIC allows comparing models fit by Maximum Likelihood (ML) or Penalized Likelihood (PL).

Usage

```r
gic_criterion(Y, tree, model="BM", method=c("RidgeAlt", "RidgeArch", "LASSO", "ML", "RidgeAltapprox", "LASSOapprox"), targM=c("null", "Variance", "unitVariance"), param=NULL,
```

Arguments

Y
  A matrix of phenotypic traits values (the variables are represented as columns)

tree
  An object of class 'phylo' (see ape documentation)

model
  The evolutionary model, "BM" is Brownian Motion, "OU" is Ornstein-Uhlenbeck, 
  "EB" is Early Burst, and "lambda" is Pagel's lambda transformation.

method
  The penalty method. "RidgeArch": Archetype (linear) Ridge penalty, "RidgeAlt": 
  Quadratic Ridge penalty, "LASSO": Least Absolute Selection and Shrinkage 
  Operator, "ML": Maximum Likelihood.

targM
  The target matrix used for the Ridge regularizations. "null" is a null target, 
  "Variance" for a diagonal unequal variance target, "unitVariance" for an equal 
  diagonal target. Only works with "RidgeArch", "RidgeAlt" methods.

param
  Parameter for the evolutionary model (see "model" above).

tuning
  The tuning/regularization parameter.

REML
  Use REML (default) or ML for estimating the parameters.

... Additional options. Not used yet.

Details

gic_criterion allows comparing the fit of various models estimated by Penalized Likelihood (see 
?fit_t_pl). Use the wrapper GIC instead for models fit with fit_t_pl.

Value

  a list with the following components

LogLikelihood  the log-likelihood estimated for the model with estimated parameters

GIC            the GIC criterion

bias           the value of the bias term estimated to compute the GIC

Note

  The tuning parameter is assumed to be zero when using the "ML" method.

Author(s)

  J. Clavel
References


See Also

GIC.fit_pl_rpanda, fit_t_pl

Examples

```r
set.seed(123)
n <- 32 # number of species
p <- 2 # number of traits
tree <- pbtree(n=n) # phylogenetic tree
R <- Posdef(p) # a random symmetric matrix (covariance)

# simulate a dataset
Y <- mvSIM(tree, model="BM1", nsim=1, param=list(sigma=R))

# Compute the GIC for ML
gic_criterion(Y, tree, model="BM", method="ML", tuning=0) # ML

# Compare with PL?
#test <- fit_t_pl(Y, tree, model="BM", method="RidgeAlt")
#GIC(test)
```

---

**greenalgae**  
*Green algae diversity since the Jurassic*

Description

Green algae fossil diversity since the Jurassic

Usage

```r
data(greenalgae)
```
Details

Green algae fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the two following variables:

- age: a numeric vector corresponding to the geological age, in Myrs before the present
- greenalgae: a numeric vector corresponding to the estimated green algae change at that age

References


Examples

data(greenalgae)
plot(greenalgae)

Description

Paleotemperature data across the Cenozoic inferred from delta O18 measurements

Usage

data(InfTemp)

Details

Paleotemperature data inferred from delta 018 measurements using the equation of Epstein et al. (1953). The format is a dataframe with the two following variables:

- Age: a numeric vector corresponding to the geological age, in Myrs before the present
- Temperature: a numeric vector corresponding to the inferred temperature at that age

References

Examples

data(InfTemp)
pplot(InfTemp)

JSDtree

Jensen-Shannon distance between phylogenies

Description

Computes the Jensen-Shannon distance metric between spectral density profiles of phylogenies.

Usage

JSDtree(phylo, meth=c("standard"))

Arguments

- **phylo**: a list of objects of type 'phylo' (see ape documentation)
- **meth**: the method used to compute the spectral density, which can either be "standard", "normal1", or "normal2". if set to "normal1", computes the spectral density normalized to the degree matrix. if set to "normal2", computes the spectral density normalized to the number of eigenvalues. if set to "standard", computes the unnormalized version of the spectral density (see the associated paper for an explanation)

Value

a matrix providing the Jensen-Shannon distance values between phylogeny pairs

Author(s)

E Lewitus

References

Lewitus, E., Morlon, H., Characterizing and comparing phylogenies from their Laplacian spectrum, bioRxiv doi: http://dx.doi.org/10.1101/026476

See Also

JSDtree_cluster, spectR, BICompare

Examples

trees<-TESS::tess.sim.age(n=20, age=10, 0.15, 0.05, MRCA=TRUE)
JSDtree(trees)
Description
Clusters phylogenies using hierarchical and k-medoids clustering

Usage
JSDtree_cluster(JSDtree, alpha=0.9, draw=T)

Arguments
JSDtree a matrix of distances between phylogenies pairs, typically the output of the JS-Dtree function when the distance is measured as the Jensen-Shannon distance
alpha the confidence value for demarcating clusters in the hierarchical clustering plot; the default is 0.9
draw plot heatmap and hierarchical cluster in new windows

Value
plots a heatmap and a hierarchical cluster with bootstrap support, and outputs results of the k-medoids clustering in the form of a list with the following components
clusters the optimal number of clusters around medoids (see pamk documentation)
cluster_assignments assignments of trees to clusters
cluster_support a list with the following components: widths: a table specifying the cluster to which each tree belongs, the neighbor (i.e. most similar) cluster, and the silhouette width of the observation (see silhouette documentation); clus.avg.widths: average silhouette width for each cluster; vg.width: average silhouette width across all clusters

Note
The k-medoids clustering may not work with fewer than 10 trees

Author(s)
E Lewitus

References
Lewitus, E., Morlon, H., Characterizing and comparing phylogenies from their Laplacian spectrum, bioRxiv doi: http://dx.doi.org/10.1101/026476
landplant

See Also

JSDtree

Examples

trees<-TES::tess.sim.age(n=20,age=10,0.15,0.05,MRCA=TRUE)
res<-JSDtree(trees)
#JSDtree_cluster(res,alpha=0.9,draw=T)

Description

Land plant fossil diversity since the Jurassic

Usage

data(landplant)

Details

Land plant fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the two following variables:

age  a numeric vector corresponding to the geological age, in Myrs before the present
landplant  a numeric vector corresponding to the estimated land plant change at that age

References


Examples

data(landplant)
plot(landplant)
likelihood.bd

Description
Computes the likelihood of a phylogeny under a birth-death model with potentially time-varying rates and potentially missing extant species. Notations follow Morlon et al. PNAS 2011.

Usage
likelihood.bd(phylo, tot_time, f.lamb, f.mu, f, cst.lamb = FALSE, cst.mu = FALSE, expo.lamb = FALSE, expo.mu = FALSE, dt=0, cond = "crown")

Arguments
phylo
an object of type ’phylo’ (see ape documentation)
tot_time
the age of the phylogeny (crown age, or stem age if known). If working with crown ages, tot_time is given by max(node.age(phylo)$ages).
f.lamb
a function specifying the time-variation of the speciation rate \( \lambda \). This function as a single argument (time). Any function may be used.
f.mu
a function specifying the time-variation of the speciation rate \( \mu \). This function as a single argument (time). Any function may be used.
f
the fraction of extant species included in the phylogeny
cst.lamb
logical: should be set to TRUE only if f.lamb is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.
cst.mu
logical: should be set to TRUE only if f.mu is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.
expo.lamb
logical: should be set to TRUE only if f.lamb is exponential to use analytical instead of numerical computation in order to reduce computation time.
expo.mu
logical: should be set to TRUE only if f.mu is exponential to use analytical instead of numerical computation in order to reduce computation time.
dt
the default value is 0. In this case, integrals in the likelihood are computed using R "integrate" function, which can be quite slow. If a positive dt is given as argument, integrals are computed using a piece-wise contant approximation, and dt represents the length of the intervals on which functions are assumed to be constant. For an exponential dependency of the speciation rate with time, we found that dt=1e-3 gives a good trade-off between precision and computation time.
cond
conditioning to use to fit the model:
- FALSE: no conditioning (not recommended);
- "stem": conditioning on the survival of the stem lineage (use when the stem age is known, in this case tot_time should be the stem age);
- "crown" (default): conditioning on a speciation event at the crown age and survival of the 2 daughter lineages (use when the stem age is not known, in this case tot_time should be the crown age).

Details

When specifying f.lamb and f.mu, time runs from the present to the past (hence if the speciation rate decreases with time, f.lamb must be a positive function of time).

Value

the loglikelihood value of the phylogeny, given f.lamb and f.mu

Author(s)

H Morlon

References


Examples

data(Cetacea)
tot_time <- max(node.age(Cetacea)$ages)
# Compute the likelihood for a pure birth model (no extinction) with
# an exponential variation of speciation rate with time
lamb_par <- c(0.1, 0.01)
f.mu <- function(t){0}
f <- 87/89
lh <- likelihood_bd(Cetacea,tot_time,f.lamb,f.mu,f,cst.mu=TRUE,expo.lamb=TRUE, dt=1e-3)

\[
\text{likelihood\_coal\_cst} \quad \text{Likelihood of a phylogeny under the equilibrium diversity model}
\]

Description

Computes the likelihood of a phylogeny under the equilibrium diversity model with potentially time-varying rates and potentially missing extant species. Notations follow Morlon et al. PloSB 2010.

Usage

likelihood_coal_cst(Vtimes, ntips, tau0, gamma, N0)
Arguments

Vtimes a vector of branching times (sorted from present to past)
ntips the number of tips in the phylogeny
tau0 the turnover rate at present
gamma the parameter controlling the exponential variation in turnover rate. With gamma=0, the turnover rate is constant over time.
N0 the number of extant species

Details

Time runs from the present to the past. Hence, a positive gamma (for example) means that the turnover rate declines from past to present.

Value

a list containing the following components:

res the loglikelihood value of the phylogeny, given tau0 and gamma
all vector of all the individual loglikelihood values corresponding to each branching event

Author(s)

H Morlon

References


Examples

data(Cetacea)
Vtimes <- sort(branching.times(Cetacea))
tau0 <- 0.1
gamma <- 0.001
ntips <- Ntip(Cetacea)
N0 <- 89
likelihood <- likelihood_coal_cst(Vtimes,ntips,tau0,gamma,N0)
likelihood_coal_var  Likelihood of a birth-death model using a coalescent approach

Description
Computes the likelihood of a phylogeny under the expanding diversity model with potentially time-varying rates and potentially missing extant species to a phylogeny. Notations follow Morlon et al. PloSB 2010.

Usage
likelihood_coal_var(Vtimes, ntips, lamb0, alpha, mu0, beta, N0, pos = TRUE)

Arguments
- Vtimes: a vector of branching times (sorted from present to past)
- ntips: number of species in the phylogeny
- lamb0: the speciation rate at present
- alpha: the parameter controlling the exponential variation in speciation rate.
- mu0: the extinction rate at present
- beta: the parameter controlling the exponential variation in extinction rate.
- N0: the number of extinct species
- pos: logical: should be set to FALSE only to not enforce positive speciation and extinction rates

Details
Time runs from the present to the past. Hence, a positive alpha (for example) means that the speciation rate declines from past to present.

Value
a list containing the following components:
- res: the loglikelihood value of the phylogeny, given the parameters
- all: vector of all the individual loglikelihood values corresponding to each branching event

Author(s)
H Morlon

References
**likelihood_sgd**

*Likelihood of a phylogeny under the SGD model*

**Description**

Computes the likelihood of a phylogeny under the SGD model with exponential increasing of the metacommunity, and potentially missing extant species. Notations follow Manceau et al. (2015).

**Usage**

```r
likelihood_sgd(phylo, tot_time, b, d, nu, f)
```

**Arguments**

- `phylo`: an object of type 'phylo' (see ape documentation)
- `tot_time`: the age of the phylogeny (crown age, or stem age if known). If working with crown ages, tot_time is given by max(node.age(phylo)$ages).
- `b`: the (constant) birth rate of individuals in the model.
- `d`: the (constant) death rate of individuals in the model.
- `nu`: the (constant) mutation rate of individuals in the model.
- `f`: the fraction of extant species included in the phylogeny

**Value**

The likelihood value of the phylogeny, given the model and the parameter values `b`, `d`, `nu`.

**Author(s)**

M Manceau

**References**

Examples

```r
data(Cetacea)
tot_time <- max(node.age(Cetacea)$ages)
b <- 1e6
d <- 1e6-0.5
nu <- 0.6
f <- 87/89
# lh <- likelihood_sgd(Cetacea, tot_time, b, d, nu, f)
```

likelihood_subgroup_model

*Likelihood of a dataset under models with biogeography fit to a subgroup.*

Description

Computes the likelihood of a dataset under either the linear or exponential diversity dependent model with specified \( \sigma^2 \) and slope values and with a geography.object formed using `CreateGeoObject`.

Usage

```r
likelihood_subgroup_model(data, phylo, geography.object, model=c("MC","DDe","DDlin"), par, return.z0=FALSE, maxN=NULL)
```

Arguments

- `phylo`: an object of type 'phylo' (see ape documentation) produced as "map" from CreateGeobyClassObject. NB: the length of this object need not match number of items in data, since map may include tips outside of group with some part of their branch in the group.
- `data`: a named vector of continuous data for a subgroup of interest with names corresponding to phylo$tip.label.
- `geography.object`: a list of sympatry/group membership through time created using `CreateGeobyClassObject`.
- `model`: model chosen to fit trait data, "DDe" is the diversity-dependent linear model, and "DDexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.
- `par`: a vector listing a value for \( \log(\sigma^2) \) (see Note) and either \( b \) (for the linear diversity dependent model) or \( r \) (for the exponential diversity dependent model), in that order.
- `return.z0`: logical indicating whether to return an estimate of the trait value at the root given the parameter values (if TRUE, function returns root value rather than negative log-likelihood).
- `maxN`: when fitting DDlin model, it is necessary to specify the maximum number of sympatric lineages to ensure that the rate returned does not correspond to negative \( \sigma^2 \) values at any point in time (see Details).
Details

When specifying par, \( \log(\text{sig2}) \) (see Note) must be listed before the slope parameter \((b\ or\ r)\). 
\[ \text{maxN can be calculated using } \text{maxN} = \text{max} (\text{vapply(geo.object$geography.object, function(x)max(rowSums(x)),1)}) \]
where geo.object is the output of \texttt{CreateGeoObject} 

Value

The negative log-likelihood value of the dataset (accordingly, the negative of the output should be recorded as the likelihood), given the phylogeny, \( \text{sig2} \) and slope values, and \texttt{geo.object}.

If \texttt{return.z0=TRUE}, the estimated root value for the par values is returned instead of the negative log-likelihood.

Note

To stabilize optimization, this function exponentiates the input \( \text{sig2} \) value, thus the user must input the \( \log(\text{sig2}) \) value to compute the correct log likelihood (see example).

Author(s)

Jonathan Drury \texttt{jonathan.p.drury@gmail.com}
Julien Clavel

References


See Also

\texttt{fit_t_comp CreateGeoObject likelihood_t_DD}

Examples

data(BGB.examples)

\begin{verbatim}
Canidae.phylo<-BGB.examples$Canidae.phylo
names(dummy.group)<-Canidae.phylo$tip.label

Canidae.simmap<-make.simmap(Canidae.phylo, dummy.group)
set.seed(123)
Canidae.data<-rnorm(length(Canidae.phylo$tip.label))
names(Canidae.data)<-Canidae.phylo$tip.label
\end{verbatim}
Canidae.A<-Canidae.data[which(dummy.group=="A")]
Canidae.geobyclass.object<-CreateGeoByClassObject(phylo=Canidae.phylo,
simmap=Canidae.simmap, trim.class="A", ana.events=BGB.examples$Canidae.ana.events,
clado.events=BGB.examples$Canidae.clado.events,stratified=FALSE, rnd=5)

par <- c(log(0.01),-0.0005)
maxN<-max(vapply(Canidae.geobyclass.object$geography.object$geography.object,
function(x)max(rowSums(x),1)))

lh <- -likelihood_subgroup_model(data=Canidae.A, phylo=Canidae.geobyclass.object$map,
geo.object=Canidae.geobyclass.object$geography.object, model="DDlin", par=par,
return.z0=FALSE, maxN=maxN)

---

**likelihood_t_DD**  
*Likelihood of a dataset under diversity-dependent models.*

**Description**

Computes the likelihood of a dataset under either the linear or exponential diversity dependent model with specified sigma2 and slope values.

**Usage**

`likelihood_t_DD(phylo, data, par, model=c("DDlin","DDexp"))`

**Arguments**

- **phylo**: an object of type 'phylo' (see ape documentation)
- **data**: a named vector of continuous data with names corresponding to phylo$tip.label
- **par**: a vector listing a value for log(sig2) (see Note) and either b (for the linear diversity dependent model) or r (for the exponential diversity dependent model), in that order.
- **model**: model chosen to fit trait data, "DDlin" is the diversity-dependent linear model, and "DDexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.

**Details**

When specifying par, log(sig2) must be listed before the slope parameter (b or r).

**Value**

the negative log-likelihood value of the dataset (accordingly, the negative of the output should be recorded as the likelihood), given the phylogeny and sig2 and slope values.
Note

To stabilize optimization, this function exponentiates the input \(\text{sig2}\) value, thus the user must input the \(\log(\text{sig2})\) value to compute the correct log likelihood (see example).

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com
Julien Clavel

References


See Also

fit_t_comp, likelihood_t_DD_geog

Examples

```r
data(Anolis.data)
phylo <- Anolis.data$phylo
pPC1 <- Anolis.data$data

# Compute the likelihood that the r value is twice the ML estimate for the DDexp model
par <- c(0.08148371, (2*-0.3223835))
lh <- -likelihood_t_DD(phylo,pPC1,par,model="DDexp")
```

```
likelihood_t_DD_geog  Likelihood of a dataset under diversity-dependent models with biogeography.
```

Description

Computes the likelihood of a dataset under either the linear or exponential diversity dependent model with specified \(\text{sigmaR}\) and slope values and with a geography\_object formed using `CreateGeoObject`.

Usage

```r
likelihood_t_DD_geog(phylo, data, par, geo.object, model=c("DDlin","DDexp"),maxN=NA)
```
Arguments

- **phylo**: an object of type 'phylo' (see ape documentation)
- **data**: a named vector of continuous data with names corresponding to `phylo$tip.label`
- **par**: a vector listing a value for \( \log(\text{sig2}) \) (see Note) and either \( b \) (for the linear diversity dependent model) or \( r \) (for the exponential diversity dependent model), in that order.
- **geo.object**: a list of sympatry through time created using `CreateGeoObject`
- **model**: model chosen to fit trait data, "Dlin" is the diversity-dependent linear model, and "Dexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.
- **maxN**: when fitting Dlin model, it is necessary to specify the maximum number of sympatric lineages to ensure that the rate returned does not correspond to negative sig2 values at any point in time (see Details).

Details

When specifying **par**, \( \log(\text{sig2}) \) (see Note) must be listed before the slope parameter (\( b \) or \( r \)).

**maxN** can be calculated using \[
\text{maxN} = \max(\text{vapply(geo.object$geography.object, function(x)max(rowSums(x)),1)}),
\]
where **geo.object** is the output of `CreateGeoObject`

Value

the negative log-likelihood value of the dataset (accordingly, the negative of the output should be recorded as the likelihood), given the phylogeny, sig2 and slope values, and **geo.object**.

Note

To stabilize optimization, this function exponentiates the input sig2 value, thus the user must input the \( \log(\text{sig2}) \) value to compute the correct log likelihood (see example).

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com
Julien Clavel

References


See Also

`fit_t_comp`, `CreateGeoObject`, `likelihood_t_DD`
### Examples

```r
    data(Anolis.data)
    phylo <- Anolis.data$phylo
    pPC1 <- Anolis.data$data
    geography.object <- Anolis.data$geography.object

    # Compute the likelihood with geography using ML parameters for fit without geography
    par <- c(log(0.01153294),-0.0006692378)
    maxN<-max(vapply(geography.object$geography.object,function(x)max(rowSums(x)),1))
    lh <- -likelihood_t_DD_geog(phylo,pPC1,par,geography.object,model="DDlin",maxN=maxN)
```

### Description

Computes the likelihood of a dataset under either the linear or exponential environmental model, or an user defined environmental model. This function is used internally by `fit_t_env`.

### Usage

```r
    likelihood_t_env(phylo, data, model=c("EnvExp", "EnvLin"), ...)
```

### Arguments

- **phylo**: an object of class 'phylo' (see ape documentation)
- **data**: a named vector of continuous data with names corresponding to `phylo$tip.label`
- ...:
  - param: a vector with the parameters used in the environmental function. The first value is `sigR` and the second is `beta`.
  - fun: a time continuous function of an environmental variable (see e.g. `?fit_t_env`)
  - times: a vector of branching times starting at zero (e.g. `max(branching.times(phylo))`)
  - mtot: root age of the tree (e.g. `max(branching.times(phylo))`)
  - error: a vector of standard error (se) for each species

  If the "times" argument is not provided, the "phylo" object is used to compute it as well as "mtot".

  Note that the argument "mu" can be used to specify the root state (e.g. when using an mcmc sampler)

- **model**: model chosen to fit trait data, "EnvExp" is the exponential-environmental model, and "EnvLin" is the linear-environmental model. Otherwise, an user specified model can be provided.

### Details

The "fun" argument can be filled by an environmental dataframe.
**Value**

the log-likelihood value of the environmental model

**Author(s)**

Julien Clavel

**References**


**See Also**

`fit_t_env`

**Examples**

```r
data(Cetacea)
data(InfTemp)

# Simulate a trait with temperature dependence on the Cetacean tree
set.seed(123)

trait <- sim_t_env(Cetacea, param=c(0.1, -0.2), env_data=InfTemp, model="EnvExp", root.value=0, step=0.001, plot=TRUE)

# Compute the likelihood
likelihood_t_env(Cetacea, trait, param=c(0.1, 0), fun=InfTemp, model="EnvExp")

# Provide the times
brtime <- branching.times(Cetacea)
mot <- max(brtime)
times <- mot - brtime

likelihood_t_env(Cetacea, trait, param=c(0.1, 0), fun=InfTemp, times=times, mtot=mot, model="EnvExp")

# Provide the environmental function rather than the dataset (faster if used recursively)
spline_result <- sm.spline(InfTemp[,1], InfTemp[,2], df=50)
env_func <- function(t){predict(spline_result,t)}
t <- unique(InfTemp[,1])
# We build the interpolated smoothing spline function
env_data <- splinefun(t, env_func(t))

likelihood_t_env(Cetacea, trait, param=c(0.1, 0), fun=env_data, times=times, mtot=mot, model="EnvExp")
```
**likelihood_t_MC**

*Likelihood of a dataset under the matching competition model.*

**Description**

Computes the likelihood of a dataset under the matching competition model with specified sigma2 and S values.

**Usage**

`likelihood_t_MC(phylo, data, par)`

**Arguments**

- `phylo`: an object of type 'phylo' (see ape documentation)
- `data`: a named vector of continuous data with names corresponding to `phylo$tip.label`
- `par`: a vector listing a value for `log(sigR)` (see Note) and S (parameters of the matching competition model), in that order

**Details**

When specifying `par`, `log(sig2)` must be listed before S.

**Value**

the negative log-likelihood value of the dataset (accordingly, the negative of the output should be recorded as the likelihood), given the phylogeny and sig2 and S values

**Note**

To stabilize optimization, this function exponentiates the input sig2 value, thus the user must input the log(sig2) value to compute the correct log likelihood (see example).

**Author(s)**

Jonathan Drury jonathan.p.drury@gmail.com
Julien Clavel

**References**


likelihood_t_MC_geog

See Also

fit_t_comp likelihood_t_MC_geog

Examples

data(Anolis.data)
phylo <- Anolis.data$phylo
pPC1 <- Anolis.data$data

# Compute the likelihood that the S value is twice the ML estimate
par <- c(0.0003139751, 2*0.06387258)
lh <- -likelihood_t_MC(phylo, pPC1, par)

likelihood_t_MC_geog  Likelihood of a dataset under the matching competition model with biogeography.

Description

Computes the likelihood of a dataset under the matching competition model with specified sigma2 and S values and with a geography.object formed using CreateGeoObject.

Usage

likelihood_t_MC_geog(phylo, data, par, geo.object)

Arguments

phylo  an object of type 'phylo' (see ape documentation)
data  a named vector of continuous data with names corresponding to phylo$tip.label
par  a vector listing a value for log(sig2) (see Note) and S (parameters of the matching competition model), in that order
geo.object  a geography object indicating sympatry through time, created using CreateGeoObject

Details

When specifying par, log(sig2) must be listed before S.

Value

the negative log-likelihood value of the dataset (accordingly, the negative of the output should be recorded as the likelihood), given the phylogeny, sig2 and S values, and geography.object.

Note

S must be negative (if it is positive, the likelihood function will multiply input by -1). To stabilize optimization, this function exponentiates the input sig2 value, thus the user must input the log(sig2) value to compute the correct log likelihood (see example).
Author(s)
Jonathon Drury jonathan.p.drury@gmail.com
Julien Clavel

References

See Also
fit_t_comp CreateGeoObject likelihood_t_MC

Examples
data(Anolis.data)
phylo <- Anolis.data$phylo
pPC1 <- Anolis.data$data
geography.object <- Anolis.data$geography.object

# Compute the likelihood with geography using ML parameters for fit without geography
par <- c(0.0003139751, -0.06387258)
lh <- -likelihood_t_MC_geog(phylo,pPC1,par,geography.object)

lines.fit_t.env Add to a plot line segments joining the phenotypic evolutionary rate through time estimated by the fit_t_env function

Description
Plot estimated evolutionary rate as a function of the environmental data and time.

Usage
## S3 method for class 'fit_t.env'
lines(x, steps = 100, ...)

Arguments
x an object of class 'fit_t.env' obtained from a fit_t_env fit.
steps the number of steps from the root to the present used to compute the evolutionary rate $\sigma^2$ through time.
... further arguments to be passed to plot. See ?plot.
Value

`lines.fit_t.env` returns invisibly a list with the following components used to add the line segments to the current plot:

- **time_steps**: the times steps where the climatic function was evaluated to compute the rate. The number of steps is controlled through the argument `steps`.
- **rates**: the estimated evolutionary rate through time estimated at each `time_steps`.

Note

All the graphical parameters (see `par`) can be passed through (e.g. line type: `lty`, line width: `lwd`, color: `col` ...)

Author(s)

J. Clavel

References


See Also

- `plot.fit_t.env`
- `likelihood_t.env`

Examples

```r
data(Cetacea)
data(InfTemp)

# Plot estimated evolutionary rate as a function of the environmental data and time.
set.seed(123)
trait <- sim_t_env(Cetacea, param=c(0.1,-0.2), env_data=InfTemp, model="EnvExp",
                   root.value=0, step=0.01, plot=TRUE)

## Fit the Environmental-exponential model with different smoothing parameters
result1=fit_t_env(Cetacea, trait, env_data=InfTemp, scale=TRUE)
result2=fit_t_env(Cetacea, trait, env_data=InfTemp, scale=TRUE, df=10)

# first plot result1
plot(result1, lwd=3)

# add result2 to the current plot
lines(result2, lty=2, lwd=3, col="red")
```
Description

For each model taken as input, fits the model and returns its AIC value in a recap table.

Usage

modelSelection(object, data)

Arguments

- object: a vector of objects of class 'PhenotypicModel'.
- data: vector of tip trait data.

Details

Warning: This function relies on the standard R optimizer "optim". It may not always converge well. Please double check the convergence by trying distinct parameter sets for the initialisation.

Value

A recap table presenting the AIC value of each model.

Author(s)

M Manceau

References

PhenotypicACDC-class

---

### `ostracoda`

**Ostracod diversity since the Jurassic**

---

**Description**

Ostracod fossil diversity since the Jurassic

**Usage**

```r
data(sealevel)
```

**Details**

Ostracod fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the following variables:

- `age` a numeric vector corresponding to the geological age, in Myrs before the present
- `ostracoda` a numeric vector corresponding to the estimated ostracod change at that age

**References**


**Examples**

```r
data(ostracoda)
plot(ostracoda)
```

---

**PhenotypicACDC-class**  
*Class* "PhenotypicACDC"

---

**Description**

Subclass of the PhenotypicModel class intended to represent the model of ACcelerating or DeCel-erating phenotypic evolution.

**Objects from the Class**

Objects can be created by calls of the form `new("PhenotypicACDC", ...).`
---

**PhenotypicACDC-class**

**Slots**

- `matrixCoalescenceTimes`: Object of class "matrix"
- `name`: Object of class "character"
- `period`: Object of class "numeric"
- `aAGamma`: Object of class "function"
- `numbersCopy`: Object of class "numeric"
- `numbersPaste`: Object of class "numeric"
- `initialCondition`: Object of class "function"
- `paramsNames`: Object of class "character"
- `constraints`: Object of class "function"
- `params0`: Object of class "numeric"
- `tipLabels`: Object of class "character"
- `tipLabelsSimu`: Object of class "character"
- `comment`: Object of class "character"

**Extends**

Class "PhenotypicModel", directly.

**Methods**

- **getTipDistribution** signature(object = "PhenotypicACDC"): ... 

**Author(s)**

Marc Manceau

**References**


**Examples**

`showClass("PhenotypicACDC")`
Description

A subclass of the PhenotypicModel class, intended to represent models of phenotypic evolution with a "A" matrix diagonalizable.

Objects from the Class

Objects can be created by calls of the form `new("PhenotypicADiag", ...)`.

Slots

- name: Object of class "character"
- period: Object of class "numeric"
- aAGamma: Object of class "function"
- numbersCopy: Object of class "numeric"
- numbersPaste: Object of class "numeric"
- initialCondition: Object of class "function"
- paramsNames: Object of class "character"
- constraints: Object of class "function"
- params0: Object of class "numeric"
- tipLabels: Object of class "character"
- tipLabelsSimu: Object of class "character"
- comment: Object of class "character"

Extends

Class "PhenotypicModel", directly.

Methods

- `getTipDistribution` signature(object = "PhenotypicADiag"): ...

Author(s)

Marc Manceau

References

**PhenotypicBM-class**

**Examples**
```
showClass("PhenotypicADiag")
```

---

**Description**
A subclass of the PhenotypicModel class, intended to represent the model of Brownian phenotypic evolution.

**Objects from the Class**
Objects can be created by calls of the form new("PhenotypicBM", ...).

**Slots**
- `matrixCoalescenceTimes`: Object of class "matrix"
- `name`: Object of class "character"
- `period`: Object of class "numeric"
- `aAGamma`: Object of class "function"
- `numbersCopy`: Object of class "numeric"
- `numbersPaste`: Object of class "numeric"
- `initialCondition`: Object of class "function"
- `paramsNames`: Object of class "character"
- `constraints`: Object of class "function"
- `params0`: Object of class "numeric"
- `tipLabels`: Object of class "character"
- `tipLabelsSimu`: Object of class "character"
- `comment`: Object of class "character"

**Extends**
Class "PhenotypicModel", directly.

**Methods**
- `getTipDistribution` signature(object = "PhenotypicBM"): ...

**Author(s)**
Marc Manceau
References

Examples

```
showClass("PhenotypicBM")
```

Description
A subclass of the PhenotypicModel class, intended to represent the model of Density-Dependent phenotypic evolution.

Objects from the Class
Objects can be created by calls of the form `new("PhenotypicDD", ...)`.

Slots
- `matrixCoalescenceJ`: Object of class "matrix"
- `nLivingLineages`: Object of class "numeric"
- `name`: Object of class "character"
- `period`: Object of class "numeric"
- `aAGamma`: Object of class "function"
- `numbersCopy`: Object of class "numeric"
- `numbersPaste`: Object of class "numeric"
- `initialCondition`: Object of class "function"
- `paramsNames`: Object of class "character"
- `constraints`: Object of class "function"
- `params0`: Object of class "numeric"
- `tipLabels`: Object of class "character"
- `tipLabelsSimu`: Object of class "character"
- `comment`: Object of class "character"

Extends
Class "PhenotypicModel", directly.
PhenotypicGMM-class

Methods

getTipDistribution signature(object = "PhenotypicDD"): ...

Author(s)

Marc Manceau

References


Examples

showClass("PhenotypicDD")

PhenotypicGMM-class	Class "PhenotypicGMM"

Description

A subclass of the PhenotypicModel class, intended to represent the Generalist Matching Mutualism model of phenotypic evolution. This is a model of phenotypic evolution with interactions between two clades, running on two trees.

Objects from the Class

Objects can be created by calls of the form new("PhenotypicGMM", ...).

Slots

n1: Object of class "numeric" ~~
n2: Object of class "numeric" ~~
name: Object of class "character" ~~
period: Object of class "numeric" ~~
aAGamma: Object of class "function" ~~
numbersCopy: Object of class "numeric" ~~
numbersPaste: Object of class "numeric" ~~
initialCondition: Object of class "function" ~~
paramsNames: Object of class "character" ~~
constraints: Object of class "function" ~~
params0: Object of class "numeric" ~~
tipLabels: Object of class "character" ~~
tipLabelsSimu: Object of class "character" ~~
comment: Object of class "character" ~~
Extends
   Class "PhenotypicModel", directly.

Methods
   getTipDistribution signature(object = "PhenotypicGMM"): ...

Author(s)
   Marc Manceau

References

Examples
   showClass("PhenotypicGMM")

Description
   This class describes a model of phenotypic evolution running on a phylogenetic tree, with or without interactions between lineages.

Objects from the Class
   Objects can be created by calls of the form new("PhenotypicModel", ...). Alternatively, you may just want to use the "createModel" function for predefined models.

Slots
   name: Object of class "character" ~~
   period: Object of class "numeric" ~~
   aAGamma: Object of class "function" ~~
   numbersCopy: Object of class "numeric" ~~
   numbersPaste: Object of class "numeric" ~~
   initialCondition: Object of class "function" ~~
   paramsNames: Object of class "character" ~~
   constraints: Object of class "function" ~~
   params0: Object of class "numeric" ~~
tipLabels: Object of class "character" ~~
tipLabelsSimu: Object of class "character" ~~
comment: Object of class "character" ~~

Methods

[<- signature(x = "PhenotypicModel", i = "ANY", j = "ANY", value = "ANY"): ...
[ signature(x = "PhenotypicModel", i = "ANY", j = "ANY", drop = "ANY"): ...
fitTipData signature(object = "PhenotypicModel"): ...
getDataLikelihood signature(object = "PhenotypicModel"): ...
getTipDistribution signature(object = "PhenotypicModel"): ...
modelSelection signature(object = "PhenotypicModel"): ...
print signature(x = "PhenotypicModel"): ...
show signature(object = "PhenotypicModel"): ...
simulateTipData signature(object = "PhenotypicModel"): ...

Author(s)

Marc Manceau

References


Examples

showClass("PhenotypicModel")

PhenotypicOU-class Class "PhenotypicOU"

Description

A subclass of the PhenotypicModel class, intended to represent the Ornstein-Uhlenbeck model of phenotypic evolution.

Objects from the Class

Objects can be created by calls of the form new("PhenotypicOU", ...).
Slots

matrixCoalescenceTimes: Object of class "matrix" ~~
name: Object of class "character" ~~
period: Object of class "numeric" ~~
aAGamma: Object of class "function" ~~
numbersCopy: Object of class "numeric" ~~
numbersPaste: Object of class "numeric" ~~
initialCondition: Object of class "function" ~~
paramsNames: Object of class "character" ~~
constraints: Object of class "function" ~~
params0: Object of class "numeric" ~~
tipLabels: Object of class "character" ~~
tipLabelsSimu: Object of class "character" ~~
comment: Object of class "character" ~~

Extends

Class "PhenotypicModel", directly.

Methods

getTipDistribution signature(object = "PhenotypicOU"): ... 

Author(s)

Marc Manceau

References


Examples

showClass("PhenotypicOU")
PhenotypicPM-class

PhenotypicPM-class  Class "PhenotypicPM"

Description
A subclass of the PhenotypicModel class, intended to represent the Phenotypic Matching model of phenotypic evolution, by Nuismer and Harmon (Eco Lett, 2014).

Objects from the Class
Objects can be created by calls of the form `new("PhenotypicPM", ...)`. 

Slots
- name: Object of class "character" ~
- period: Object of class "numeric" ~
- aAGamma: Object of class "function" ~
- numbersCopy: Object of class "numeric" ~
- numbersPaste: Object of class "numeric" ~
- initialCondition: Object of class "function" ~
- paramsNames: Object of class "character" ~
- constraints: Object of class "function" ~
- params0: Object of class "numeric" ~
- tipLabels: Object of class "character" ~
- tipLabelsSimu: Object of class "character" ~
- comment: Object of class "character" ~

Extends
Class "PhenotypicModel", directly.

Methods
- `getTipDistribution` signature(object = "PhenotypicPM"): ...

Author(s)
Marc Manceau

References
Examples

showClass("PhenotypicPM")

Phocoenidae  Phocoenidae phylogeny

Description

Ultrametric phylogenetic tree of the 6 extant Phocoenidae (porpoise) species

Usage

data(Phocoenidae)

Details

This phylogeny was extracted from Steeman et al. Syst Bio 2009 cetacean phylogeny

References


Examples

data(Phocoenidae)
print(Phocoenidae)
plot(Phocoenidae)

phyl.pca_pl  Regularized Phylogenetic Principal Component Analysis (PCA).

Description

Performs a principal component analysis (PCA) on a regularized evolutionary variance-covariance matrix obtained using the fit_t_pl function.

Usage

phyl.pca_pl(object, plot=TRUE, ...)

**Arguments**

- **object**: A penalized likelihood model fit obtained by the `fit_t_pl` function.
- **plot**: Plot of the PC’s axes. Default is TRUE (see details).
- **...**: Options to be passed through. (e.g., axes=c(1,2), col, pch, cex, mode="cov" or "corr", etc.)

**Details**

`phyl.pca_pl` allows computing a phylogenetic principal component analysis (following Revell 2009) using a regularized evolutionary variance-covariance matrix from penalized likelihood models fit to high-dimensional datasets (where the number of variables p is potentially larger than n; see details for the models options in `fit_t_pl`). Models estimates are more accurate than maximum likelihood methods, particularly in the high-dimensional case. Plotting options, the number of axes to display (axes=c(1,2) is the default), and whether the covariance (mode="cov") or correlation (mode="corr") should be used can be specified through the ellipsis "..." argument.

**Value**

A list with the following components:

- **values**: the eigenvalues of the evolutionary variance-covariance matrix
- **scores**: the PC scores
- **loadings**: the component loadings
- **nodes_scores**: the scores for the ancestral states at the nodes (projected on the space of the tips)
- **mean**: the mean/ancestral value used to center the data
- **vectors**: the eigenvectors of the evolutionary variance-covariance matrix

**Note**

Contrary to conventional PCA, the principal axes of the phylogenetic PCA are not orthogonal, they represent the main axes of (independent) evolutionary changes.

**Author(s)**

J. Clavel

**References**


**See Also**

`fit_t_pl, ancestral, GIC.fit_pl.rpanda, gic_criterion`
Examples

```r
set.seed(1)
n <- 32 # number of species
p <- 31 # number of traits
tree <- pbtree(n=n) # phylogenetic tree
R <- Posdef(p) # a random symmetric matrix (covariance)

# simulate a dataset
Y <- mvSIM(tree, model="BM1", nsim=1, param=list(sigma=R))

# fit a multivariate Pagel lambda model with Penalized likelihood
fit <- fit_t_pl(Y, tree, model="lambda", method="RidgeAlt")

# Perform a phylogenetic PCA using the model fit (Pagel lambda model)
pca_results <- phyl.pca_pl(fit, plot=TRUE)

# retrieve the scores
head(pca_results$scores)
```

**Phyllostomidae**

*Phyllostomidae phylogeny*

**Description**

Ultrametric phylogenetic tree of 150 of the 165 extant known Phyllostomidae species

**Usage**

```
data(Phyllostomidae)
```

**Details**

This phylogeny is the maximum clade credibility tree used in Rolland et al. (2014), which originally comes from the Bininda-Emonds tree (Bininda-Emonds et al. 2007).

**References**


**See Also**

*Phyllostomidae_genera*
Phyllostomidae_genera

Examples

data(Phyllostomidae)
print(Phyllostomidae)
#plot(Phyllostomidae)

---

Phyllostomidae_genera  Phylogenies of Phyllostomidae genera

Description

List of 25 ultrametric phylogenetic trees corresponding to 25 Phyllostomidae genera

Usage

data(Phyllostomidae_genera)

See Also

Phyllostomidae

Examples

data(Phyllostomidae_genera)
print(Phyllostomidae_genera)

---

plot.fit_t.env  

Plot the phenotypic evolutionary rate through time estimated by the `fit_t.env` function

Description

Plot estimated evolutionary rate as a function of the environmental data and time.

Usage

```r
## S3 method for class 'fit_t.env'
plot(x, steps = 100, ...)```

Arguments

- `x`: an object of class 'fit_t.env' obtained from a `fit_t.env` fit.
- `steps`: the number of steps from the root to the present used to compute the evolutionary rate $\sigma^2$ through time.
- `...`: further arguments to be passed to `plot`. See `?plot`. 

plot.fit_t.env

Value

plot.fit_t.env returns invisibly a list with the following components used in the current plot:

- **time_steps**: the times steps where the climatic function was evaluated to compute the rate. The number of steps is controlled through the argument `steps`.
- **rates**: the estimated evolutionary rate through time estimated at each `time_steps`.

Note

All the graphical parameters (see `par`) can be passed through (e.g. line type: `lty`, line width: `lwd`, color: `col` ...)

Author(s)

J. Clavel

References


See Also

- `lines.fit_t.env`, `likelihood_t_env`

Examples

data(Cetacea)
data(InfTemp)

# Simulate a trait with temperature dependence on the Cetacean tree
set.seed(123)

trait <- sim_t_env(Cetacea, param=c(0.1, 0.2), env_data=InfTemp, model="EnvExp", root.value=0, step=0.01, plot=TRUE)

## Fit the Environmental-exponential model

result1 <- fit_t_env(Cetacea, trait, env_data=InfTemp, scale=TRUE)
plot(result1)

# Further options
plot(result1, lty=2, lwd=2, col="red")
plot_BICompare

Display modalities on a phylogeny.

Description
Plot a phylogeny with branches colored according to modalities.

Usage
plot_BICompare(phylo, BICompare)

Arguments
- phylo: an object of type 'phylo' (see ape documentation)
- BICompare: an object of class 'BICompare', output of the 'BICompare' function

Value
a plot of the phylogeny with branches colored according to which modalities they belong to.

Author(s)
E Lewitus

References
Lewitus, E., Morlon, H., Characterizing and comparing phylogenies from their Laplacian spectrum, bioRxiv doi: http://dx.doi.org/10.1101/026476

See Also
BICompare

Examples

data(Cetacea)
#result <- BICompare(Cetacea,5)
#plot_BICompare(Cetacea, result)
plot_dtt  

---

**plot_dtt**

*Plot diversity through time*

---

**Description**

Plot the estimated number of species through time

**Usage**

```r
plot_dtt(fit.bd, tot_time, N0)
```

**Arguments**

- `fit.bd`: an object of class `fit.bd`, output of the `fit_bd` function
- `tot_time`: the age of the underlying phylogeny (crown age, or stem age if known). If working with crown ages, `tot_time` is given by `max(node.age(phylo)$ages)`.
- `N0`: number of extant species. If all extant species are represented in the phylogeny, `N0` is given by `length(phylo$tip.label)`

**Value**

Plot representing how the estimated number of species vary through time

**Author(s)**

H Morlon

**References**


**See Also**

`fit_bd`

**Examples**

```r
data(Balaenopteridae)
tot_time<max(node.age(Balaenopteridae)$ages)

# Fit the pure birth model (no extinction) with exponential variation of the speciation rate
# with time
f.lamb <-function(t,y){y[1] * exp(y[2] * t)}
f.mus<-function(t,y){0}
lamb_par<-c(0.08, 0.01)
```
mu_par <- c()
result <- fit_bd(Balaenopteridae, tot_time, f.lamb, f.mu, lamb_par, mu_par, f=1,
                expo.lamb = TRUE, fix.mu=TRUE)

# plot estimated number of species through time
# plot_dtt(result, tot_time, N0=9)

plot_fit_bd

---

**Description**

Plot estimated speciation, extinction & net diversification rates through time

**Usage**

```r
plot_fit_bd(fit_bd, tot_time)
```

**Arguments**

- `fit_bd`: an object of class 'fit.bd', output of the 'fit_bd' function
- `tot_time`: the age of the phylogeny (crown age, or stem age if known). If working with crown ages, `tot_time` is given by max(node.age(phylo)$ages).

**Value**

Plots representing how the estimated speciation, extinction & net diversification rate functions vary through time

**Author(s)**

H Morlon

**See Also**

- `fit_bd`

**Examples**

```r
data(Balaenopteridae)
tot_time <- max(node.age(Balaenopteridae)$ages)

# Fit the pure birth model (no extinction) with exponential variation of the speciation rate
# with time
f.lamb <- function(t, y){y[1] * exp(y[2] * t)}
f.mu <- function(t, y){0}
lamb_par <- c(0.08, 0.01)
```
mu_par <- c()
result <- fit_bd(Balaenopteridae, tot_time, f.lamb, f.mu, lamb_par, mu_par, 
                 expo.lamb = TRUE, fix.mu = TRUE)

# plot fitted rates
# plot_fit_bd(result, tot_time)

plot_fit_env

Plot specification, extinction & net diversification rate functions of a fitted environmental model

Description
Plot estimated speciation, extinction & net diversification rates as a function of the environmental data and time

Usage
plot_fit_env(fit.env, env_data, tot_time)

Arguments
- fit.env: an object of class 'fit.env', output of the 'fit_env' function
- env_data: environmental data, given as a data frame with two columns. The first column is time, the second column is the environmental data (temperature for instance).
- tot_time: the age of the phylogeny (crown age, or stem age if known). If working with crown ages, tot_time is given by max(node.age(phylo)$ages).

Value
Plots representing how the estimated speciation, extinction & net diversification rate functions vary as a function of the environmental data & time

Author(s)
H Morlon and FL Condamine

See Also
fit_env

Examples

data(Balaenopteridae)
tot_time <- max(node.age(Balaenopteridae)$ages)
data(InTemp)
dof <- smooth.spline(InTemp[, 1], InTemp[, 2])$df
# Fit the pure birth model (no extinction) with exponential variation of the speciation rate
# with time
f.lamb <- function(t,x,y){y[1] * exp(y[2] * x)}
f.mu<- function(t,x,y){0}
lamb_par<-c(0.10, 0.01)
mu_par<-c()
#result <- fit_env(Balaenopteridae,InfTemp,tot_time,f.lamb,f.mu,
# lamb_par,mu_par,f=1, fix.mu=TRUE, df=dof, dt=1e-3)

# plot fitted rates
#plot_fit_env(result, InfTemp, tot_time)

plot_spectR

Spectral density plot of a phylogeny.

Description
Plot the spectral density of a phylogeny and all eigenvalues ranked in descending order.

Usage
plot_spectR(spectR)

Arguments
spectR an object of class 'spectR', output of the 'spectR' function

Value
A 2-panel plot with the spectral density profile on the first panel and the eigenvalues ranked in descending order on the second panel

Author(s)
E Lewitus

References
Lewitus, E., Morlon, H., Characterizing and comparing phylogenies from their Laplacian spectrum, bioRxiv doi: http://dx.doi.org/10.1101/026476

See Also
spectR

Examples
data(Cetacea)
result <- spectR(Cetacea)
#plot_spectR(result)
**Description**

Generates a positive definite and symmetric matrix with specified eigen-values

**Usage**

```r
Posdef(p, ev = rexp(p, 1/100))
```

**Arguments**

- `p`  
The dimension of the matrix
- `ev`  
The eigenvalues. If not specified, eigenvalues are taken from an exponential distribution.

**Details**

`Posdef` generates random positive definite covariance matrices with specified eigen-values that can be used to simulate multivariate datasets (see Uyeda et al. 2015 - and supplied R codes).

**Value**

Returns a symmetric positive-definite matrix with eigen-values = ev.

**Author(s)**

J. Clavel

**References**


**See Also**

`gic.fit_pl`, `rpanda`, `fit_t_pl`, `phyl.pca_pl`
Examples

```r
set.seed(123)

n <- 32 # number of species
p <- 40 # number of traits

tree <- pbtree(n=n) # phylogenetic tree
R <- Posdef(p) # a random symmetric matrix (covariance)
# simulate a dataset
Y <- mvSIM(tree, model="BM1", nsim=1, param=list(sigma=R))

test <- fit_t_pl(Y, tree, model="BM", method="RidgeAlt")
GIC(test)
```

Description

Radiolaria diversity since the Jurassic

Usage

```r
data(sealevel)
```

Details

Radiolaria fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the two following variables:

- `age` a numeric vector corresponding to the geological age, in Myrs before the present
- `radiolaria` a numeric vector corresponding to the estimated ostracod change at that age

References


Examples

```r
data(radiolaria)
plot(radiolaria)
```
redalgae  

*Red algae diversity since the Jurassic*

**Description**

Red algae fossil diversity since the Jurassic

**Usage**

`data(redalgae)`

**Details**

Red algae fossil diversity since the Jurassic compiled from the Neptune Database (Lazarus, 1994) and Paleobiology Database (https://paleobiodb.org/). Diversity curves are estimated at the genus level using shareholder quorum subsampling (Alroy, 2010) at two-million-year bins. The format is a dataframe with the two following variables:

- `age` a numeric vector corresponding to the geological age, in Myrs before the present
- `redalgae` a numeric vector corresponding to the estimated Red algae change at that age

**References**


**Examples**

`data(redalgae)`

`plot(redalgae)`

---

sealevel  

*Sea level data since the Jurassic*

**Description**

Global sea level change since the Jurassic

**Usage**

`data(sealevel)`
Details

Eustatic sea level change since the Jurassic calculated by Miller et al. (2005) from satellite measurements, tide gauges, shoreline markers, reefs, atolls, oxygen isotopes, the flooding history of continental margins, cratons. The format is a dataframe with the two following variables:

- age  a numeric vector corresponding to the geological age, in Myrs before the present
- sea level  a numeric vector corresponding to the estimated sea level change at that age

References


Examples

data(sealevel)
plot(sealevel)

_________________________________________________________

silica  Silica data across the Cenozoic

Description

Silica weathering ratio across the Cenozoic

Usage

data(silica)

Details

Silica weathering ratio across the Cenozoic calculated by Cermeno et al. (2015) using the lithium isotope record of seawater from Misra and Froelich (2012). The format is a dataframe with the two following variables:

- age  a numeric vector corresponding to the geological age, in Myrs before the present
- silica weathering ratio  a numeric vector corresponding to the estimated CO2 at that age

References


Examples

```r
data(silica)
plot(silica)
```

---

`sim.convergence.geo`  
*Simulation of trait data under the model of convergent character displacement described in Drury et al. 2017*

**Description**

Simulates the evolution of a continuous character that evolves depending on pairwise similarity in another, OU-evolving trait (e.g., a trait that covaries with resource use). sig2 and z0 are shared between two traits, max and alpha are for focal trait, OU parameters for non-focal trait.

**Usage**

```r
sim.convergence.geo(phylo, pars, Nsegments=2500, plot=FALSE, geo.object)
```

**Arguments**

- `phylo`: an object of type 'phylo' (see ape documentation)
- `pars`: A matrix with a number of rows corresponding to the desired number of simulations, columns containing values for `sig2` in [,1], `m` in [,2], `alpha` in [,3], `root.value` in [,4], `psi` of the OU model for the non-focal, resource use trait in [,5], and `theta` in the OU model for the non-focal resource use trait in [,6]
- `Nsegments`: the minimum number of time steps to simulate
- `plot`: if TRUE, returns two plots: the top plot is focal trait undergoing convergence, the bottom plot is non-focal trait evolving under BM or OU
- `geo.object`: geography object created using CreateGeoObject

**Details**

Adjusting `Nsegments` will impact the length of time the simulations take. The length of each segment (`max(nodeHeights(phylo))/Nsegments`) should be much smaller than the smallest branch (`min(phylo$edge.length)`).

**Value**

A list of two matrices with the simulated values for each lineage (one simulation per row; columns correspond to species) for trait1 (focal trait undergoing convergence) and non.focal (resource-use trait that determines strength of convergence in trait1)

**Author(s)**

J.P. Drury jonathan.p.drury@gmail.com
References


See Also

CreateGeoObject

Examples

data(Anolis.data)
phylo<-Anolis.data$phylo
geo.object<-Anolis.data$geography.object

#simulate with the OU process present and absent
pars<-expand.grid(0.05,-0.1,1,0,c(2,0),0)
sim.convergence.geo(phylo,pars,Nsegments=2500, plot=FALSE, geo.object)

Description

Simulates the evolution of a continuous character under a model of evolution where trait values are repelled according to between-species similarity in trait values, taking into account biogeography using a biogeo.object formatted from RPANDA (see CreateGeoObject function in RPANDA package)

Usage

sim.divergence.geo(phylo,pars, Nsegments=2500, plot=FALSE, geo.object)

Arguments

phylo a phylogenetic tree
pars A matrix with a number of rows corresponding to the desired number of simulations, columns containing values for sig2 in [,1], m in [,2], alpha in [,3], root.value in [,4], psi of the OU model in [,5], and theta in the OU model in [,6]
Nsegments the minimum number of time steps to simulate
plot logical indicating whether to plot the simulated trait values at each time step
geo.object geography object created using CreateGeoObject
Details

Adjusting \( n_{\text{segments}} \) will impact the length of time the simulations take. The length of each segment (\( \max(\text{nodeHeights}(\text{phylo})) / n_{\text{segments}} \)) should be much smaller than the smallest branch (\( \min(\text{phylo}[, , \text{edge.length}]) \)).

Value

A matrix with the simulated values for each lineage (one simulation per row; columns correspond to species).

Author(s)

J.P. Drury jonathan.p.drury@gmail.com F. Hartig

References


See Also

CreateGeoObject

Examples

```r
data(Anolis.data)
phylo<-Anolis.data$phylo
geo.object<-Anolis.data$geography.object

#simulate with the OU process present and absent
pars<-expand.grid(0.05,2,1,0,c(2,0),0)
sim.divergence.geo(phylo,pars,Nsegments=2500, plot=FALSE, geo.object)
```

simulateTipData

\( \text{Tip trait simulation under a model of phenotypic evolution.} \)

Description

Simulates tip trait data under a specified model of phenotypic evolution, with three distinct behaviours specified with the 'method' argument.

Usage

```
simulateTipData(object, params, method, v)
```
Arguments

object an object of class 'PhenotypicModel'.
params vector of parameters, given in the same order as in the 'model' object.
method an integer specifying the behaviour of the function. If method = 1 (default value), the tip distribution is first computed, before returning a simulated dataset drawn in this distribution. If method = 2, the whole trajectory is simulated step by step, plotted, and returned. Otherwise, the whole trajectory is simulated step by step, and then returned without being plotted.
v boolean specifying the verbose mode. Default value: FALSE.

Value

a vector of trait values at the tips of the tree.

Author(s)

M Manceau

References


Examples

#Loading an example tree
tree <- read.tree(text=newick)

#Creating the models
modelBM <- createModel(tree, 'BM')
modelOU <- createModel(tree, 'OU')

#Simulating tip traits under both models with distinct behaviours of the functions:
dataBM <- simulateTipData(modelBM, c(0,0,0,1))
dataOU <- simulateTipData(modelOU, c(0,0,1,5,1), method=1)
dataBM2 <- simulateTipData(modelBM, c(0,0,0,1), method=2)
Methods

signature(object = "PhenotypicModel") This is the only method available for this function. Same behaviour for any PhenotypicModel.

Description

Simulates a birth-death tree (starting with one lineage) with speciation and/or extinction rate that varies as a function of an input environmental curve. Notations follow Morlon et al. PNAS 2011 and Condamine et al. ELE 2013.

Usage

sim_env_bd(env_data, f.lamb, f.mu, lamb_par, mu_par, df=NULL, time.stop=0, return.all.extinct=TRUE, prune.extinct=TRUE)

Arguments

env_data environmental data, given as a data frame with two columns. The first column is time, the second column is the environmental data (temperature for instance).

time.stop the age of the phylogeny.

f.lamb a function specifying the hypothesized functional form of the variation of the speciation rate $\lambda$ with time and the environmental variable. Any functional form may be used. This function has three arguments: the first argument is time; the second argument is the environmental variable; the third argument is a numeric vector of the parameters controlling the time and environmental variation (to be estimated).

f.mu a function specifying the hypothesized functional form of the variation of the extinction rate $\mu$ with time and the environmental variable. Any functional form may be used. This function has three arguments: the first argument is time; the second argument is the environmental variable; the third argument is a numeric vector of the parameters controlling the time and environmental variation (to be estimated).

lamb_par a numeric vector of initial values for the parameters of f.lamb to be estimated (these values are used by the optimization algorithm). The length of this vector is used to compute the total number of parameters in the model, so to fit a model with constant speciation rate (for example), lamb_par should be a vector of length 1. Otherwise aic values will be wrong.

mu_par a numeric vector of initial values for the parameters of f.mu to be estimated (these values are used by the optimization algorithm). The length of this vector is used to compute the total number of parameters in the model, so to fit a model without extinction (for example), mu_par should be empty (vector of length 0). Otherwise aic values will be wrong.
the degree of freedom to use to define the spline. As a default, smooth.spline(env_data[,1],
env_data[,2])$df is used. See sm.spline for details.

return.all.extinct
return all extinction lineages in simulated tree.

prune.extinct
prune extinct lineages in simulated tree.

Details
In the f.lamb and f.mu functions, time runs from the present to the past.

Value
a list with the following components

tree the simulated tree with number tips
times the times of speciation events starting from the past
nblineages the labels of surviving lineages and total number of surviving lineages

Note
The speed of convergence of the fit might depend on the degree of freedom chosen to define the
spline.

Author(s)
E Lewitus and H Morlon

References
Morlon, H., Parsons, T.L. and Plotkin, J.B. (2011) Reconciling molecular phylogenies with the
fossil record Proc Nat Acad Sci 108: 16327-16332
mental change, Eco Lett 16: 72-85

See Also
fit_env, fit_bd

Examples

data(InfTemp)
dof<-smooth.spline(InfTemp[,1], InfTemp[,2])$df
# Simulates a tree with lambda varying as an exponential function of temperature
# and mu fixed to 0 (no extinction). Here t stands for time and x for temperature.
f.lamb <-function(t,x){y[1] * exp(y[2] * x)}
f.mu<-function(t,x){0}
lamb_par<-c(0.10, 0.01)
mu_par<-c()
#result_exp <- sim_env_bd(InfTemp,f.lamb,f.mu,lamb_par,mu_par,time.stop=10)
Algorithm for simulating a phylogenetic tree under the SGD model

Description

Simulates a phylogeny arising from the SGD model with exponentially increasing metapopulation size. Notations follow Manceau et al. (2015).

Usage

```r
sim_sgd(tau, b, d, nu)
```

Arguments

- `tau` the simulation time, which corresponds to the length of the phylogeny
- `b` the (constant) per-individual birth rate
- `d` the (constant) per-individual death rate
- `nu` the (constant) per-individual mutation rate

Value

A phylogenetic tree of class "phylo" (see ape documentation)

Author(s)

M Manceau

References


Examples

```r
tau <- 10
b <- 1e6
d <- b-0.5
nu <- 0.6
tree <- sim_sgd(tau,b,d,nu)
plot(tree)
```
Recursive simulation (root-to-tip) of competition models

Description

Simulates datasets for a given phylogeny under matching competition (MC), diversity dependent linear (DDlin), or diversity dependent exponential (DDexp) models of trait evolution. Simulations are carried out from the root to the tip of the tree.

Usage

```r
sim_t_comp(phylo, pars, root.value, Nsegments=1000, model="MC, DDexp, DDlin")
```

Arguments

- `phylo`: an object of type 'phylo' (see ape documentation)
- `pars`: a vector containing the two parameters for the chosen model; all models require `sig2`, and additionally, the MC model requires `S`, specifying the level of competition (larger negative values correspond to higher levels of competition), the DDlin model requires `b` and DDexp require `r`, the slope parameters (negative in cases of decline in evolutionary rates with increasing diversity). `sig2` must be listed first.
- `root.value`: a number specifying the trait value for the ancestor
- `Nsegments`: a value specifying the total number of time segments to simulate across for the phylogeny (see Details)
- `model`: model chosen to fit trait data, "MC" is the matching competition model of Nismer & Harmon 2014, "DDlin" is the diversity-dependent linear model, and "DDexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.

Details

Adjusting `Nsegments` will impact the length of time the simulations take. The length of each segment (max(nodeHeights(phylo))/Nsegments) should be much smaller than the smallest branch (min(phylo$edge.length)).

Value

a named vector with simulated trait values for `n` species in the phylogeny

Author(s)

J Drury jonathan.p.drury@gmail.com
References


See Also

fit_t_comp

Examples

data(Cetacea)

# Simulate data under the matching competition model
MC.data<-sim_t_comp(Cetacea,pars=c(sig2=0.01,S=-0.1),root.value=0,Nsegments=1000,model="MC")

# Simulate data under the diversity dependent linear model
DDlin.data<-sim_t_comp(Cetacea,pars=c(sig2=0.01,b=-0.0001),root.value=0,Nsegments=1000,
model="DDlin")

# Simulate data under the diversity dependent linear model
DDexp.data<-sim_t_comp(Cetacea,pars=c(sig2=0.01,r=-0.01),root.value=0,Nsegments=1000,model="DDexp")

sim_t_env

Recursive simulation (root-to-tip) of the environmental model

Description

Simulates datasets for a given phylogeny under the environmental model (see ?fit_t_env)

Usage

sim_t_env(phylo, param, env_data, model, root.value=0, step=0.001, plot=FALSE, ...)
sim_t_env

Arguments

- **phylo**: An object of class `phylo` (see ape documentation)
- **param**: A numeric vector of parameters for the user-defined climatic model. For the `EnvExp` and `EnvLin`, there is only two parameters. The first is sigma and the second beta.
- **env_data**: Environmental data, given as a time continuous function (see, e.g. `splinefun`) or a data frame with two columns. The first column is time, the second column is the environmental data (temperature for instance).
- **model**: The model describing the functional form of variation of the evolutionary rate $\sigma^2$ with time and the environmental variable. Default models are "EnvExp" and "EnvLin" (see details). An user defined function of any functional form can be used (forward in time). This function has three arguments: the first argument is time; the second argument is the environmental variable; the third argument is a numeric vector of the parameters controlling the time and environmental variation (to be estimated). See the example below.
- **root.value**: A number specifying the trait value for the ancestor
- **step**: This argument describe the length of the segments to simulate across for the phylogeny. The smaller is the segment, the greater is the accuracy of the simulation at the expense of the computation time.
- **plot**: If TRUE, the simulated process is plotted.
- **...**: Arguments to be passed through. For instance, "col" for plot=TRUE.

Details

The users defined function is simulated forward in time i.e.: from the root to the tips. The speed of the simulations might depend on the value used for the "step" argument. It's possible to estimate the traits with the MLE from another fitted object (see the example below).

Value

A named vector with simulated trait values for $n$ species in the phylogeny

Author(s)

J. Clavel

References


See Also

`plot.fit_t.env`, `likelihood_t_env`
Examples

```r
data(Cetacea)
data(InfTemp)

set.seed(123)
# define the parameters
param <- c(0.1, -0.5)
# define the environmental function
my_fun <- function(t, env, param){ param[1]*exp(param[2]*env(t))}

# simulate the trait
trait <- sim_t_env(Cetacea, param=param, env_data=InfTemp, model=my_fun, root.value=0, step=0.001, plot=TRUE)

# fit the model to the simulated trait.
fit <- fit_t_env(Cetacea, trait, env_data=InfTemp, model=my_fun, param=c(0.1,0))

# Then use the results from the previous fit to simulate a new dataset
trait2 <- sim_t_env(Cetacea, param=fit, step=0.001, plot=TRUE)
fit2 <- fit_t_env(Cetacea, trait2, env_data=InfTemp, model=my_fun, param=c(0.1,0))

# When providing the environmental function:
require(pspline)
spline_result <- sm.spline(x=InfTemp[,1], y=InfTemp[,2], df=50)
env_func <- function(t){predict(spline_result,t)}
t<-unique(InfTemp[,1])

# We build the interpolated smoothing spline function
env_data<-splinefun(t,env_func(t))

# provide the environmental function to simulate the traits
trait3 <- sim_t_env(Cetacea, param=param, env_data=env_data, model=my_fun, root.value=0, step=0.001, plot=TRUE)
fit3 <- fit_t_env(Cetacea, trait3, env_data=InfTemp, model=my_fun, param=c(0.1,0))
fit3
```

**spectR**

Spectral density plot of a phylogeny

**Description**

Computes the spectra of eigenvalues for the modified graph Laplacian of a phylogenetic tree, identifies the spectral gap, then convolves the eigenvalues with a Gaussian kernel, and plots them alongside all eigenvalues ranked in descending order.
Usage

spectR(phylo, method=c("standard"))

Arguments

phylo an object of type 'phylo' (see ape documentation)
method the method used to compute the spectral density, which can either be "standard" or "normal". If set to "standard", computes the unnormalized version of the spectral density. If set to "normal", computes the spectral density normalized to the degree matrix (see the associated paper for an explanation)

Details

Note that the eigengap should in principle be computed with the "standard" option

Value

a list with the following components:
eigenvalues the vector of eigenvalues
principal_eigenvalue the largest (or principal) eigenvalue of the spectral density profile
asymmetry the skewness of the spectral density profile
peak_height the largest y-axis value of the spectral density profile
eigengap the position of the largest difference between eigenvalues, giving the number of modalities in the tree

Author(s)

E Lewitus

References

Lewitus, E., Morlon, H., Characterizing and comparing phylogenies from their Laplacian spectrum, bioRxiv doi: http://dx.doi.org/10.1101/026476

See Also

plot_spectR, JSDtree, BICompare

Examples

data(Cetacea)
spectR(Cetacea, method="standard")
Index

*Topic methods
  fitTipData-methods, 24
  getDataLikelihood-methods, 49
  getTipDistribution-methods, 50
  modelSelection-methods, 75
  simulateTipData-methods, 103

[, PhenotypicModel, ANY, ANY, ANY-method
  (PhenotypicModel-class), 82
[<-, PhenotypicModel, ANY, ANY, ANY-method
  (PhenotypicModel-class), 82
  ancestral, 5, 47, 87
  Anolis.data, 6
  Balaenopteridae, 7
  BGB.examples, 8
  BICcompare, 9, 56, 91, 111
  Calomys, 10
  Cetacea, 10
  co2, 11
  co2.res, 12
  coccolithophore, 13
  d13c, 22
  foraminifera, 47
  greenalgea, 54
  InfTemp, 55
  landplant, 58
  ostracoda, 76
  Phocoenidae, 86
  Phyllostomidae, 88
  Phyllostomidae.genera, 89
  radiolaria, 97
  redalgea, 98
  sealevel, 98
  silica, 99

*Topic classes
  PhenotypicACDC-class, 76
  PhenotypicADic-class, 78
  PhenotypicBM-class, 79
  PhenotypicDD-class, 80
  PhenotypicGMM-class, 81
  PhenotypicModel-class, 82
  PhenotypicOU-class, 83
  PhenotypicPM-class, 85

*Topic datasets
  Balaenopteridae, 7
  Calomys, 10
  Cetacea, 10
  co2, 11
  co2.res, 12
  coccolithophore, 13
  CreateClassObject, 13, 16
  CreateGeoByClassObject, 13, 14, 15, 40, 64
  CreateGeoObject, 7, 17, 19, 36, 38, 64, 65, 67, 68, 72, 73, 100–102
  CreateGeoObject_BioGeoBEARS, 8, 16, 18
  createModel, 20
  createModelCoevolution, 21
  d13c, 22
  fit_bd, 25, 34, 92, 93, 105
  fit_coal_cst, 28, 32
  fit_coal_var, 29, 30
  fit_env, 32, 94, 105
  fit_sgd, 35
  fit_t_comp, 17–19, 36, 65, 67, 68, 72, 73, 108
  fit_t_comp_subgroup, 14–16, 38
  fit_t_env, 6, 41, 70
  fit_t_pl, 6, 45, 52, 54, 87, 96
  fitTipData, 23
INDEX

fitTipData, PhenotypicModel-method (fitTipData-methods), 24
fitTipData-methods, 24
foraminifera, 47

getDatLikelihood, 48
getDatLikelihood, PhenotypicModel-method (getDatLikelihood-methods), 49
getDatLikelihood-methods, 49
getTipDistribution, 49
getTipDistribution, PhenotypicACDC-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicADiag-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicBM-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicDD-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicGMM-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicModel-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicOU-method (getTipDistribution-methods), 50
getTipDistribution, PhenotypicPM-method (getTipDistribution-methods), 50
getTipDistribution-methods, 50
GIC, 6
GIC.fit_pl.rpanda, 47, 51, 54, 87, 96
gic_criterion, 6, 47, 52, 52, 87
greenalgae, 54
InfTemp, 55

JSData, 56, 58, 111
JSData_cluster, 56, 57

landplant, 58
likelihood_bd, 27, 34, 59
likelihood_coal_cst, 29, 60
likelihood_coal_var, 32, 62

likelihood_sg, 36, 63
likelihood_subgroup_model, 40, 64
likelihood_t_DD, 38, 65, 66, 68
likelihood_t_DD_geog, 38, 67, 67
likelihood_t_env, 43, 69, 74, 90, 109
likelihood_t_MC, 38, 71, 73
likelihood_t_MC_geog, 38, 72, 72
lines.fit_t_env, 73, 90

make.simmap, 14, 15, 17, 39
modelSelection, 75
modelSelection, PhenotypicModel-method (modelSelection-methods), 75
modelSelection-methods, 75
mvgls, 47, 52

optim, 37, 39
ophacoda, 76

par, 74, 90
PhenotypicACDC-class, 76
PhenotypicADiag-class, 78
PhenotypicBM-class, 79
PhenotypicDD-class, 80
PhenotypicGMM-class, 81
PhenotypicModel, 77-80, 82, 84, 85
PhenotypicModel-class, 82
PhenotypicOU-class, 83
PhenotypicPM-class, 85
Phoconidae, 86
phyl.pca.pl, 6, 47, 86, 96
Phylllostomidae, 88, 89
Phylllostomidae_genera, 88, 89
plot.fit_t_env, 43, 74, 89, 109
plot_BICompare, 9, 91
plot_dtt, 27, 92
plot_fit_bd, 27, 93
plot_fit_env, 34, 94
plot_specR, 95, 111
Posdef, 96
print, PhenotypicModel-method (PhenotypicModel-class), 82

radiolaria, 97
redalgae, 98
round, 14
RPANDA (RPANDA-package), 4
RPANDA-package, 4

sealevel, 98
show, PhenotypicModel-method
   (PhenotypicModel-class), 82
silica, 99
sim.convergence.geo, 100
sim.divergence.geo, 101
sim_env_bd, 104
sim_sgd, 106
sim_t_comp, 38, 107
sim_t_env, 108
simulateTipData, 102
simulateTipData, PhenotypicModel-method
   (simulateTipData-methods), 103
simulateTipData-methods, 103
spectR, 9, 56, 95, 110