

Package: RPANDA (via r-universe)

September 8, 2024

Version 2.3

Date 2024-01-22

Type Package

Title Phylogenetic ANALyses of DiversificAtion

Depends R (>= 3.5.0), ape, picante, methods,

Suggests testthat

Imports BB, bipartite, cluster, coda, deSolve, fields, fpc, geiger, glassoFast, graphics, grDevices, igraph, Matrix, mvtnorm, mvMORPH (>= 1.1.6), GUniFrac, parallel, ParallelLogger, phangorn, phytools, pracma, pspline, pvclust, raster, RColorBrewer, R.utils, Rmpfr, stats, TESS, utils, vegan

Encoding UTF-8

Author H el ene Morlon [aut, cre, cph], Eric Lewitus [aut, cph], Fabien Condamine [aut, cph], Marc Manceau [aut, cph], Julien Clavel [aut, cph], Jonathan Drury [aut, cph], Olivier Billaud [aut, cph], Odile Maliet [aut, cph], Leandro Aristide [aut, cph], Benoit Perez-Lamarque [aut, cph], Nathan Mazet [aut, cph]

Maintainer H el ene Morlon <helene.morlon@bio.ens.psl.eu>

Description Implements macroevolutionary analyses on phylogenetic trees. See Morlon et al. (2010)

<DOI:10.1371/journal.pbio.1000493>, Morlon et al. (2011)

<DOI:10.1073/pnas.1102543108>, Condamine et al. (2013)

<DOI:10.1111/ele.12062>, Morlon et al. (2014)

<DOI:10.1111/ele.12251>, Manceau et al. (2015)

<DOI:10.1111/ele.12415>, Lewitus & Morlon (2016)

<DOI:10.1093/sysbio/syv116>, Drury et al. (2016)

<DOI:10.1093/sysbio/syw020>, Manceau et al. (2016)

<DOI:10.1093/sysbio/syw115>, Morlon et al. (2016)

<DOI:10.1111/2041-210X.12526>, Clavel & Morlon (2017)

<DOI:10.1073/pnas.1606868114>, Drury et al. (2017)

<DOI:10.1093/sysbio/syx079>, Lewitus & Morlon (2017)

<DOI:10.1093/sysbio/syx095>, Drury et al. (2018)

<DOI:10.1371/journal.pbio.2003563>, Clavel et al. (2019)

<DOI:10.1093/sysbio/syy045>, Maliet et al. (2019)
 <DOI:10.1038/s41559-019-0908-0>, Billaud et al. (2019)
 <DOI:10.1093/sysbio/syz057>, Lewitus et al. (2019)
 <DOI:10.1093/sysbio/syz061>, Aristide & Morlon (2019)
 <DOI:10.1111/ele.13385>, Maliet et al. (2020)
 <DOI:10.1111/ele.13592>, Drury et al. (2021)
 <DOI:10.1371/journal.pbio.3001270>, Perez-Lamarque & Morlon
 (2022) <DOI:10.1111/mec.16478>, Perez-Lamarque et al. (2022)
 <DOI:10.1101/2021.08.30.458192>, Mazet et al. (2023)
 <DOI:10.1111/2041-210X.14195>, Drury et al. (2024)
 <DOI:10.1016/j.cub.2023.12.055>.

License GPL-2

URL <https://github.com/hmorlon/PANDA>

RoxygenNote 7.2.3

Repository <https://phylotastic.r-universe.dev>

RemoteUrl <https://github.com/hmorlon/PANDA>

RemoteRef HEAD

RemoteSha 5d02681797f7cb4063578c5421590bef6fa48c7a

Contents

RPANDA-package	5
add.gts	7
ancestral	9
Anolis.data	10
apply_prob_dtt	11
Balaenopteridae	13
BGB.examples	14
BICcompare	15
build_network.BipartiteEvol	16
Calomys	17
Caprimulgidae	18
Caprimulgidae_ClaDS2	19
Cetacea	20
Cetacea_clades	21
ClaDS0_example	22
co2	23
co2_res	23
coccolithophore	24
CreateClassObject	25
CreateGeobyClassObject	26
CreateGeoObject	28
CreateGeoObject_BioGeoBEARS	30
createModel	32
createModelCoevolution	33

d13c	34
define_species.BipartiteEvol	35
delineate_phylotypes	37
div.models	39
div.rates	40
fitTipData	42
fitTipData-methods	43
fit_bd	43
fit_bd_backbone	47
fit_bd_backbone_c	50
fit_bd_in_past	54
fit_ClaDS	57
fit_ClaDS0	60
fit_coal_cst	62
fit_coal_var	63
fit_env	66
fit_env_in_past	69
fit_sgd	72
fit_t_comp	74
fit_t_comp_subgroup	77
fit_t_env	80
fit_t_env_ou	84
fit_t_pl	89
fit_t_standard	91
foraminifera	93
get.comb.shift	94
get.sampling.fractions	95
getDataLikelihood	97
getDataLikelihood-methods	98
getMAPS_ClaDS	98
getMAPS_ClaDS0	99
getTipDistribution	101
getTipDistribution-methods	102
GIC.fit_pl.rpanda	103
gic_criterion	104
greenalgae	106
InfTemp	107
JSDtree	107
JSDtree_cluster	108
JSDt_cluster	109
landplant	111
likelihood_bd	111
likelihood_bd_backbone	113
likelihood_coal_cst	115
likelihood_coal_var	116
likelihood_sgd	118
likelihood_subgroup_model	119
likelihood_t_DD	121

likelihood_t_DD_geog	122
likelihood_t_env	124
likelihood_t_MC	126
likelihood_t_MC_geog	127
lines.fit_t.env	128
lines.fit_t.env.ou	130
make_gen.BipartiteEvol	132
mantel_test	134
mantel_test_nbpartners	135
modelSelection	137
modelSelection-methods	138
MPhiFFT	138
mycorrhizal_network	139
ostracoda	140
paleodiv	140
PhenotypicACDC-class	142
PhenotypicADiag-class	143
PhenotypicBM-class	144
PhenotypicDD-class	145
PhenotypicGMM-class	146
PhenotypicModel-class	148
PhenotypicOU-class	149
PhenotypicPM-class	150
Phocoenidae	151
phyl.pca_pl	152
Phyllostomidae	153
Phyllostomidae_genera	154
phylosignal_network	155
phylosignal_sub_network	158
pi_estimator	160
plot.fit_t.env	161
plot.fit_t.env.ou	163
plot_BICcompare	165
plot_ClaDS0_chains	166
plot_ClaDS_chains	167
plot_ClaDS_phylo	168
plot_div.BipartiteEvol	169
plot_dtt	171
plot_fit_bd	172
plot_fit_env	173
plot_net.BipartiteEvol	174
plot_phylosignal_sub_network	176
plot_phylo_comb	177
plot_prob_dtt	180
plot_spectR	182
Posdef	183
prob_dtt	184
radiolaria	186

redalgae	187
remove.model	188
sealevel	189
shift.estimate	190
shifts_cetacea	192
silica	193
sim.BipartiteEvol	194
sim.convergence.geo	196
sim.divergence.geo	198
simul.comb.shift	199
simulateTipData	200
simulateTipData-methods	202
sim_ClaDS	202
sim_env_bd	206
sim_MCBD	208
sim_sgd	210
sim_t_comp	211
sim_t_env	213
sim_t_env_ou	215
sim_t_tworegime	217
spectR	219
spectR_t	220
taxo_cetacea	221
theta_estimator	222

Index**224**

RPANDA-package

*RPANDA***Description**

Implements macroevolutionary analyses on phylogenetic trees

Details

More information on the RPANDA package and worked examples can be found in [Morlon et al. \(2016\)](#)

Author(s)

Hélène Morlon <helene.morlon@bio.ens.psl.eu>

Julien Clavel <julien.clavel@univ-lyon1.fr>

Fabien Condamine <fabien.condamine@gmail.com>

Jonathan Drury <jonathan.p.drury@durham.ac.uk>

Eric Lewitus <elewitus@hivresearch.org>

Marc Manceau <marc.manceau@gmail.com>
Olivier Billaud <olivier.billaud@agroparistech.fr>
Odile Maliet <maliet@biologie.ens.fr>
Leandro Aristide <aristide@biologie.ens.fr>
Benoît Perez-Lamarque <benoit.perez@ens.psl.eu>

References

- Morlon, H., Potts, M.D., Plotkin, J.B. (2010) Inferring the dynamics of diversification: a coalescent approach, *PLoS B* 8(9): e1000493
- Morlon, H., Parsons, T.L. and Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record, *Proc Nat Acad Sci* 108: 16327-16332
- Morlon, H., Kemps, B., Plotkin, J.B., Brisson, D. (2012) Explosive radiation of a bacterial species group, *Evolution* 66: 2577-2586
- Condamine, F.L., Rolland, J., and Morlon, H. (2013) Macroevolutionary perspectives to environmental change, *Eco Lett* 16: 72-85
- Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett* 7: 508-525
- Manceau, M., Lambert, A., Morlon, H. (2015) Phylogenies support out-of-equilibrium models of biodiversity, *Eco Lett* 18: 347-356
- Lewitus, E., Morlon, H. (2016) Characterizing and comparing phylogenies from their Laplacian spectrum, *Syst Biol* 65: 495-507
- Morlon, H., Lewitus, E., Condamine, F.L., Manceau, M., Clavel, J., Drury, J. (2016) RPANDA: an R package for macroevolutionary analyses on phylogenetic trees, *MEE* 7: 589-597
- Drury, J., Clavel, J., Manceau, M., Morlon, H. (2016) Estimating the Effect of Competition on Trait Evolution Using Maximum Likelihood Inference, *Syst Biol* 65: 700-710
- Manceau, M., Lambert, A., Morlon, H. (2017) A Unifying Comparative Phylogenetic Framework Including Traits Coevolving Across Interacting Lineages, *Syst Biol* 66: 551-568
- Clavel, J., Morlon, H. (2017) Accelerated body size evolution during cold climatic periods in the Cenozoic, *Proc Nat Acad Sci* 114: 4183-4188
- Drury, J., Tobias, J., Burns, K., Mason, N., Shultz, A., and Morlon, H. (2018) Contrasting impacts of competition on ecological and social trait evolution in songbirds. *PLOS Biolog* 16: e2003563
- Clavel, J., Aristide, L., Morlon, H. (2019). A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst Biol* 68: 93-116
- Maliet, O., Hartig, F., Morlon, H. (2019). A model with many small shifts for estimating species-specific diversification rates. *Nature Ecol Evol* 3: 1086-1092
- Condamine, F.L., Rolland, J., Morlon, H. (2019) Assessing the causes of diversification slowdowns: temperature-dependent and diversity-dependent models receive equivalent support *Ecology Letters* 22: 1900-1912
- Aristide, L., Morlon, H. (2019) Understanding the effect of competition during evolutionary radiations: an integrated model of phenotypic and species diversification *Ecology Letters* 22: 2006-2017

Billaud, O., Moen, D. S., Parsons, T. L., Morlon, H. (2019) Estimating Diversity Through Time using Molecular Phylogenies: Old and Species-Poor Frog Families are the Remnants of a Diverse Past *Systematic Biology* 69: 363–383

Lewitus, E., Aristide, L., Morlon, H. (2019) Characterizing and Comparing Phylogenetic Trait Data from Their Normalized Laplacian Spectrum *Systematic Biology* 69: 234–248

Maliet, O., Loeuille, N., Morlon, H. (2020) An individual-based model for the eco-evolutionary emergence of bipartite interaction networks *Ecology Letters*

Perez-Lamarque, B., Öpik, M., Maliet, O., Afonso Silva, A.C., Selosse, M-A., Martos, F., Morlon, H. (2022), Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont, *Molecular Ecology*, 31:3496–512.

Perez-Lamarque, B., Maliet, O., Pichon, B., Selosse, M-A., Martos, F., Morlon, H. (2022) Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. *bioRxiv*, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology.

add.gts

Geological time scale

Description

Adds geological time scale (GTS) to plots.

Usage

```
add.gts(thickness, quaternary = T, is.phylo = F,
        xpd.x = T, time.interval = 1, names = NULL, fill = T,
        cex = 1, padj = -0.5, direction = "rightwards")
```

Arguments

thickness	numeric < 0. Define the thickness of the scale.
quaternary	boolean. Whether to merge Pleistocene and Holocene into Quaternary. Default is TRUE.
is.phylo	boolean. Whether the plot is a phylogeny or not. Default is FALSE.
time.interval	numeric. Define the minimum time interval (in million years) for the geological time scale. Default is 1 and displays ticks every million year but with numbers at every five million years.
xpd.x	boolean. Whether to expand the last period of the geological time scale before root age (mainly for tree). Default is TRUE.
names	a character vector with the names of geological periods (stages). Can be used to write abbreviations. Default is NULL and display full names (except for Quaternary and Pliocene).
fill	boolean. If TRUE (default), background is alternatively filled with grey and white bands to distinguish geological periods. If FALSE, dashed lines are drawn to limit geological periods.

cex	numeric. Size of the names of geological periods.
padj	padj argument defining space between the axis and the values of the axis (see par() for more details).
direction	character. Direct the geological time scale. Can be either "rightwards" (default) of "leftwards" (NOT IMPLEMENTED YET).

Details

This function plots a geological times scale (GTS). It has been designed for adding GTS to plot of phylogeny, diversification rates and paleodiversity dynamics through time but can be used with any R plot. Time should be negative for other plots than phylogenies.

Value

Draws geological time scale on x axis.

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

Examples

```
## Not run:
# with a phylogeny
data("Cetacea")
# first plot to get the dimensions of the gts
plot(Cetacea, cex = 0.5, label.offset = 0.2, tip.color = "white")
add.gts(-3, quaternary = T, is.phylo = T, xpd.x = F,
        names = c("Q.", "Pli.", "Miocene", "Oligocene", "Eoc.))
# second plot to display the tree on the gts
par(new = T)
plot(Cetacea, cex = 0.5, label.offset = 0.2)
mtext("Time (Myrs)", side = 1, line = 3, at = 18)

# see Appendix S4 from Mazet et al. (2023) for more examples.

## End(Not run)
```

ancestral	<i>Estimation of traits ancestral states.</i>
-----------	---

Description

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

Usage

```
ancestral(object, ...)
```

Arguments

object	A model fit object obtained by the <code>fit_t_XX</code> class of functions.
...	Further arguments to be passed through (not used yet).

Details

`ancestral` reconstructs the ancestral states at the root and possibly for each nodes 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 nodes (not yet implemented for all the methods)

Note

The function is used internally in `phyl.pca_pl` (Clavel et al. 2019).

Author(s)

J. Clavel

References

- Clavel, J., Aristide, L., Morlon, H., 2019. A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst. Biol.* 68: 93-116.
- Cunningham C.W., Omland K.E., Oakley T.H. 1998. Reconstructing ancestral character states: a critical reappraisal. *Trends Ecol. Evol.* 13:361-366.
- Martins E.P., Hansen T.F. 1997. Phylogenies and the comparative method: a general approach to incorporating phylogenetic information into the analysis of interspecific data. *Am. Nat.* 149:646-667.

See Also

[fit_t_pl](#), [fit_t_env](#), [phyl.pca_pl](#), [GIC](#), [gic_criterion](#)

Examples

```
if(require(mvMORPH)){
  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)

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

- Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020
- Mahler, D.L., Ingram, T., Revell, L., and Losos, J. 2013. Exceptional convergence on the macroevolutionary landscape in island lizard radiations. *Science*. 341:292-295.
- Mahler, D.L. and Ingram, T. 2014. Phylogenetic comparative methods for studying clade-wide convergence. In *Modern Phylogenetic Comparative Methods and Their Application in Evolutionary Biology*, ed. L. Garamszegi. pp.425-450.

See Also

[CreateGeoObject](#)

Examples

```
data(Anolis.data)
plot(Anolis.data$phylo)
print(Anolis.data$data)
print(Anolis.data$geography.object)
```

apply_prob_dtt

Calculates paleodiversity dynamics with the probabilistic approach.

Description

Applies prob_dtt() to outputs from shift.estimate().

Usage

```
apply_prob_dtt(phylo, data, sampling.fractions, shift.res,
               combi = 1, backbone.option = "crown.shift",
               m = NULL)
```

Arguments

phylo an object of type 'phylo' (see ape documentation)

data a data.frame containing a database of monophyletic groups for which potential shifts can be investigated. This database should be based on taxonomy, ecology or traits and contain a column named "Species" with species names as in phylo.

sampling.fractions	the output resulting from get.sampling.fractions.
shift.res	the output resulting from shift.estimate.
backbone.option	type of the backbone analysis: <ul style="list-style-type: none"> • "stem.shift": for every shift, the probability of the speciation event at the stem age of the subclade is included in the likelihood of the backbone thanks to the argument spec_times. • "crown.shift": for every shift, both the probability of the speciation event at the stem age of the subclade and the probability that the stem of the subclade survives to the crown age are included in the likelihood of the backbone thanks to the argument branch_times.
combi	numeric. The combination of shifts defined by its rank in the global comparison.
m	NULL or numeric. The set of maximum values for m ranges. Should be as long as the number of parts in the combination. Default is NULL (see details).

Details

This function calls the function prob_dtt() to calculate paleodiversity dynamics with the probabilistic approach for the different parts of a combination of diversification shifts.

As explained in Billaud et al. (2020), all the sum of probabilities per million year must be equal to 1. However, it can be difficult to reach 1 for groups showing a paleodiversity decline because the range of paleodiversity over which we need to calculate the probabilities can be very large. To circumvent this issue, apply_prob_dtt() set the range of the paleodiversity to the maximum of the deterministic estimate from the function paleodiv() and successively multiplies this maximum by 2, 3, 5, 7 and 10 until the sums of probabilities for each million year reach a minimum of 95%. In few cases, this value of 95% is not reached for few million years. In this case, it might come from an extremely high range of m and maximum values can be manually set up with the argument m.

Value

A list of results from prob_dtt() for subclades and backbone(s).

Author(s)

Nathan Mazet

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.
- Billaud, O., Moen, D.S., Parsons, T.L., Morlon, H., (2020). Estimating Diversity Through Time Using Molecular Phylogenies: Old and Species-Poor Frog Families are the Remnants of a Diverse Past. *Systematic Biology* 69, 363–383.
- Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[fit_bd](#), [plot_prob_dtt](#), [prob_dtt](#)

Examples

```
# loading data
data("Cetacea")
data("taxo_cetacea")
data("shifts_cetacea")

taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]

# apply_prob_dtt() needs the sampling fractions
f_df_cetacea <- get.sampling.fractions(phylo = Cetacea,
                                     data = taxo_cetacea_no_genus,
                                     plot = TRUE, cex = 0.3, lad = FALSE)

# use of apply_prob_dtt()
prob_dtt_cetacea <- apply_prob_dtt(phylo = Cetacea,
                                   data = taxo_cetacea_no_genus,
                                   shift.res = shifts_cetacea,
                                   sampling.fractions = f_df_cetacea,
                                   combi = 1)
```

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

- Steeman, M.E., et al. (2009) Radiation of extant cetaceans driven by restructuring of the oceans *Syst Biol* 58:573-585
- Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

Examples

```
data(Balaenopteridae)
print(Balaenopteridae)
plot(Balaenopteridae)
```

BGB.examples

BioGeoBEARS stochastic maps

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

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Matzke, N. 2014. Model selection in historical biogeography reveals that founder-event speciation is a crucial process in island clades. *Systematic Biology* 63:951-970.

See Also

[CreateGeoObject_BioGeoBEARS](#)

Examples

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

BICcompare *Identify modalities in a phylogeny*

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

```
BICcompare(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_BICcompare](#), [spectR](#), [JSDtree](#)

Examples

```
data(Cetacea)
#BICcompare(Cetacea,5)
```

`build_network.BipartiteEvol`*Build the interaction network in BipartiteEvol*

Description

Build the phylogenies from the output of BipartiteEvol and the corresponding genealogies and phylogenies

Usage

```
build_network.BipartiteEvol( gen, spec)
```

Arguments

gen	The output of a run of make_gen.BipartiteEvol
spec	The output of a run of define_species.BipartiteEvol

Value

A matrix M where M[i,j] is the number of individuals from species i (from guild P) interacting with an individual from species j (from guild H)

Author(s)

O. Maliet

References

Maliet, O., Loeuille, N. and Morlon, H. (2020), An individual-based model for the eco-evolutionary emergence of bipartite interaction networks. Ecol Lett. doi:10.1111/ele.13592

See Also

[sim.BipartiteEvol](#)

Examples

```
# run the model
set.seed(1)

if(test){

mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 800,
                       D = 3, muP = 0.1 , muH = 0.1,
                       alphaP = 0.12,alphaH = 0.12,
                       rP = 10, rH = 10,
```



```

        verbose = 100, thin = 5)

#build the genealogies
gen = make_gen.BipartiteEvol(mod)
plot(gen$H)

#compute the phylogenies
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#plot the result
plot_div.BipartiteEvol(gen,phy1, 1)

#build the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = nx, spatial = FALSE)

## add time steps to a former run
seed=as.integer(10)
set.seed(seed)

mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 200,
                      D = 3, muP = 0.1 , muH = 0.1,
                      alphaP = 0.12,alphaH = 0.12,
                      rP = 10, rH = 10,
                      verbose = 100, thin = 5,
                      P=mod$P,H=mod$H) # former run output

# update the genealogy
gen = make_gen.BipartiteEvol(mod,
                             treeP=gen$P, treeH=gen$H)

# update the phylogenies...
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#... and the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = 10, spatial = FALSE)

}

```

Calomys

Calomys phylogeny

Description

Ultrametric phylogenetic tree of 11 of the 13 extant *Calomys* species

Usage

```
data(Calomys)
```

Details

This phylogeny is from Pigot et al. PloS Biol 2012

References

Pigot et al.(2012) Speciation and extinction drive the appearance of directional range size evolution in phylogenies and the fossil record *PloS Biol* 10:1-9

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

Examples

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

Caprimulgidae

The Caprimulgidae phylogeny.

Description

The MCC phylogeny for the *Caprimulgidae*, from Jetz et al. (2012).

Usage

```
data("Caprimulgidae")
```

Source

Jetz, W., G. Thomas, J. Joy, K. Hartmann, and A. Mooers. 2012. The global diversity of birds in space and time. *Nature* 491:444.

Examples

```
data("Caprimulgidae")
plot(Caprimulgidae)
```

Caprimulgidae_ClaDS2 *An example run of ClaDS2.*

Description

An example of the run on the inference of ClaDS2 on the *Caprimulgidae* phylogeny, thinned every 10 iterations.

Usage

```
data("Caprimulgidae_ClaDS2")
```

Format

A list object with fields :

`tree` The *Caprimulgidae* phylogeny on which we ran the model.

`sample_fraction` The sample fraction for the clade.

`sampler` The chains obtained by running ClaDS2 on the *Caprimulgidae* phylogeny.

Details

The *Caprimulgidae* phylogeny was obtained from Jetz et al. (2012)

Author(s)

O. Maliet

Source

Jetz, W., G. Thomas, J. Joy, K. Hartmann, and A. Mooers. 2012. The global diversity of birds in space and time. *Nature* 491:444.

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS](#), [plot_ClaDS_chains](#), [getMAPS_ClaDS0](#)

Examples

```

data("Caprimulgidae_ClaDS2")

# plot the mcmc chains
plot_ClaDS_chains(Caprimulgidae_ClaDS2$sampler)

# extract the Maxima A Posteriori for each parameter
maps = getMAPS_ClaDS(Caprimulgidae_ClaDS2$sampler, thin = 1)
print(paste0("sigma = ", maps[1], " ; alpha = ",
  maps[2], " ; epsilon = ", maps[3], " ; l_0 = ", maps[4] ))

# plot the inferred branch specific speciation rates
plot_ClaDS_phylo(Caprimulgidae_ClaDS2$tree, maps[-(1:4)])

```

Cetacea

Cetacean phylogeny

Description

Ultrametric phylogenetic tree for 87 of the 89 extant cetacean species

Usage

```
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

Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

Condamine, F.L., Rolland, J., Morlon, H. (2013) Macroevolutionary perspectives to environmental change *Eco Lett* 16: 72-85

Examples

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

Cetacea_clades

Stochastic map of clade membership in Cetacean phylogeny

Description

simmap object of clade membership in Cetacean phylogeny

Usage

```
data(Cetacea_clades)
```

Details

See [Cetacea](#)

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

Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

Condamine, F.L., Rolland, J., Morlon, H. (2013) Macroevolutionary perspectives to environmental change *Eco Lett* 16: 72-85

Examples

```
data(Cetacea_clades)
print(Cetacea_clades)
plot(Cetacea_clades)
```

ClaDS0_example *An example run of ClaDS0.*

Description

An example of the run on the inference of ClaDS0 on a simulated phylogeny, thinned every 10 iterations.

Usage

```
data("ClaDS0_example")
```

Format

A list object with fields :

tree The simulated phylogeny on which we ran the model.

speciation_rates The simulated speciation rates.

Cl0_chains The output of the run_ClaDS0 run.

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS0](#)

Examples

```
data(ClaDS0_example)

# plot the resulting chains for the first 4 parameters
plot_ClaDS0_chains(ClaDS0_example$Cl0_chains, param = 1:4)

# extract the Maximum A Posteriori for each of the parameters
MAPS = getMAPS_ClaDS0(ClaDS0_example$tree,
                     ClaDS0_example$Cl0_chains,
                     thin = 10)

# plot the simulated (on the left) and inferred speciation rates (on the right)
# on the same color scale
plot_ClaDS_phylo(ClaDS0_example$tree,
                 ClaDS0_example$speciation_rates,
                 MAPS[-(1:3)])
```

`co2`*co2 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 eported 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

Mayhew, P.J., Jenkins, G.B., Benton, T.G. (2008) A long-term association between global temperature and biodiversity, origination and extinction in the fossil record *Proceedings of the Royal Society B* 275:47-53

Mayhew, P.J., Bell, M.A., Benton, T.G, McGowan, A.J. (2012) Biodiversity tracks temperature over time *109:15141-15145*

Berner R.A., Kothavala, Z. (2001) GEOCARB III: A revised model of atmospheric CO2 over Phanerozoic time *Am J Sci* 301:182–204

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

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

Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

Condamine, F.L., Rolland, J., Morlon, H. (2013) Macroevolutionary perspectives to environmental change *Eco Lett* 16: 72-85

Examples

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

coccolithophore

Coccolithophore diversity since the Jurassic

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

- Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832
- Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

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(map, rnd=5, return.mat=FALSE)
```

Arguments

map	stochastic map from make.simmap in <code>phytools</code>
rnd	integer indicating the number of decimal places to which times should be rounded (default value is 5) (see round)
return.mat	logical indicating whether to return <code>simmap</code> in a format to be passed to other internal functions (usually FALSE)

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

Drury, J., Tobias, J., Burns, K., Mason, N., Shultz, A., and Morlon, H. in review. Contrasting impacts of competition on ecological and social trait evolution in songbirds. *PLOS Biology*.

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

See Also

[fit_t_comp_subgroup,CreateGeobyClassObject](#)

Examples

```
data(Anolis.data)

#Create a make.simap object
require(phytools)
geo<-c(rep("cuba",7),rep("hispaniola",9),"puerto_rico")
names(geo)<-Anolis.data$phylo$tip.label
stochastic.map<-phytools::make.simap(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,simap,trim.class,ana.events,clado.events,
    stratified=FALSE,rnd=5)
```

Arguments

<code>phylo</code>	the object of type 'phylo' (see ape documentation) used to build ancestral range stochastic maps in BioGeoBEARS
<code>simmap</code>	a phylo object created using <code>make.simmap</code> in phytools
<code>trim.class</code>	category in the simmap object that represents the subgroup of interest (see Details and Examples)
<code>ana.events</code>	the "ana.events" table produced in BioGeoBEARS that lists anagenetic events in the stochastic map
<code>clado.events</code>	the "clado.events" table produced in BioGeoBEARS that lists cladogenetic events in the stochastic map
<code>stratified</code>	logical indicating whether the ancestral biogeography stochastic map was built from a stratified analysis in BioGeoBEARS
<code>rnd</code>	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`.

Value

Returns a list with the following components:

<code>map</code>	a simmap object with phylogeny trimmed to subgroup of interest (including all branches determined to belong to that subgroup)
<code>geography.object</code>	a list with the following components:
<code>geography.matrix</code>	a list of matrices specifying both sympatry & group membership (<code>==1</code>) or allopatry and/or non-membership in the focal subgroup (<code>==0</code>) for each species pair for each internode interval (see Details)
<code>times</code>	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)
<code>spans</code>	a vector specifying the distances between times (used internally in other functions)

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References

Drury, J., Tobias, J., Burns, K., Mason, N., Shultz, A., and Morlon, H. in review. Contrasting impacts of competition on ecological and social trait evolution in songbirds. *PLOS Biology*.

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

See Also

[fit_t_comp_subgroup](#), [CreateGeoObject_BioGeoBEARS](#), [createClassObject](#)

Examples

```
data(BGB.examples)

Canidae.phylo<-BGB.examples$Canidae.phylo
dummy.group<-c(rep("B",3),rep("A",12),rep("B",2),rep("A",6),rep("B",5),rep("A",6))
names(dummy.group)<-Canidae.phylo$tip.label

Canidae.simap<-phytools::make.simap(Canidae.phylo,dummy.group)

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

Canidae.geobyclass.object<-CreateGeobyClassObject(phylo=Canidae.phylo,simap=Canidae.simap,
trim.class="A",ana.events=BGB.examples$Canidae.ana.events,
clado.events=BGB.examples$Canidae.clado.events,stratified=FALSE, rnd=5)

phytools::plotSimmap(Canidae.geobyclass.object$map)
```

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.simap (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:

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

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

References

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

See Also

[fit_t_comp](#)

Examples

```
data(Analis.data)
#Create a geography.object with a modified edge matrix
#First, specify which region each branch belonged to:
Analis.regions<-c(rep("cuba",14),rep("hispaniola",17),"puerto_rico")
Analis.map<-cbind(Analis.data$phylo$edge,Analis.regions)
CreateGeoObject(Analis.data$phylo,map=Analis.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)<-Analis.data$phylo$tip.label
stochastic.map<-phytools::make.simmap(Analis.data$phylo,
  geo, model="ER", nsim=1)
CreateGeoObject(Analis.data$phylo,map=stochastic.map)
```

CreateGeoObject_BioGeoBEARS

Create biogeography object using a stochastic map from BioGeoBEARS

Description

Create biogeography object using a stochastic map created from any model in BioGeoBEARS (see documentation in BioGeoBEARS package).

Usage

```
CreateGeoObject_BioGeoBEARS( full.phylo, trimmed.phylo = NULL, ana.events,
clado.events, stratified=FALSE, simmap.out=FALSE)
```

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
simmap.out	logical indicating whether output should be a stochastic map (simmap) object (see note)

Details

Note: generating a stochastic map output using `simmap.out=TRUE` and passing to `fit_t_comp` for diversity dependent models with biogeography greatly speeds up model fitting compared to output generated when `simmap.out=FALSE`. This cannot be used for matching competition or any two-regime models with biogeography.

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

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Matzke, N. 2014. Model selection in historical biogeography reveals that founder-event speciation is a crucial process in island clades. *Systematic Biology* 63:951-970.

See Also

[fit_t_comp](#) [CreateGeoObject](#)

Examples

```
data(BGB.examples)

##Example with a non-stratified tree

Canidae.geography.object<-CreateGeoObject_BioGeoBEARS(full.phylo=BGB.examples$Canidae.phylo,
ana.events=BGB.examples$Canidae.ana.events, clado.events=BGB.examples$Canidae.clado.events)

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

Canidae.trimmed.geography.object<-CreateGeoObject_BioGeoBEARS(
full.phylo=BGB.examples$Canidae.phylo, trimmed.phylo=Canidae.trimmed,
ana.events=BGB.examples$Canidae.ana.events, clado.events=BGB.examples$Canidae.clado.events)

##Example with a stratified tree

Ochotonidae.geography.object<-CreateGeoObject_BioGeoBEARS(
full.phylo = BGB.examples$Ochotonidae.phylo, ana.events = BGB.examples$Ochotonidae.ana.events,
clado.events = BGB.examples$Ochotonidae.clado.events, stratified = TRUE)

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

Ochotonidae.trimmed.geography.object<-CreateGeoObject_BioGeoBEARS(
```

```
full.phylo=BGB.examples$Ochotonidae.phylo, trimmed.phylo=Ochotonidae.trimmed,
ana.events=BGB.examples$Ochotonidae.ana.events,
clado.events=BGB.examples$Ochotonidae.clado.events, stratified=TRUE)
```

createModel

Creation of a PhenotypicModel

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

```
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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

Examples

```
#Loading an example tree
newick <- "(((A:1,B:0.5):2,(C:3,D:2.5):1):6,E:10.25):2,(F:6.5,G:8.25):3):1;"
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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages Systematic Biology

Examples

```
#Loading example trees
newick1 <- "(((A:1,B:1):3,(C:3,D:3):1):2,E:6);"
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

d13c data since the Jurassic

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

Hannisdal, B., Peters, S.E. (2011) Phanerozoic Earth system evolution and marine biodiversity *Science* 334:1121-1124

Lazarus, D., Barron, J., Renaudie, J., Diver, P., Turke, A. (2014) Cenozoic Planktonic Marine Diatom Diversity and Correlation to Climate Change *PLoS ONE* 9:e84857

Examples

```
data(d13c)
plot(d13c)
```

```
define_species.BipartiteEvol
    Build the phylogenies for BipartiteEvol
```

Description

Build the phylogenies from the output of BipartiteEvol and the corresponding genealogies

Usage

```
define_species.BipartiteEvol(genealogy, threshold = 1,
    distanceH = NULL, distanceP = NULL, verbose = T,
    monophyly = TRUE, seed = NULL)
```

Arguments

genealogy	The output of a run of make_gen.BipartiteEvol
threshold	The species definition ratchet (s)
distanceH	Distance (ie nb of mutations) matrix between the individual of clade H
distanceP	Distance (ie nb of mutations) matrix between the individual of clade P
verbose	Should the progression of the computation be printed?
monophyly	Should the species delineations be strictly monophyletic species (TRUE - default) or not (FALSE)? If not, the threshold must be equal to 1.
seed	If monophyly==FALSE, the seed is used to pick one representative individual per (potentially non-monophyletic) species.

Details

If monophyly==TRUE, species delineation is performed using the model of Speciation by Genetic Differentiation (Manceau et al., 2015) where the ‘threshold’ (the number of mutations needed to belong to different species) can vary. It results in monophyletic species. If monophyly==FALSE, we consider that each new mutation (i.e. each new combination of traits) gives rise to a new species (Perez-Lamarque et al., 2021). As a result, species are not necessarily formed by a monophyletic group of individuals.

Value

a list with

P	The species identity of each individual in guild P
H	The species identity of each individual in guild H
Pphylo	The phylogeny for guild P
Hphylo	The phylogeny for guild H

Author(s)

O. Maliet & B. Perez-Lamarque

References

Manceau, M., A. Lambert, and H. Morlon. (2015). Phylogenies support out-of-equilibrium models of biodiversity. *Ecology letters* 18:347–356.

Maliet, O., Loeuille, N. and Morlon, H. (2020). An individual-based model for the eco-evolutionary emergence of bipartite interaction networks. *Ecol Lett.* doi:10.1111/ele.13592

Perez-Lamarque, B., Maliet, O., Pichon B., Selosse, M-A., Martos, F., Morlon H. (2021). Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. *bioRxiv.* doi: <https://doi.org/10.1101/2021.08.30.458192>

See Also

[sim.BipartiteEvol](#)

Examples

```
# run the model
set.seed(1)

if(test){

mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 800,
                       D = 3, muP = 0.1 , muH = 0.1,
                       alphaP = 0.12,alphaH = 0.12,
                       rP = 10, rH = 10,
                       verbose = 100, thin = 5)

#build the genealogies
gen = make_gen.BipartiteEvol(mod)
plot(gen$H)

#compute the phylogenies
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#plot the result
plot_div.BipartiteEvol(gen,phy1, 1)

#build the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = nx, spatial = FALSE)

## add time steps to a former run
seed=as.integer(10)
set.seed(seed)
```

```

mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 200,
                      D = 3, muP = 0.1 , muH = 0.1,
                      alphaP = 0.12,alphaH = 0.12,
                      rP = 10, rH = 10,
                      verbose = 100, thin = 5,
                      P=mod$P,H=mod$H) # former run output

# update the genealogy
gen = make_gen.BipartiteEvol(mod,
                             treeP=gen$P, treeH=gen$H)

# update the phylogenies...
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#... and the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = 10, spatial = FALSE)
}

```

delineate_phylotypes *Automatic phylotypes delineation*

Description

This function traverses a tree from the root to the tips, at every node computes the average similarity of all sequences descending from the node, and collapses the sequences into a single phylotype if their sequence dissimilarity is lower than a given threshold. The average similarity can be computed using raw measured of the average similarity or using measures of genetic diversity (nucleotidic diversity "pi" (Nei & Li, 1979) or Watterson "theta" (Watterson, 1975)) which correct for gaps in the nucleotidic alignments (Ferretti et al., 2012).

Usage

```
delineate_phylotypes(tree, thresh=97, sequences, method="pi")
```

Arguments

tree	a phylogenetic tree of all the sequences. It must be an object of class "phylo" and must be rooted.
thresh	a numeric digit between 0 and 100 indicating the minimal average similarity to collapse sequences within the same phylotype. By default, the average similarity is 97.
sequences	a matrix representing the nucleotidic alignment of all the sequences present in the phylogenetic tree.

method indicates which method to use to compute the average similarity: "mean" computes the average raw distances between pairs of sequences, "pi" (default) measures the nucleotidic diversity (Nei & Li, 1979) while controlling for gaps in the alignment, and "theta" measures the Watterson theta genetic diversity (Watterson, 1975) also controlling for gaps.

Value

A table with its row names corresponding to the sequence names. The first column corresponds to the phylotype assignation and the second columns indicates the name of the representative sequence of each phylotype (longest sequence available). Phylotypes are numbered starting at 1, and all the phylotypes named "0" correspond to singletons.

Author(s)

Benoît Perez-Lamarque

References

Perez-Lamarque B, Öpik M, Maliet O, Silva A, Selosse M-A, Martos F, and Morlon H. 2022. Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont, *Molecular Ecology*, 31:3496–512.

Ferretti L, Raineri E, Ramos-Onsins S. 2012. Neutrality tests for sequences with missing data. *Genetics* 191: 1397–1401.

Morlon H, O'Connor TK, Bryant JA, Charkoudian LK, Docherty KM, Jones E, Kembel SW, Green JL, Bohannan BJM. 2015. The biogeography of putative microbial antibiotic production. *PLoS ONE* 10.

Nei M & Li WH, Mathematical model for studying genetic variation in terms of restriction endonucleases, 1979, *Proc. Natl. Acad. Sci. USA*.

Watterson GA , On the number of segregating sites in genetical models without recombination, 1975, *Theor. Popul. Biol.*

See Also

[pi_estimator](#) [theta_estimator](#)

Examples

```
library(phytools)

data(woodmouse)

alignment <- as.character(woodmouse) # nucleotidic alignment

tree <- midpoint.root(nj(dist.dna(woodmouse, pairwise.deletion = TRUE,
model = "K80"))) # rooted neighbor-joining tree

# delineate_phylotypes(tree, thresh = 99, alignment, method = "pi")
```

div.models

*Model comparison of diversification models***Description**

Applies a set of birth-death models to a phylogeny.

Usage

```
div.models(phylo, tot_time, f,
           backbone = F, spec_times = NULL, branch_times = NULL,
           models = c("BCST", "BCST_DCST", "BVAR",
                     "BVAR_DCST", "BCST_DVAR", "BVAR_DVAR"),
           cond, verbose = T, n.max = NULL, rate.max = NULL)
```

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 | numeric. The sampling fraction given as the number of species in the phylogeny over the number of species described in the taxonomy. |
| backbone | character. Allows to analyse a backbone. Default is FALSE and spec_times and branch_times are then ignored. Otherwise: <ul style="list-style-type: none"> • "stem.shift": for every shift, the probability of the speciation event at the stem age of the subclade is included in the likelihood of the backbone thanks to the argument spec_times. • "crown.shift": for every shift, both the probability of the speciation event at the stem age of the subclade and the probability that the stem of the subclade survives to the crown age are included in the likelihood of the backbone thanks to the argument branch_times. |
| spec_times | a numeric vector of the stem ages of subclades. Used only if backbone = "stem.shift". Default is NULL. |
| branch_times | a list of numeric vectors. Each vector contain the stem and crown ages of subclades (in this order). Used only if backbone = "crown.shift". Default is NULL. |
| models | a vector of character. Defines the set of birth-death models to applies e.g. BCST means pure-birth constant rate model, BCST_DVAR means birth constant rate and death variable rate model. Default is c("BCST", "BCST_DCST", "BVAR", "BVAR_DCST", "BCST_DVAR", "BVAR_DVAR") and applies all combinations of constant or variable rates for speciation and extinction. Time dependency is only exponential. |
| cond | conditioning to use to fit the model: <ul style="list-style-type: none"> • FALSE: no conditioning (not recommended); |

	<ul style="list-style-type: none"> • "stem": conditioning on the survival of the stem lineage (used 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 on survival of the two daughter lineages (used when the stem age is not known, in this case tot_time should be the crown age).
verbose	boolean. Whether to print model names and AICc values during the calculation.
rate.max	numeric. Set a limit of diversification rates in terms of rate values.
n.max	numeric. Set a limit of diversification rates in terms of diversity estimates with the deterministic approach.

Details

Parameters of birth-death models are defined backward in time such as a positive alpha corresponds to a speciation rate decreasing through time from the past to the present.

Value

A data.frame with number of parameters, likelihood, AICc and parameter values for all models.

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

Examples

```
data("Cetacea")
res <- div.models(Cetacea, tot_time = max(node.age(Cetacea)$ages),
                 f = 87/89, cond = "crown")
```

div.rates

Diversification rates through time

Description

Calculates diversification rates through time from shift.estimate() output.

Usage

```
div.rates(phylo, shift.res, combi = 1, part = "backbone",
          time.interval = 1, backbone.option = "crown.shift")
```


Arguments

phylo	an object of type 'phylo' (see ape documentation)
shift.res	the output resulting from shift.estimate.
combi	numeric. The combination of shifts defined by its rank in the global comparison.
part	character. Specifies for which parts of the combination diversification rates has to be calculated. Default is "backbone" and provides only the backbone rate. Can be "all" for all the parts of a combination or "subclades" for subclades only.
backbone.option	type of the backbone analysis (see backbone.option in shift.estimate for more details): <ul style="list-style-type: none">• "stem.shift": rates are calculated from the stem age for subclades.• "crown.shift": rates are calculated from the crown age for subclades.
time.interval	numeric. Define the time interval (in million years) at which diversification rates are calculated. Default is 1 for a value at each million year.

Value

a list of matrix with two rows (speciation and extinction) and as many columns as million years from the root to the present.

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[shift.estimate](#)

Examples

```
# loading data
data("Cetacea")
data("shifts_cetacea")

# with shifts_cetacea the output from shift.estimate()
rates <- div.rates(phylo = Cetacea, shift.res = shifts_cetacea,
                  combi = 1, part = "all")
```

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, error, params0, GLSstyle, v)
```

Arguments

object	an object of class 'PhenotypicModel'.
data	vector of tip trait data.
error	vector of intraspecific (i.e., tip-level) standard error of the mean. Specify NULL if no error data are available
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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

Examples

```
#Loading an example tree
newick <- "(((A:1,B:0.5):2,(C:3,D:2.5):1):6,E:10.25):2,(F:6.5,G:8.25):3):1;"
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)
```

fitTipData-methods *~~ Methods for Function fitTipData ~~*

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

```
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 <code>max(node.age(phylo)\$ages)</code> .
f.lamb	a function specifying the hypothesized functional form (e.g. constant, linear, exponential, etc.) of the variation of the speciation rate λ 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 μ 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 <i>optim</i> 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.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.
fix.mu	logical: if set to TRUE, the extinction rate μ 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 <code>dt=1e-3</code> 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 λ with time is estimated to be positive (for example), this means that the speciation rate decreases from past to present. Note that abs(f.lamb) and abs(f.mu) are used in the likelihood computation as speciation and extinction rates should always be positive. A consequence of this is that negative speciation/extinction rates estimates can be returned. They should be interpreted in absolute terms. See Morlon et al. 2020 for a more detailed explanation.

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

- Morlon, H., Parsons, T.L. and Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332
- Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett* 17:508-525
- Morlon, H., Rolland, J. and Condamine, F. (2020) Response to Technical Comment 'A cautionary note for users of linear diversification dependencies', *Eco Lett*

See Also

[plot_fit_bd](#), [plot_dtt](#), [likelihood_bd](#), [fit_env](#)

Examples

```

# 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.lamb <-function(t,y){y[1] * exp(y[2] * t)}
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 pure birth model (no extinction) with linear variation of
# the speciation rate with time
f.lamb <-function(t,y){abs(y[1] + y[2] * t)}
# alternative formulation that can be used depending on the choice made to avoid negative rates:
# f.lamb <-function(t,y){pmax(0,y[1] + y[2] * t)}, see Morlon et al. (2020)
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
f.lamb<-function(t,y){y[1] * exp(y[2] * t)}
f.mu <-function(t,y){y[1]}
lamb_par <- c(0.05, 0.01)
mu_par <-c(0.005)
#result_bexp_dcst <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,
#                          f=87/89,expo.lamb=TRUE,cst.mu=TRUE,dt=1e-3)
#result_bexp_dcst$model <- "birth-death with exponential variation in speciation rate
#                          and constant extinction"

# Find the best model
#index <- which.min(c(result_cst$aicc, result_exp$aicc, result_lin$aicc,result_bexp_dcst$aicc))
#rbind(result_cst, result_exp, result_lin, result_bexp_dcst)[index,]

```

fit_bd_backbone	<i>Maximum likelihood fit of the general birth-death model (backbone)</i>
-----------------	---

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. Modified version of fit_bd for backbones.

Usage

```
fit_bd_backbone(phylo, tot_time, f.lamb, f.mu, lamb_par, mu_par, f = 1,
                backbone, spec_times, branch_times,
                meth = "Nelder-Mead", cst.lamb = FALSE, cst.mu = FALSE,
                expo.lamb = FALSE, expo.mu = FALSE, fix.mu = FALSE,
                dt=1e-3, cond = "crown", model)
```

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 <code>max(node.age(phylo)\$ages)</code> .
f.lamb	a function specifying the hypothesized functional form (e.g. constant, linear, exponential, etc.) of the variation of the speciation rate λ 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 μ 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
backbone	character. Allows to analyse a backbone. Default is FALSE and spec_times and branch_times are then ignored. Otherwise

	<ul style="list-style-type: none"> • "stem.shift": for every shift, the probability of the speciation event at the stem age of the subclade is included in the likelihood of the backbone thanks to the argument <code>spec_times</code>. • "crown.shift": for every shift, both the probability of the speciation event at the stem age of the subclade and the probability that the stem of the subclade survives to the crown age are included in the likelihood of the backbone thanks to the argument <code>branch_times</code>.
<code>spec_times</code>	a numeric vector of the stem ages of subclades. Used only if <code>backbone = "stem.shift"</code> . Default is <code>NULL</code> .
<code>branch_times</code>	a list of numeric vectors. Each vector contains the stem and crown ages of subclades (in this order). Used only if <code>backbone = "crown.shift"</code> . Default is <code>NULL</code> .
<code>meth</code>	optimization to use to maximize the likelihood function, see <i>optim</i> for more details.
<code>cst.lamb</code>	logical: should be set to <code>TRUE</code> only if <code>f.lamb</code> is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.
<code>cst.mu</code>	logical: should be set to <code>TRUE</code> only if <code>f.mu</code> is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.
<code>expo.lamb</code>	logical: should be set to <code>TRUE</code> only if <code>f.lamb</code> is exponential to use analytical instead of numerical computation in order to reduce computation time.
<code>expo.mu</code>	logical: should be set to <code>TRUE</code> only if <code>f.mu</code> is exponential to use analytical instead of numerical computation in order to reduce computation time.
<code>fix.mu</code>	logical: if set to <code>TRUE</code> , the extinction rate μ is fixed and will not be optimized.
<code>dt</code>	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 <code>dt</code> is given as argument, integrals are computed using a piece-wise constant approximation, and <code>dt</code> 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 <code>dt=1e-3</code> gives a good trade-off between precision and computation time.
<code>cond</code>	conditioning to use to fit the model: <ul style="list-style-type: none"> • <code>FALSE</code>: no conditioning (not recommended); • "stem": conditioning on the survival of the stem lineage (use when the stem age is known, in this case <code>tot_time</code> should be the stem age); • "crown" (default): conditioning on a speciation event at the crown age and survival of the two daughter lineages (use when the stem age is not known, in this case <code>tot_time</code> should be the crown age).
<code>model</code>	character. The model name as defined in the function <code>div.models</code> .

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 λ with time is estimated to be positive (for example), this means that the speciation rate decreases from past to present. Note that `abs(f.lamb)` and `abs(f.mu)` are used in the likelihood computation as speciation and extinction rates should always be positive. A consequence of this is that negative speciation/extinction rates estimates can be returned. They should be interpreted in absolute terms. See Morlon et al. 2020 for a more detailed explanation.

Value

a list with the following components

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

Author(s)

Hélène Morlon, Nathan Mazet

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

Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett* 17:508-525
 Morlon, H., Rolland, J. and Condamine, F. (2020) Response to Technical Comment 'A cautionary note for users of linear diversification dependencies', *Eco Lett* Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[plot_fit_bd](#), [plot_dtt](#), [likelihood_bd](#), [fit_env](#)

Examples

```
# 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$model <- "pure birth with constant speciation rate"
# Fit the pure birth model (no extinction) with exponential variation
# of the speciation rate with time
f.lamb <-function(t,y){y[1] * exp(y[2] * t)}
f.mu<-function(t,y){0}
lamb_par<-c(0.05, 0.01)
mu_par<-c()
#result_exp <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,
#                    f=87/89,expo.lamb=TRUE,fix.mu=TRUE,dt=1e-3)
#result_exp$model <- "pure birth with exponential variation in speciation rate"
# Fit the pure birth model (no extinction) with linear variation of
# the speciation rate with time
f.lamb <-function(t,y){abs(y[1] + y[2] * t)}
# alternative formulation that can be used depending on the choice made to avoid negative rates:
# f.lamb <-function(t,y){pmax(0,y[1] + y[2] * t)}, see Morlon et al. (2020)
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$model <- "pure birth with linear variation in speciation rate"
# Fit a birth-death model with exponential variation of the speciation
# rate with time and constant extinction
f.lamb<-function(t,y){y[1] * exp(y[2] * t)}
f.mu <-function(t,y){y[1]}
lamb_par <- c(0.05, 0.01)
mu_par <-c(0.005)
#result_bexp_dcst <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,
#                          f=87/89,expo.lamb=TRUE,cst.mu=TRUE,dt=1e-3)
#result_bexp_dcst$model <- "birth-death with exponential variation in speciation rate
#                          and constant extinction"
# Find the best model
#index <- which.min(c(result_cst$aicc, result_exp$aicc, result_lin$aicc,result_bexp_dcst$aicc))
#rbind(result_cst, result_exp, result_lin, result_bexp_dcst)[index,]

```

fit_bd_backbone_c	<i>Maximum likelihood fit of the general birth-death model (backbone and constraints)</i>
-------------------	---

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. Modified version of fit_bd for backbones and to add constraints on rate estimates.

Usage

```
fit_bd_backbone_c(phylo, tot_time, f.lamb, f.mu, lamb_par, mu_par, f = 1,
                 backbone, spec_times, branch_times,
```

```
meth = "Nelder-Mead", cst.lamb = FALSE, cst.mu = FALSE,
expo.lamb = FALSE, expo.mu = FALSE, fix.mu = FALSE,
dt=1e-3, cond = "crown", model, rate.max, n.max)
```

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 <code>max(node.age(phylo)\$ages)</code> .
f.lamb	a function specifying the hypothesized functional form (e.g. constant, linear, exponential, etc.) of the variation of the speciation rate λ 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 μ 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
backbone	character. Allows to analyse a backbone. Default is FALSE and spec_times and branch_times are then ignored. Otherwise <ul style="list-style-type: none"> • "stem.shift": the stems of subclades are included in subclade analyses; • "crown.shift": the stems of subclades are included in the backbone analysis.
spec_times	a numeric vector of the stem ages of subclades. Used only if backbone = "stem.shift". Default is NULL.
branch_times	a list of numeric vectors. Each vector contains the stem and crown ages of subclades (in this order). Used only if backbone = "crown.shift". Default is NULL.
meth	optimization to use to maximize the likelihood function, see <i>optim</i> 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.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.
fix.mu	logical: if set to TRUE, the extinction rate μ 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: <ul style="list-style-type: none"> • 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).
model	character. The model name as defined in the function div.models.
rate.max	numeric. Set a limit of diversification rates in terms of rate values.
n.max	numeric. Set a limit of diversification rates in terms of diversity estimates with the deterministic approach.

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 λ with time is estimated to be positive (for example), this means that the speciation rate decreases from past to present. Note that `abs(f.lamb)` and `abs(f.mu)` are used in the likelihood computation as speciation and extinction rates should always be positive. A consequence of this is that negative speciation/extinction rates estimates can be returned. They should be interpreted in absolute terms. See Morlon et al. 2020 for a more detailed explanation.

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)

Hélène Morlon, Nathan Mazet

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

Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett* 17:508-525
 Morlon, H., Rolland, J. and Condamine, F. (2020) Response to Technical Comment ‘A cautionary note for users of linear diversification dependencies’, *Eco Lett* Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[plot_fit_bd](#), [plot_dtt](#), [likelihood_bd](#), [fit_env](#)

Examples

```
# 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.lamb <-function(t,y){y[1] * exp(y[2] * t)}
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 pure birth model (no extinction) with linear variation of
# the speciation rate with time
f.lamb <-function(t,y){abs(y[1] + y[2] * t)}
# alternative formulation that can be used depending on the choice made to avoid negative rates:
# f.lamb <-function(t,y){pmax(0,y[1] + y[2] * t)}, see Morlon et al. (2020)
```

```
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$model <- "pure birth with linear variation in speciation rate"
# Fit a birth-death model with exponential variation of the speciation
# rate with time and constant extinction
f.lamb<-function(t,y){y[1] * exp(y[2] * t)}
f.mu <-function(t,y){y[1]}
lamb_par <- c(0.05, 0.01)
mu_par <-c(0.005)
#result_bexp_dcst <- fit_bd(Cetacea,tot_time,f.lamb,f.mu,lamb_par,mu_par,
#                           f=87/89,expo.lamb=TRUE,cst.mu=TRUE,dt=1e-3)
#result_bexp_dcst$model <- "birth-death with exponential variation in speciation rate
#                           and constant extinction"
# Find the best model
#index <- which.min(c(result_cst$aicc, result_exp$aicc, result_lin$aicc,result_bexp_dcst$aicc))
#rbind(result_cst, result_exp, result_lin, result_bexp_dcst)[index,]
```

fit_bd_in_past	<i>Maximum likelihood fit of the general birth-death model excluding the recent past</i>
----------------	--

Description

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

Usage

```
fit_bd_in_past(phylo, tot_time, time_stop, f.lamb, f.mu, desc, tot_desc, lamb_par, mu_par,
  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) that does not include any recent speciation (i.e. no speciation events between time_stop and the present).
time_stop	the age of the phylogeny where to stop the birth-death process: it excludes the recent past (between the present and time_stop), while conditioning on the survival of the lineages from time_stop to the present.
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 λ 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 μ 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.
desc	the number of lineages present at present in the reconstructed phylogenetic tree.
tot_desc	the total number of extant species (including in the unsampled ones).
meth	optimization to use to maximize the likelihood function, see <i>optim</i> 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.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.
fix.mu	logical: if set to TRUE, the extinction rate μ 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: <ul style="list-style-type: none"> • 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 λ with time is estimated to be positive (for example), this means that the speciation rate decreases from past to present. Note that abs(f.lamb) and abs(f.mu) are used in the likelihood computation as speciation and extinction rates should always be positive. A consequence of this is that negative speciation/extinction rates estimates can be returned. They should be interpreted in absolute terms. See Morlon et al. 2020 for a more detailed explanation.

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, E Lewitus, B Perez-Lamarque

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
- Lewitus, E., Bittner, L., Malviya, S., Bowler, C., & Morlon, H. (2018) Clade-specific diversification dynamics of marine diatoms since the Jurassic *Nature Ecology and Evolution*, 2(11), 1715–1723
- Perez-Lamarque, B., Öpik, M., Maliet, O., Afonso Silva, A., Selosse, M-A., Martos, F., Morlon, H., Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont, *Molecular Ecology* 31: 3496–3512

See Also

[fit_env_in_past](#), [fit_bd](#), [plot_fit_bd](#), [plot_dtt](#)

Examples

```

library(ape)
library(phytools)

data(Cetacea)

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

# slice the Cetaceae tree 10 Myr ago:
time_stop=10
sliced_tree <- Cetacea
sliced_sub_trees <- treeSlice(sliced_tree,slice = tot_time-time_stop, trivial=TRUE)
for (i in 1:length(sliced_sub_trees)){if (Ntip(sliced_sub_trees[[i]])>1){
  sliced_tree <- drop.tip(sliced_tree,tip=sliced_sub_trees[[i]]$tip.label[2:Ntip(sliced_sub_trees[[i]])])
}}
for (i in which(node.depth.edgelen(sliced_tree)>(tot_time-time_stop))){sliced_tree$edge.length[which(sliced_

Ntip(sliced_tree) # 27 lineages present 10 Myr have survived until today

# Now we can fit birth-death models excluding the 10 last Myr

# 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_in_past(sliced_tree, tot_time, time_stop, f.lamb, f.mu,
  desc=Ntip(Cetacea), tot_desc=89, lamb_par, mu_par,
  cst.lamb = TRUE, fix.mu=TRUE, dt=1e-3)

```

fit_ClaDS

Fit ClaDS to a phylogeny

Description

Performs the inference of branch-specific speciation rates and the model's hyper parameters for the model with constant extinction rate (ClaDS1) or constant turnover rate (ClaDS2).

Usage

```

fit_ClaDS(tree,sample_fraction,iterations, thin = 50, file_name = NULL, it_save = 1000,
  model_id="ClaDS2", nCPU = 1, mcmcSampler = NULL, ...)

```

Arguments

tree	An object of class 'phylo'
sample_fraction	The sampling fraction for the clade on which the inference is performed.
iterations	Number of steps in the MCMC, should be a multiple of thin.
thin	Number of iterations between two chain state's recordings.
file_name	Name of the file in which the result will be saved. Use file_name = NULL (the default) to disable this option.
it_save	Number of iteration between each backup of the result in file_name.
model_id	"ClaDS1" for constant extinction rate, "ClaDS2" (the default) for constant turnover rate.
nCPU	The number of CPUs to use. Should be either 1 or 3.
mcmcSampler	Optional output of <code>fit_ClaDS</code> to continue an already started run.
...	Optional arguments, see details.

Details

This function uses a blocked Differential Evolution (DE) MCMC sampler, with sampling from the past of the chains (Ter Braak, 2006; ter Braak and Vrugt, 2008). This sampler is self-adaptive because proposals are generated from the past of the chains. In this sampler, three chains are run simultaneously. Block updates is implemented by first drawing the number of parameters to be updated from a truncated geometric distribution with mean 3, drawing uniformly which parameter to update, and then following the normal DE algorithm.

The available optional arguments are :

Nchain Number of MCMC chains (default to 3).

res_ClaDS0 The output of ClaDS0 to use as a startpoint. If NULL (the default) a random startpoint is used for the branch-specific speciation rates for each chain.

l0 The starting value for lambda_0 (not used if res_ClaDS0 != NULL).

s0 The starting value for sigma (not used if res_ClaDS0 != NULL).

nlambda Number of subdivisions for the rate space discretization (use in the likelihood computation). Default to 1000.

nt Number of subdivisions for the time space discretization (use in the likelihood computation). Default to 30.

Value

A 'list' object with fields :

post	The posterior function.
startvalue	The starting value for the MCMC.
numPars	The number of parameter in the model, including the branch-specific speciation rates.

Nchain	The number of MCMC chains ran simultaneously.
currentLPs	The current values of the logposterior for th Nchains chains.
proposalGenerator	The proposal distribution for the MCMC sampler.
former	The last output of post for each of the chains.
thin	Number of iterations between two chain state's recordings.
alpha_effect	A vector of size nrow(tree\$edge), where the ith element is the number of branches on the path from the crown of the tree and branch i (used internally in other functions).
consoleupdates	The frequency at which the sampler state should be printed.
likelihood	The likelihood function, used internally.
relToAbs	A function mapping the relative changes in speciation rates to the absolute speciation rates for the object phylo, used internally.

Author(s)

O. Maliet

References

Ter Braak, C. J. 2006. A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces. *Statistics and Computing* 16:239-249.

ter Braak, C. J. and J. A. Vrugt. 2008. Differential evolution markov chain with snooker updater and fewer chains. *Statistics and Computing* 18:435-446.

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS0](#), [plot_ClaDS_chains](#).

Examples

```
if(test){
  data("Caprimulgidae")

  sample_fraction = 0.61

  sampler = fit_ClaDS(Caprimulgidae, sample_fraction, 1000, thin = 50,
                    file_name = NULL, model_id="ClaDS2", nCPU = 1)
  plot_ClaDS_chains(sampler)

  # continue the same run
  sampler = fit_ClaDS(Caprimulgidae, sample_fraction, 50, mcmcSampler = sampler)
```

```

# plot the result of the analysis (saved in "Caprimulgidae_ClaDS2", after thinning)

data("Caprimulgidae_ClaDS2")

# plot the mcmc chains
plot_ClaDS_chains(Caprimulgidae_ClaDS2$sampler)

# extract the Maxima A Posteriori for each parameter
maps = getMAPS_ClaDS(Caprimulgidae_ClaDS2$sampler, thin = 1)
print(paste0("sigma = ", maps[1], " ; alpha = ",
  maps[2], " ; epsilon = ", maps[3], " ; l_0 = ", maps[4] ))

# plot the inferred branch specific speciation rates
plot_ClaDS_phylo(Caprimulgidae_ClaDS2$tree, maps[-(1:4)])
}

```

fit_ClaDS0

Infer ClaDS0's parameter on a phylogeny

Description

Infer branch-specific speciation rates and the model's hyper parameters for the pure-birth model

Usage

```

fit_ClaDS0(tree, name, pamhLocalName = "pamhLocal",
  iteration = 1e+07, thin = 20000, update = 1000,
  adaptation = 10, seed = NULL, nCPU = 3)

```

Arguments

tree	An object of class 'phylo'.
name	The name of the file in which the results will be saved. Use name = NULL to disable this option.
pamhLocalName	The function is writing in a text file to make the execution quicker, this is the name of this file.
iteration	Number of iterations after which the gelman factor is computed and printed. The function stops if it is below 1.05
thin	Number of iterations between two chain state's recordings.
update	Number of iterations between two adjustments of the proposal parameters during the adaptation phase of the sampler.
adaptation	Number of times the proposal is adjusted during the adaptation phase of the sampler.
seed	An optional seed for the MCMC run.
nCPU	The number of CPUs to use. Should be either 1 or 3.

Details

This function uses a Metropolis within Gibbs MCMC sampler with a bactrian proposal (ref) with an initial adaptation phase. During this phase, the proposal is adjusted "adaptation" times every "update" iterations to reach a goal acceptance rate of 0.3.

To monitor convergence, 3 independant MCMC chains are run simultaneously and the Gelman statistics is computed every "iteration" iterations. The inference is stopped when the maximum of the one dimensional Gelman statistics (computed for each of the parameters) is below 1.05.

Value

A mcmc.list object with the three MCMC chains.

Author(s)

O. Maliet

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[getMAPS_ClaDS0](#), [plot_ClaDS0_chains](#), [fit_ClaDS](#)

Examples

```
set.seed(1)

if(test){

obj= sim_ClaDS( lambda_0=0.1,
               mu_0=0.5,
               sigma_lamb=0.7,
               alpha_lamb=0.90,
               condition="taxa",
               taxa_stop = 20,
               prune_extinct = TRUE)

tree = obj$tree
speciation_rates = obj$lamb[obj$rates]
extinction_rates = obj$mu[obj$rates]

plot_ClaDS_phylo(tree,speciation_rates)

sampler = fit_ClaDS0(tree=tree,
                    name="ClaDS0_example.Rdata",
                    nCPU=1,
                    pamhLocalName = "local",
                    iteration=500000,
```

```

        thin=2000,
        update=1000, adaptation=5)

# extract the Maximum A Posteriori for each of the parameters
MAPS = getMAPS_ClaDS0(tree, sampler, thin = 10)

# plot the simulated (on the left) and inferred speciation rates (on the right)
# on the same color scale
plot_ClaDS_phylo(tree, speciation_rates, MAPS[-(1:3)])

}

```

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 (known 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 <i>optim</i> 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
tau0	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

- Hey, J. (1992) Using phylogenetic trees to study speciation and extinction, *Evolution*, 46: 627-640
- Morlon, H., Potts, M.D., Plotkin, J.B. (2010) Inferring the dynamics of diversification: a coalescent approach, *PLoS B*, 8(9): e1000493
- Morlon, H., Kemps, B., Plotkin, J.B., Brisson, D. (2012) Explosive radiation of a bacterial species group, *Evolution*, 66: 2577-2586
- Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett*, 17:508-525

See Also

[likelihood_coal_cst](#), [fit_coal_var](#)

Examples

```
data(Cetacea)

if(test){
  result <- fit_coal_cst(Cetacea, tau0=1.e-3, gamma=-1, cst.rate=FALSE, N0=89)
  print(result)
}
```

fit_coal_var

Fit birth-death model using a coalescent approach

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

```
fit_coal_var(phylo, lamb0 = 0.1, alpha = 1, mu0 = 0.01, beta = 0,
             meth = "Nelder-Mead", N0 = 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 <i>optim</i> 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.
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

<code>model</code>	the name of the fitted model
<code>LH</code>	the maximum log-likelihood value
<code>aicc</code>	the second order Akaike's Information Criterion
<code>model.parameters</code>	the estimated parameters

Author(s)

H Morlon

References

Morlon, H., Potts, M.D., Plotkin, J.B. (2010) Inferring the dynamics of diversification: a coalescent approach, *PLoS B* 8(9): e1000493

Morlon, H., Kemp, B., Plotkin, J.B., Brisson, D. (2012) Explosive radiation of a bacterial species group, *Evolution*, 66: 2577-2586

Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett*, 17:508-525

See Also

[likelihood_coal_var](#), [fit_coal_cst](#)

Examples

```
data(Cetacea)

if(test){
  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 <code>max(node.age(phylo)\$ages)</code> .
f.lamb	a function specifying the hypothesized functional form of the variation of the speciation rate λ 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 μ 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).
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, <code>smooth.spline(env_data[,1], env_data[,2])\$df</code> is used. See <i>sm.spline</i> for details.
f	the fraction of extant species included in the phylogeny
meth	optimization to use to maximize the likelihood function, see <i>optim</i> 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 μ 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: <ul style="list-style-type: none"> • 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. Note that `abs(f.lamb)` and `abs(f.mu)` are used in the likelihood computation as speciation and extinction rates should always be positive. A consequence of this is that negative speciation/extinction rates estimates can be returned. They should be interpreted in absolute terms. See Morlon et al. 2020 for a more detailed explanation.

Value

a list with the following components

fit_env_in_past	<i>Maximum likelihood fit of the environmental birth-death model excluding the recent past</i>
-----------------	--

Description

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

Usage

```
fit_env_in_past(phylo, env_data, tot_time, time_stop, f.lamb, f.mu, desc, tot_desc,
               lamb_par, mu_par, df= NULL, 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) that does not include any recent speciation (i.e. no speciation events between time_stop and the present).
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 where to stop the birth-death process: it excludes the recent past (between the present and time_stop), while conditioning on the survival of the lineages from time_stop to the present.
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 λ 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 μ 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).
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 <i>sm.spline</i> for details.
desc	the number of lineages present at present in the reconstructed phylogenetic tree.
tot_desc	the total number of extant species (including in the unsampled ones).
meth	optimization to use to maximize the likelihood function, see <i>optim</i> 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 μ 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: <ul style="list-style-type: none"> • 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. Note that abs(f.lamb) and abs(f.mu) are used in the likelihood computation as speciation and extinction rates should always be positive. A consequence of this is that negative

speciation/extinction rates estimates can be returned. They should be interpreted in absolute terms. See Morlon et al. 2020 for a more detailed explanation.

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, F Condamine, E Lewitus, B Perez-Lamarque

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
- Condamine, F.L., Rolland, J., and Morlon, H. (2013) Macroevolutionary perspectives to environmental change, *Eco Lett* 16: 72-85
- Lewitus, E., Bittner, L., Malviya, S., Bowler, C., & Morlon, H. (2018) Clade-specific diversification dynamics of marine diatoms since the Jurassic *Nature Ecology and Evolution*, 2(11), 1715–1723
- Perez-Lamarque, B., Öpik, M., Maliet, O., Afonso Silva, A., Selosse, M-A., Martos, F., Morlon, H., Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont, *Molecular Ecology* 31: 3496–3512

See Also

[plot_fit_env](#), [fit_bd_in_past](#), [fit_env](#)

Examples

```
library(ape)
library(phytools)
library(pspline)

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

```

dof<-smooth.spline(InfTemp[,1], InfTemp[,2])$df

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

# slice the Cetaceae tree 5 Myr ago:
time_stop=5
sliced_tree <- Cetacea
sliced_sub_trees <- treeSlice(sliced_tree,slice = tot_time-time_stop, trivial=TRUE)
for (i in 1:length(sliced_sub_trees)){if (Ntip(sliced_sub_trees[[i]])>1){
  sliced_tree <- drop.tip(sliced_tree,tip=sliced_sub_trees[[i]]$tip.label[2:Ntip(sliced_sub_trees[[i]])])
}}
for (i in which(node.depth.edgelen(sliced_tree)>(tot_time-time_stop))){sliced_tree$edge.length[which(sliced_

Ntip(sliced_tree) # 52 lineages present 5 Myr have survived until today

# Now we can fit environment-dependent birth-death models excluding the 5 last Myr

# 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()

#result_env <- fit_env_in_past(sliced_tree, InfTemp, tot_time, time_stop, f.lamb,
#                               f.mu, lamb_par,mu_par,
#                               desc=Ntip(Cetacea), tot_desc=89,
#                               fix.mu=TRUE,df=dof,dt=1e-3)

```

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 <i>optim</i> 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

Manceau, M., Lambert, A., Morlon, H. (2015) Phylogenies support out-of-equilibrium models of biodiversity Ecology Letters 18: 347-356

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(phylo, data, error=NULL, model=c("MC", "DDexp", "DDlin"), pars=NULL,
  geography.object=NULL, regime.map=NULL)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
data	a named vector of trait values with names matching phylo\$tip.label
error	A named vector with standard errors (SE) of trait values for each species (with names matching "phylo\$tip.label"). The default is NULL, in this case potential error is ignored in the fit. If set to NA, the SE is estimated from the data (to be used when there are no error measurements, a nuisance parameter is estimated). Note: When standard errors are provided, a nuisance parameter is also estimated.
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.
pars	vector specifying starting parameter values for maximum likelihood optimization. If unspecified, default values are used (see Details)
geography.object	if incorporating biogeography, a list of sympatry through time created using CreateGeoObject
regime.map	if running two-regime versions of models, a stochastic map of the two regimes stored as a simmap object output from make.simmap

Details

Note: if including known measurement error, the model fit incorporates this known error and, in addition, estimates an unknown, nuisance contribution to measurement error. The current implementation does not differentiate between the two, so, for instance, it is not possible to estimate the nuisance measurement error without providing the known, intraspecific error values.

For single-regime fits without measurement error, par takes the default values of $\text{var}(\text{data})/\max(\text{nodeHeights}(\text{phylo}))$ 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.

For two-regime fits without measurement error, the second and third values for par correspond to the first and second S, b, or r value (run trial fit to see which regime corresponds to each slope).

For fits including measurement error, the default starting value for sig2 is $0.95 \cdot \text{var}(\text{data}) / \max(\text{nodeHeights}(\text{phylo}))$, and nuisance values start at $0.05 \cdot \text{var}(\text{data}) / \max(\text{nodeHeights}(\text{phylo}))$. In all cases, the nuisance parameter is the last in the par vector, with the order of other variables as described above.

For two-regime fits, particularly under the matching competition model, we recommend fitting with several different starting values.

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
nuisance	maximum-likelihood estimate of nuisance, the unknown, nuisance contribution to measurement error (see details)
convergence	convergence diagnostics from <code>optim</code> function (see <code>optim</code> 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

- Drury, J., Clavel, J., Tobias, J., Rolland, J., Sheard, C., and Morlon, H. Tempo and mode of morphological evolution are decoupled from latitude in birds. *PLOS Biology* doi:10.1371/journal.pbio.3001270
- Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* 65:700-710
- Nuismer, S. & Harmon, L. 2015. Predicting rates of interspecific interaction from phylogenetic trees. *Ecology Letters* 18:17-27.
- Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

See Also

[sim_t_comp](#) [CreateGeoObject](#) [likelihood_t_MC](#) [likelihood_t_MC_geog](#) [likelihood_t_DD](#) [likelihood_t_DD_geog](#) [fit_t_comp_subgroup](#)

Examples

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

# 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)

# Now fit models that estimate parameters separately according to different 'regimes'
MC.two_regime.fit<-fit_t_comp(phylo, pPC1, model="MC", regime.map=regime.map)
DDlin.two_regime.fit<-fit_t_comp(phylo, pPC1, model="DDlin", regime.map=regime.map)
DDexp.two_regime.fit<-fit_t_comp(phylo, pPC1, model="DDexp", regime.map=regime.map)

# Now fit models that estimate parameters separately according to different 'regimes',
# including biogeography
MC.two_regime.geo.fit<-fit_t_comp(phylo, pPC1, model="MC",
  geography.object=geography.object, regime.map=regime.map)
DDlin.two_regime.geo.fit<-fit_t_comp(phylo, pPC1, model="DDlin",
  geography.object=geography.object, regime.map=regime.map)
DDexp.two_regime.geo.fit<-fit_t_comp(phylo, pPC1, model="DDexp",
  geography.object=geography.object, regime.map=regime.map)
```

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

```
fit_t_comp_subgroup(full.phylo, data, subgroup, subgroup.map,
  model=c("MC", "DDexp", "DDlin"), ana.events=NULL, clado.events=NULL,
  stratified=FALSE, regime.map=NULL, error=NULL, 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
data	a named vector of trait values for subgroup members with names matching full.phylo\$tip.label
subgroup	subgroup whose members are competing
subgroup.map	a phylo object created using make.simmap in phytools that contains reconstructed subgroup membership
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.
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
regime.map	a phylo object created using make.simmap in phytools that contains reconstructed competitive regime membership (see Details)
error	A named vector with standard error (SE) for each species (with names matching "phylo\$tip.label"). Default is NULL, if NA, then the SE is estimated from the data (a nuisance parameter for unknown errors). Note: When standard error are provided the nuisance parameter is also estimated.
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 ; for DD models without biogeography, method="BB" is also supported, which uses spg)
bounds	(optional) list of bounds to pass to optimization algorithm (see details at optim)

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 `sig` 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. We recommend running ML optimization with several different starting values to ensure convergence.

Currently, this function can be used to implement the following models: 1. Subgroup pruning with biogeography: matching competition, diversity dependent 2. Subgroup pruning without biogeography: diversity dependent 3. Subgroup pruning without biogeography (two-regimes): diversity dependent (for more details, see [fit_t_comp](#))

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 <code>sig2</code> parameter
S	maximum-likelihood estimate of <code>S</code> parameter of matching competition model (see Note)
b	maximum-likelihood estimate of <code>b</code> parameter of linear diversity dependence model (see Note)
r	maximum-likelihood estimate of <code>r</code> parameter of exponential diversity dependence model (see Note)
z0	maximum-likelihood estimate of <code>z0</code> , the value at the root of the tree
convergence	convergence diagnostics from optim function (see optim documentation)
nuisance	maximum-likelihood estimate of nuisance, the unknown, nuisance contribution to measurement error when <code>error</code> argument is used (that is NA or a vector provided by the user)

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

- Drury, J., Clavel, J., Tobias, J., Rolland, J., Sheard, C., and Morlon, H. Tempo and mode of morphological evolution are decoupled from latitude in birds. *PLOS Biology* doi:10.1371/journal.pbio.3001270
- Drury, J., Tobias, J., Burns, K., Mason, N., Shultz, A., and Morlon, H. 2018. Contrasting impacts of competition on ecological and social trait evolution in songbirds. *PLOS Biology* 16(1): e2003563.
- Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* 65: 700-710
- Nuismer, S. & Harmon, L. 2015. Predicting rates of interspecific interaction from phylogenetic trees. *Ecology Letters* 18:17-27.
- Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

See Also

[likelihood_subgroup_model](#) [CreateGeobyClassObject](#) [fit_t_comp](#)

Examples

```
data(BGB.examples)

#Prepare dataset with subgroups and biogeography

Canidae.phylo<-BGB.examples$Canidae.phylo
dummy.group<-c(rep("B",3),rep("A",12),rep("B",2),rep("A",6),rep("B",5),rep("A",6))
names(dummy.group)<-Canidae.phylo$tip.label

Canidae.simap<-phytools::make.simap(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")]

#Fit model with subgroup pruning and biogeography
MC.fit_subgroup_geo<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
  ana.events=BGB.examples$Canidae.ana.events,
  clado.events=BGB.examples$Canidae.clado.events,
  stratified=FALSE,subgroup.map=Canidae.simap,
  data=Canidae.A,subgroup="A",model="MC")

DDexp.fit_subgroup_geo<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
```

```

ana.events=BGB.examples$Canidae.ana.events,
clado.events=BGB.examples$Canidae.clado.events,
stratified=FALSE, subgroup.map=Canidae.simmmap,
data=Canidae.A, subgroup="A", model="DDexp")

DDlin.fit_subgroup_geo<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
ana.events=BGB.examples$Canidae.ana.events,
clado.events=BGB.examples$Canidae.clado.events,
stratified=FALSE, subgroup.map=Canidae.simmmap,
data=Canidae.A, subgroup="A", model="DDlin")

#Fit model with subgroup pruning and no biogeography (for DD models only)
DDexp.fit_subgroup_no_geo<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
data=Canidae.A, subgroup="A", subgroup.map=Canidae.simmmap, model="DDexp")

DDlin.fit_subgroup_no_geo<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
data=Canidae.A, subgroup="A", subgroup.map=Canidae.simmmap, model="DDlin")

#Prepare regime map for fitting two-regime models with subgroup pruning (for DD models only)
regime<-c(rep("regime1",15),rep("regime2",19))
names(regime)<-Canidae.phylo$tip.label
regime.map<-phytools::make.simmmap(Canidae.phylo,regime)

#Fit model with subgroup pruning and two-regimes (for DD models only)
DDexp.fit_subgroup_two.regime<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
data=Canidae.A, subgroup="A", subgroup.map=Canidae.simmmap,
model="DDexp", regime.map=regime.map)

DDlin.fit_subgroup_two.regime<-fit_t_comp_subgroup(full.phylo=Canidae.phylo,
data=Canidae.A, subgroup="A", subgroup.map=Canidae.simmmap,
model="DDlin", regime.map=regime.map)

```

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

```

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 errors (SE) of trait values for each species (with names matching "phylo\$tip.label"). The default is NULL, in this case potential error is ignored in the fit. If set to NA, the SE is estimated from the data (to be used when there are no error measurements, a nuisance parameter is estimated). Note: When standard errors are provided, a nuisance parameter is also estimated.
model	The model describing the functional form of variation of the evolutionary rate σ^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):

```
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:

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

where param[1] is the σ^2 parameter and param[2] is the β parameter. Note that in this later case, two starting values should be provided in the param argument.

e.g.:

```
sigma=0.1
```

```
beta=0
```

```
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)
nuisance	maximum-likelihood estimate of nuisance, the unknown, nuisance contribution to measurement error when error argument is used (i.e., NA or a vector provided by the user)

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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Sciences*, 114(16): 4183-4188.

See Also

[plot.fit_t.env](#), [likelihood_t.env](#)

Examples

```
if(test){
  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

if(require(apspline)){

  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))

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

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)

}

```

Description

Fits Ornstein-Uhlenbeck (OU) model of trait evolution for which the optimum depends on an environmental function, or more generally a time varying function.

Usage

```
fit_t_env_ou(phylo, data, env_data, error=NULL, model,
             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 errors (SE) of trait values for each species (with names matching "phylo\$tip.label"). The default is NULL, in this case potential error is ignored in the fit. If set to NA, the SE is estimated from the data (to be used when there are no error measurements, a nuisance parameter is estimated). Note: When standard errors are provided, a nuisance parameter is also estimated.
model	A user defined model. If not provided, a default model is used (see details)
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_ou allows fitting OU-environmental models of trait evolution (Troyer et al. 2020, Goswami & Clavel 2024). Compared to model implemented in fit_t_env where the rate of phenotypic evolution evolves as a function of an environmental variable (Clavel & Morlon 2020), here it's the optimum of a generalized Ornstein-Uhlenbeck (also called Hull-White model) that can changes as a function of an environmental variable $T(t)$. More formally, the model is defined by the following process:

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

Note that this model works only on NON-ULTRAMETRIC trees (e.g., with fossils)

The default model has the optimum changing as a function of environmental changes though times as defined below:

$$\theta(t) = \theta_0 + \beta T(t)$$

Users defined models should have the following form (see also examples below):

```
fun <- function(t, env, param, theta0){ theta0 + 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.

theta_0: is the state at the root of the tree.

For instance, the default model function can be coded as:

```
fun <- function(t, env, param, theta0){ theta0 + param[1]*env(t)}
```

where `param[1]` is the β parameter. Note that in this case, one starting value should be provided in the `param` argument.

e.g.:

```
beta=0
```

```
fit_t_env(tree, data, env_data=InfTemp, model=fun, param=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.

-`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 custom 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)
nuisance	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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Science*, 114(16): 4183-4188.

Troyer, E., Betancur-R, R., Hughes, L., Westneat, M., Carnevale, G., White W.T., Pogonoski, J.J., Tyler, J.C., Baldwin, C.C., Orti, G., Brinkworth, A., Clavel, J., Arcila, D., 2022 - The impact of paleoclimatic changes on body size evolution in marine fishes. *Proceedings of the National Academy of Sciences*, 119 (29), e2122486119.

Goswami, A. & Clavel, J., 2024. Morphological evolution in a time of Phenomics. *EcoEvoRxiv*, <https://doi.org/10.32942/X22G7Q>

See Also

[plot.fit_t.env.ou.sim_t_env_ou](#)

Examples

```
data(InfTemp)

# Simulate a trait with temperature dependence of the optimum on a simulated tree

set.seed(9999) # for reproducibility

# Let's start by simulating a trait under a climatic OU
beta = 0.6          # relationship to the climate curve
sim_theta = 4       # value of the optimum if the relationship to the climate curve is 0
sim_sigma2 = 0.025 # variance of the scatter = sigma^2
sim_alpha = 0.36    # alpha value = strength of the OU; quite high here...
```

```

delta = 0.001      # time step used for the forward simulations => here its 1000y steps
tree <- phytools::pbtree(n=200, d=0.3) # simulate a bd tree with some extinct lineages
root_age = 60      # height of the root (almost all the Cenozoic here)
tree$edge.length <- root_age*tree$edge.length/max(phytools::nodeHeights(tree))
# here - for this contrived example - I scale the tree so that the root is at 60 Ma

trait <- sim_t_env_ou(tree, sigma=sqrt(sim_sigma2), alpha=sim_alpha, theta0=sim_theta,
                    param=beta, env_data=InfTemp, step=0.01, scale=TRUE, plot=TRUE)

## Fit the Environmental model (default)

result1 <- fit_t_env_ou(phylo = tree, data = trait, env_data =InfTemp,
                      method = "Nelder-Mead", df=50, scale=TRUE)
plot(result1)

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

# We re-define the function for the OU model with linear trend to the climatic curve
# NOTE: the env(t) function should return the value at the root for t=0

my_fun<-function(t, env, param, theta0){
  theta0 + param[1]*env(t)
}

# starting value for param[1]. Here we use an arbitrary value of 0.1
beta_guess = 0.1

# fit the model
result2 <- fit_t_env_ou(phylo = tree, data = trait, env_data =InfTemp,
                      model = my_fun, param = beta_guess,
                      method = "Nelder-Mead", df=50, scale=TRUE)

# Retrieve the parameters and compare to 'result1'
result2
lines(result2, col="red", lty=2)

## Fit user defined environmental function

require(apspline)
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 (not scaled here)
env_data<-splinefun(t,env_func(t))

# We then fit the model

result3 <- fit_t_env_ou(phylo = tree, data = trait, env_data = env_data,
                      model = my_fun, param = 0.01, method = "Nelder-Mead")

```


fit_t_pl

*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 "RidgeAltapprox" 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. (2019)*.

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

References

Clavel, J., Aristide, L., Morlon, H., 2019. A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst. Biol.* 68: 93-116.

See Also

[ancestral](#), [phyl.pca_pl](#), [GIC.fit_pl.rpanda](#), [gic_criterion mvgl](#)

Examples

```

if(test){
  require(mvMORPH)
  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 the model
  fit_t_pl(Y, tree, model="BM", method="RidgeAlt")

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

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

```

fit_t_standard	<i>Fits standard models of trait evolution incorporating known and nuisance measurement error</i>
----------------	---

Description

Fits Brownian motion (BM), Ornstein-Uhlenbeck (OU), or early burst (EB) models of trait evolution to a given dataset and phylogeny.

Usage

```
fit_t_standard(phylo, data, model=c("BM", "OU", "EB"), error=NULL, two.regime=FALSE,
  method="Nelder-Mead", echo=TRUE, ...)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation); if two.regime=TRUE, this must be a simmap object from make.simmap with two regimes
data	a named vector of trait values with names matching phylo\$tip.label
model	model chosen to fit trait data, "BM" is the Brownian motion model, "OU" is the Ornstein-Uhlenbeck model, and "EB" is the early burst model.

error	A named vector with standard errors (SE) of trait values for each species (with names matching "phylo\$tip.label"). The default is NULL, in this case potential error is ignored in the fit. If set to NA, the SE is estimated from the data (to be used when there are no error measurements, a nuisance parameter is estimated). Note: When standard errors are provided, a nuisance parameter is also estimated.
two.regime	if TRUE, fits a two-regime model
method	optimization method from link{optim}
echo	prints information to console during fit
...	Optional arguments. e.g. "upper=xx", "lower=xx" to specify bounds on the parameter search. "fixedRoot=TRUE" to use an OU model where the root state is assumed fixed (instead of sampled from the stationary distribution)

Details

Note: if including known measurement error, the model fit incorporates this known error and, in addition, estimates an unknown, nuisance contribution to measurement error. The current implementation does not differentiate between the two, so, for instance, it is not possible to estimate the nuisance measurement error without providing the known, intraspecific error values.

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
alpha	maximum-likelihood estimate of alpha parameter of OU model (see Note)
r	maximum-likelihood estimate of the slope parameter of early burst model
z0	maximum-likelihood estimate of z0, the value at the root of the tree
nuisance	maximum-likelihood estimate of nuisance, the unknown, nuisance contribution to measurement error (see details)
convergence	convergence diagnostics from <code>optim</code> function (see <code>optim</code> documentation)

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

Julien Clavel

See Also

[fit_t_comp](#) [sim_t_tworegime](#)

Examples

```

if(test){
  data(Cetacea_clades)
  data<-sim_t_tworegime(Cetacea_clades,pars=c(sig2=0.01,r1=-0.01,r2=-0.02),
    root.value=0,Nsegments=1000,model="EB")
  error<-rep(0.05,length(Cetacea_clades$tip.label))
  names(error)<-Cetacea_clades$tip.label

  #Fit single-regime models
  BM1.fit<-fit_t_standard(Cetacea_clades,data,model="BM",error,two.regime=FALSE)
  OU1.fit<-fit_t_standard(Cetacea_clades,data,model="OU",error,two.regime=FALSE)
  EB1.fit<-fit_t_standard(Cetacea_clades,data,model="EB",error,two.regime=FALSE)

  #Now fit models that incorporate biogeography, NOTE these models take longer to fit
  BM2.fit<-fit_t_standard(Cetacea_clades,data,model="BM",error,two.regime=TRUE)
  OU2.fit<-fit_t_standard(Cetacea_clades,data,model="OU",error,two.regime=TRUE)
  EB2.fit<-fit_t_standard(Cetacea_clades,data,model="EB",error,two.regime=TRUE)
}

```

foraminifera

Foraminifera diversity since the Jurassic

Description

Foraminifera fossil diversity since the Jurassic

Usage

```
data(foraminifera)
```

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

- Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832
- Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

Examples

```
data(foraminifera)
plot(foraminifera)
```

get.comb.shift	<i>Combinations of shifts of diversification.</i>
----------------	---

Description

Provides all the combinations of nodes of a phylogeny where shifts of diversification can be tested.

Usage

```
get.comb.shift(phylo, data, sampling.fractions,
               clade.size = 5, Ncores = 1)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
data	a data.frame containing a database of monophyletic groups for which potential shifts can be tested. This database should be based on taxonomy, ecology or traits and must contain a column named "Species" with species names as in phylo.
sampling.fractions	the output resulting from get.sampling.fractions.
clade.size	numeric. Define the minimum number of species in a subgroup. Default is 5.
Ncores	numeric. Define the number of CPU cores to use for parallelizing the computation of combinations.

Details

clade.size argument should be the same value for the whole procedure (same that for get.sampling.fraction and shift.estimate).

Value

a vector of character summarizing the combination of shifts as a concatenation of node IDs separated by "." or "/". Node IDs at the left of "/" correspond to shifts at the origin of subclades (monophyletic and ultrametric subtrees) while node IDs at the right of "/" correspond to shifts at the origin of backbone(s) (pruned trees).

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[get.sampling.fractions](#), [shift.estimate](#)s

Examples

```
# loading data
data("Cetacea")
data("taxo_cetacea")

# no shifts tested at genus level
taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]

f_cetacea <- get.sampling.fractions(phylo = Cetacea,
                                  data = taxo_cetacea_no_genus)

comb.shift_cetacea <- get.comb.shift(phylo = Cetacea,
                                    data = taxo_cetacea_no_genus,
                                    sampling.fractions = f_cetacea,
                                    Ncores = 4)
```

```
get.sampling.fractions
```

Sampling fractions of subclades

Description

Provides the sampling fractions of a phylogenetic tree from a complete database.

Usage

```
get.sampling.fractions(phylo, data, clade.size = 5, plot = F,
                      lad = T, text.cex = 1, pch.cex = 0.8, ...)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
data	a data.frame containing a database of monophyletic groups for which potential shifts can be tested. This database should be based on taxonomy, ecology or traits and must contain a column named "Species" with species names as in phylo.
clade.size	numeric. Define the minimum number of species in a subgroup. Default is 5.

plot	boolean. If TRUE, the tree is plotted and testable nodes are highlighted with red dots. Default is FALSE.
lad	boolean. Define which way the tree should be represented if plot = T. If TRUE, the smallest clade is at the bottom plot. If FALSE, it is at the top of the plot. Default is TRUE.
text.cex	numeric. Defines the size of the text in legend.
pch.cex	numeric. Defines the size of the red points at the crown of subclades.
...	further arguments to be passed to plot or to plot.phylo.

Details

All described species should be included to properly calculate sampling fractions. The example of Cetacea uses a taxonomic database but groups can be defined on geography or traits as soon as they are monophyletic. If the taxonomy of the studied group is difficult to establish (e.i. taxonomic uncertainty, etc.), a "fake" taxonomic database can be created with random species names (Gen1_sp1, Gen1_sp2, Gen2_sp1, etc.) to circumvent taxonomic difficulties. Note that sampling fractions of the backbones are calculated in the next step of the pipeline (function get.comb.shift()).

Value

a data.frame with as many rows as nodes in the phylogeny with the following informations in columns:

nodes	the node IDs
data	the name of the subclade from data
f	the sampling fraction for this subclade
sp_in	the number of species included in the tree
sp_tt	the number of species described in the data
to_test	the node IDs for nodes that are testable according to clade.size

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[get.comb.shift](#), [shift.estimate](#)

Examples

```
# loading data
data("Cetacea")
data("taxo_cetacea")

# no shifts tested at genus level
taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]

# calculating sampling fractions with a plot
f_cetacea <- get.sampling.fractions(phylo = Cetacea, lad = FALSE,
                                   data = taxo_cetacea_no_genus,
                                   plot = TRUE, cex = 0.3)
```

getDataLikelihood *Likelihood of tip trait values.*

Description

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

Usage

```
getDataLikelihood(object, data, error, params, v)
```

Arguments

object	an object of class 'PhenotypicModel'.
data	vector of tip trait data.
error	vector of intraspecific (i.e., tip-level) standard error of the mean. Specify NULL if no error data are available.
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 : $-\log(\text{likelihood})$ of the model.

Author(s)

M Manceau

References

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

Examples

```
#Loading an example tree
newick <- "(((A:1,B:0.5):2,(C:3,D:2.5):1):6,E:10.25):2,(F:6.5,G:8.25):3):1;"
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))

#Likelihood of the data :
getDataLikelihood(modelBM, dataBM, error=NULL, c(0,0,0,1))
```

```
getDataLikelihood-methods
```

```
~~ Methods for Function getDataLikelihood ~~
```

Description

```
~~ Methods for function getDataLikelihood ~~
```

Methods

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

```
getMAPS_ClaDS
```

```
Gets the Maximum A Posteriori for each ClaDS parameter
```

Description

Extract the MAPs (Maximum A Posteriori) for the marginal posterior distributions estimated with fit_ClaDS

Usage

```
getMAPS_ClaDS(sampler, burn = 1/2, thin = 1)
```

Arguments

sampler	The output of a fit_ClaDS run.
burn	Number of iterations to drop in the beginning of the chains.
thin	Thinning parameter, one iteration out of "thin" is kept to compute the MAPs.

Value

A vector MAPS containing the MAPs for the marginal posterior distribution for each of the model's parameters.

MAPS[1:4] are the estimated hyperparameters, with MAPS[1] the sigma parameter (new rates stochasticity), MAPS[2] the alpha parameter (new rates trend), MAPS[3] the turnover rate epsilon, and MAPS[4] the initial speciation rate lambda_0.

MAPS[-(1:4)] are the estimated branch-specific speciation rates, given in the same order as the edges of the phylogeny on which the inference was performed.

Author(s)

O. Maliet

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS](#), [plot_ClaDS_chains](#), [getMAPS_ClaDS0](#)

Examples

```
data("Caprimulgidae_ClaDS2")

if(test){
MAPS = getMAPS_ClaDS(Caprimulgidae_ClaDS2$sampler, thin = 1)

print(paste0("sigma = ", MAPS[1], " ; alpha = ",
  MAPS[2], " ; epsilon = ", MAPS[3], " ; l_0 = ", MAPS[4] ))
plot_ClaDS_phylo(Caprimulgidae_ClaDS2$tree, MAPS[-(1:4)])
}
```

getMAPS_ClaDS0

Gets the Maximum A Posteriori for each ClaDS0 parameter

Description

Extract the MAPs (Maximum A Posteriori) for the marginal posterior distributions estimated with run_ClaDS0.

Usage

```
getMAPS_ClaDS0(phylo, sampler, burn=1/2, thin=1)
```

Arguments

phylo	An object of class 'phylo'.
sampler	The output of a run_ClaDS0 run.
burn	Number of iterations to drop in the beginning of the chains.
thin	Thinning parameter, one iteration out of "thin" is kept to compute the MAPs.

Value

A vector MAPS containing the MAPs for the marginal posterior distribution for each of the model's parameters.

MAPS[1:3] are the estimated hyperparameters, with MAPS[1] the sigma parameter (new rates stochasticity), MAPS[2] the alpha parameter (new rates trend), and MAPS[3] the initial speciation rate lambda_0.

MAPS[-(1:3)] are the estimated branch-specific speciation rates, given in the same order as the phylo\$edges.

Author(s)

O. Maliet

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS0](#), [plot_ClaDS0_chains](#), [getMAPS_ClaDS](#)

Examples

```
set.seed(1)

if(test){
obj= sim_ClaDS( lambda_0=0.1,
               mu_0=0.5,
               sigma_lamb=0.7,
               alpha_lamb=0.90,
               condition="taxa",
               taxa_stop = 20,
               prune_extinct = TRUE)

tree = obj$tree
speciation_rates = obj$lamb[obj$rates]
extinction_rates = obj$mu[obj$rates]
data("ClaDS0_example")

# extract the Maximum A Posteriori for each of the parameters
```

```

MAPS = getMAPS_ClaDS0(ClaDS0_example$tree,
                      ClaDS0_example$Cl0_chains,
                      thin = 10)

# plot the simulated (on the left) and inferred speciation rates (on the right)
# on the same color scale
plot_ClaDS_phylo(ClaDS0_example$tree,
                 ClaDS0_example$speciation_rates,
                 MAPS[-(1:3)])
}

```

getTipDistribution *Distribution of tip trait values.*

Description

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

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

Examples

```
#Loading an example tree
newick <- "(((A:1,B:0.5):2,(C:3,D:2.5):1):6,E:10.25):2,(F:6.5,G:8.25):3):1;"
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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

GIC.fit_pl.rpanda	<i>Generalized Information Criterion (GIC) to compare models fit by Maximum Likelihood (ML) or Penalized Likelihood (PL).</i>
-------------------	---

Description

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

Usage

```
## 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

Konishi S., Kitagawa G. 1996. Generalised information criteria in model selection. *Biometrika*. 83:875-890.

Clavel, J., Aristide, L., Morlon, H., 2019. A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst. Biol.* 68: 93-116.

See Also

[gic_criterion](#), [fit_t_pl](#) [mvgl](#)

Examples

```

if(require(mvMORPH)){

if(test){
  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

```

gic_criterion(Y, tree, model="BM", method=c("RidgeAlt", "RidgeArch", "LASSO", "ML",
"RidgeAltapprox", "LASSOapprox"), targM=c("null",
"Variance", "unitVariance"), param=NULL,
tuning=0, REML=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, "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

- Konishi S., Kitagawa G. 1996. Generalised information criteria in model selection. *Biometrika*. 83:875-890.
- Clavel, J., Aristide, L., Morlon, H., 2019. A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst. Biol.* 68: 93-116.

See Also

[GIC.fit_pl.rpanda](#), [fit_t_pl](#)

Examples

```
if(test){

  if(require(mvMORPH)){
    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

```
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

Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832

Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

Examples

```
data(greenalgae)
plot(greenalgae)
```

InfTemp

Paleotemperature data across the Cenozoic

Description

Paleotemperature data across the Cenozoic inferred from delta O18 measurements

Usage

```
data(InfTemp)
```

Details

Paleotemperature data inferred from delta O18 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

Epstein, S., Buchsbaum, R., Lowenstam, H.A., Urey, H.C. (1953) Revised carbonate-water isotopic temperature scale *Geol. Soc. Am. Bull.* 64: 1315-1326

Zachos, J.C., Dickens, G.R., Zeebe, R.E. (2008) An early Cenozoic perspective on greenhouse warming and carbon-cycle dynamics *Nature* 451: 279-283

Condamine, F.L., Rolland, J., Morlon, H. (2013) Macroevolutionary perspectives to environmental change *Eco Lett* 16: 72-85

Examples

```
data(InfTemp)
plot(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)
```

JSDtree_cluster *Clustering of phylogenies*

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 phylogenie 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>

See Also

[JSDtree](#)

Examples

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

JSDt_cluster

Clustering on the Jensen-Shannon distance between phylogenetic trait data

Description

Computes the Jensen-Shannon distance metric between spectral density profiles of phylogenetic trait data and clusters on those distances.

Usage

```
JSDt_cluster(phylo,mat,plot=F)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
mat	a matrix of trait data with one trait per column and rows aligned to phylo tips
plot	plot hierarchical cluster in a new window

Value

plots a heatmap and hierarchical cluster with bootstrap support (>0.9) and outputs results of the k-medoids clustering on the optimal number of clusters in the form of a list with the following components

clusters	a list with the following components: size, max_diss, av_diss, diameter, and separation
J-S matrix	a matrix providing the Jensen-Shannon distance values between pairs of phylogenetic trait data
cluster assignment	a table that lists for each trait its cluster assignment and silhouette width

Author(s)

E Lewitus

References

Lewitus, E., Morlon, H. (2019) Characterizing and comparing phylogenetic trait data from their normalized Laplacian spectrum, bioRxiv doi: <https://doi.org/10.1101/654087>

See Also

[spectR_t](#)

Examples

```
data(Cetacea)
n<-length(Cetacea$tip.label)
mat<-replicate(20, rnorm(n))
colnames(mat)<-1:dim(mat)[2]
#JSDt_cluster(Cetacea,mat)
```

landplant

Land plant diversity since the Jurassic

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

Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832

Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

Examples

```
data(landplant)
plot(landplant)
```

likelihood_bd

Likelihood of a phylogeny under the general birth-death model

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 <code>max(node.age(phylo)\$ages)</code> .
f.lamb	a function specifying the time-variation of the speciation rate λ . This function as a single argument (time). Any function may be used.
f.mu	a function specifying the time-variation of the speciation rate μ . 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 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 <code>dt=1e-3</code> gives a good trade-off between precision and computation time.
cond	conditioning to use to fit the model: <ul style="list-style-type: none"> • 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

Morlon, H., Parsons, T.L. and Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

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.lamb <- function(t){lamb_par[1] * exp(lamb_par[2] * t)}
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)
```

likelihood_bd_backbone

Likelihood of a phylogeny under the general birth-death model (backbone)

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. Modified version of likelihood_bd for backbones.

Usage

```
likelihood_bd_backbone(phylo, tot_time, f, f.lamb, f.mu,
                      backbone, spec_times, branch_times,
                      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 λ . This function as a single argument (time). Any function may be used.
f.mu	a function specifying the time-variation of the speciation rate μ . This function as a single argument (time). Any function may be used.
f	the fraction of extant species included in the phylogeny
backbone	character. Allows to analyse a backbone. Default is NULL and spec_times and branch_times are then ignored. Otherwise:

- "stem.shift": for every shift, the probability of the speciation event at the stem age of the subclade is included in the likelihood of the backbone thanks to the argument `spec_times`.
- "crown.shift": for every shift, both the probability of the speciation event at the stem age of the subclade and the probability that the stem of the subclade survives to the crown age are included in the likelihood of the backbone thanks to the argument `branch_times`.

<code>spec_times</code>	a numeric vector of the stem ages of subclades. Used only if <code>backbone = "stem.shift"</code> . Default is <code>NULL</code> .
<code>branch_times</code>	a list of numeric vectors. Each vector contains the stem and crown ages of subclades (in this order). Used only if <code>backbone = "crown.shift"</code> . Default is <code>NULL</code> .
<code>cst.lamb</code>	logical: should be set to <code>TRUE</code> only if <code>f.lamb</code> is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.
<code>cst.mu</code>	logical: should be set to <code>TRUE</code> only if <code>f.mu</code> is constant (i.e. does not depend on time) to use analytical instead of numerical computation in order to reduce computation time.
<code>expo.lamb</code>	logical: should be set to <code>TRUE</code> only if <code>f.lamb</code> is exponential to use analytical instead of numerical computation in order to reduce computation time.
<code>expo.mu</code>	logical: should be set to <code>TRUE</code> only if <code>f.mu</code> is exponential to use analytical instead of numerical computation in order to reduce computation time.
<code>dt</code>	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 <code>dt</code> is given as argument, integrals are computed using a piece-wise constant approximation, and <code>dt</code> 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 <code>dt=1e-3</code> gives a good trade-off between precision and computation time.
<code>cond</code>	conditioning to use to fit the model: <ul style="list-style-type: none"> • <code>FALSE</code>: no conditioning (not recommended); • "stem": conditioning on the survival of the stem lineage (use when the stem age is known, in this case <code>tot_time</code> 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 <code>tot_time</code> 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 el ene Morlon, Nathan Mazet

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 Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

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.lamb <- function(t){lamb_par[1] * exp(lamb_par[2] * t)}
f.mu <- function(t){0}
f <- 87/89
# same as likelihood_bd in this case
lh <- likelihood_bd_backbone(Cetacea, tot_time, f, f.lamb, f.mu,
                             backbone = FALSE, spec_times = NULL, branch_times = NULL,
                             cst.mu = TRUE, expo.lamb = TRUE, dt = 1e-3)
```

likelihood_coal_cst *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

Morlon, H., Potts, M.D., Plotkin, J.B. (2010) Inferring the dynamics of diversification: a coalescent approach, PLoS B 8(9): e1000493

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 extant 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

Morlon, H., Potts, M.D., Plotkin, J.B. (2010) Inferring the dynamics of diversification: a coalescent approach, PLoS B 8(9): e1000493

Examples

```
data(Cetacea)
Vtimes <- sort(branching.times(Cetacea))
lamb0 <- 0.1
alpha <- 0.001
mu0<-0
beta<-0
ntips <- Ntip(Cetacea)
N0 <- 89
likelihood <- likelihood_coal_var(Vtimes, ntips, lamb0, alpha, mu0, beta, N0)
```

`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

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

Arguments

<code>phylo</code>	an object of type 'phylo' (see ape documentation)
<code>tot_time</code>	the age of the phylogeny (crown age, or stem age if known). If working with crown ages, <code>tot_time</code> is given by <code>max(node.age(phylo)\$ages)</code> .
<code>b</code>	the (constant) birth rate of individuals in the model.
<code>d</code>	the (constant) death rate of individuals in the model.
<code>nu</code>	the (constant) mutation rate of individuals in the model.
<code>f</code>	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

Manceau M., Lambert A., Morlon H. (2015) Phylogenies support out-of-equilibrium models of biodiversity Ecology Letters 18: 347-356

Examples

```
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 `sigma2` and `slope` values and with a `geography.object` formed using [CreateGeoObject](#).

Usage

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

Arguments

<code>phylo</code>	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
<code>data</code>	a named vector of continuous data for a subgroup of interest with names corresponding to <code>phylo\$tip.label</code>
<code>geography.object</code>	a list of sympatry/group membership through time created using CreateGeobyClassObject
<code>model</code>	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.
<code>par</code>	a vector listing a value for $\log(\text{sig}2)$ (see Note) and either <code>b</code> (for the linear diversity dependent model) or <code>r</code> (for the exponential diversity dependent model), in that order.
<code>return.z0</code>	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)
<code>maxN</code>	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 <code>sig2</code> values at any point in time (see Details).
<code>error</code>	A named vector with standard errors (SE) of trait values for each species (with names matching " <code>phylo\$tip.label</code> "). The default is NULL, in this case potential error is ignored in the fit. If set to NA, the SE is estimated from the data (to be used when there are no error measurements, a nuisance parameter is estimated). Note: When standard errors are provided, a nuisance parameter is also estimated.

Details

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

`maxN` can be calculated using `maxN=max(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 `geography.object`.

If `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 `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

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

See Also

[fit_t_comp](#) [CreateGeoObject](#) [likelihood_t_DD](#)

Examples

```
data(BGB.examples)
```

```
Canidae.phylo<-BGB.examples$Canidae.phylo
dummy.group<-c(rep("B",3),rep("A",12),rep("B",2),rep("A",6),rep("B",5),rep("A",6))
names(dummy.group)<-Canidae.phylo$tip.label
```

```
Canidae.simap<-phytools::make.simap(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")]
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),-0.000005)
maxN<-max(vapply(Canidae.geobyclass.object$geo.object$geography.object,
  function(x)max(rowSums(x)),1))

lh <- -likelihood_subgroup_model(data=Canidae.A, phylo=Canidae.geobyclass.object$map,
  geography.object=Canidae.geobyclass.object$geo.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 `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

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

See Also

[fit_t_comp likelihood_t_DD_geog](#)

Examples

```
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 `sigma2` and slope values and with a `geography` object formed using [CreateGeoObject](#).

Usage

```
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 $\log(\text{sig}^2)$ (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, "DDlin" is the diversity-dependent linear model, and "DDexp" is the diversity-dependent exponential model of Weir & Mursleen 2013.
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 sig^2 values at any point in time (see Details).

Details

When specifying par, $\log(\text{sig}^2)$ (see Note) must be listed before the slope parameter (b or r).

maxN can be calculated using $\text{maxN} = \max(\text{vapply}(\text{geo.object}\$geography.object, \text{function}(x) \max(\text{rowSums}(x)), 1))$, 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, sig^2 and slope values, and geography.object.

Note

To stabilize optimization, this function exponentiates the input sig^2 value, thus the user must input the $\log(\text{sig}^2)$ value to compute the correct log likelihood (see example).

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

Julien Clavel

References

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

See Also

[fit_t_comp](#) [CreateGeoObject](#) [likelihood_t_DD](#)

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(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)

```

likelihood_t_env	<i>Likelihood of a dataset under environmental models of trait evolution.</i>
------------------	---

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

```
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 <code>phylo\$tip.label</code>
...	"param", "fun", "times", "mtot" and "error" arguments. -param: a vector with the parameters used in the environmental function. The first value is σ^2 and the second is β . -fun: a time continuous function of an environmental variable (see e.g. <code>?fit_t_env</code>) -times: a vector of branching times starting at zero (e.g. <code>max(branching.times(phylo)) - branching.times(phylo)</code>) -mtot: root age of the tree (e.g. <code>max(branching.times(phylo))</code>) -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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Science*, 114(16): 4183-4188.

See Also

[fit_t_env](#)

Examples

```

if(test){
  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)
  mtot<-max(brtime)
  times<-mtot-brtime

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

  # Provide the environmental function rather than the dataset (faster if used recursively)
  #require(pspline)
  #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=mtot, model="EnvExp")

}

```

likelihood_t_MC	<i>Likelihood of a dataset under the matching competition model.</i>
-----------------	--

Description

Computes the likelihood of a dataset under the matching competition model with specified σ^2 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 <code>phylo\$tip.label</code>
par	a vector listing a value for $\log(\sigma^2)$ (see Note) and S (parameters of the matching competition model), in that order

Details

When specifying `par`, $\log(\sigma^2)$ 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 σ^2 and S values

Note

To stabilize optimization, this function exponentiates the input σ^2 value, thus the user must input the $\log(\sigma^2)$ value to compute the correct log likelihood (see example).

Author(s)

Jonathan Drury jonathan.p.drury@gmail.com

Julien Clavel

References

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Nuismer, S. & Harmon, L. 2015. Predicting rates of interspecific interaction from phylogenetic trees. *Ecology Letters* 18:17-27.

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

<code>phylo</code>	an object of type 'phylo' (see ape documentation)
<code>data</code>	a named vector of continuous data with names corresponding to <code>phylo\$tip.label</code>
<code>par</code>	a vector listing a value for $\log(\text{sig}^2)$ (see Note) and <code>S</code> (parameters of the matching competition model), in that order
<code>geo.object</code>	a geography object indicating sympatry through time, created using CreateGeoObject

Details

When specifying `par`, $\log(\text{sig}^2)$ 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 $\log(\text{sig}^2)$ value to compute the correct log likelihood (see example).

Author(s)

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

References

Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020

Nuismer, S. & Harmon, L. 2015. Predicting rates of interspecific interaction from phylogenetic trees. *Ecology Letters* 18:17-27.

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	<i>Add to a plot line segments joining the phenotypic evolutionary rate through time estimated by the fit_t_env function</i>
-----------------	--

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 <i>fit_t_env</i> fit.
steps	the number of steps from the root to the present used to compute the evolutionary rate σ^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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Science*, 114(16): 4183-4188.

See Also

[plot.fit_t.env](#), [likelihood_t_env](#)

Examples

```
if(test){

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")

}
```

lines.fit_t.env.ou *Add to a plot line segments joining the phenotypic evolutionary optimum through time estimated by the fit_t_env_ou function*

Description

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

Usage

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

Arguments

x an object of class 'fit_t.env.ou' obtained from a *fit_t_env_ou* fit.
steps the number of steps from the root to the present used to compute the optimum $\theta(t)$ through time.
... further arguments to be passed to plot. See ?plot.

Value

lines.fit_t.env.ou 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.
values the estimated optimum 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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Sciences*, 114(16): 4183-4188.

Troyer, E., Betancur-R, R., Hughes, L., Westneat, M., Carnevale, G., White W.T., Pogonoski, J.J., Tyler, J.C., Baldwin, C.C., Orti, G., Brinkworth, A., Clavel, J., Arcila, D., 2022. The impact of paleoclimatic changes on body size evolution in marine fishes. *Proceedings of the National Academy of Sciences*, 119 (29), e2122486119.

Goswami, A. & Clavel, J., 2024. Morphological evolution in a time of Phenomics. *EcoEvoRxiv*, <https://doi.org/10.32942/X22G7Q>

See Also

[plot.fit_t.env.ou](#), [fit_t_env_ou](#)

Examples

```
if(test){

data(InfTemp)
set.seed(9999) # for reproducibility

# Let's start by simulating a trait under a climatic OU
beta = 0.6          # relationship to the climate curve
sim_theta = 4       # value of the optimum if the relationship to the climate
# curve is 0 (this corresponds to an 'intercept' in the linear relationship used below)
sim_sigma2 = 0.025  # variance of the scatter = sigma^2
sim_alpha = 0.36    # alpha value = strength of the OU; quite high here...
delta = 0.001       # time step used for the forward simulations => here its 1000y steps
tree <- pbtree(n=200, d=0.3) # simulate a bd tree with some extinct lineages
root_age = 60       # height of the root (almost all the Cenozoic here)
tree$edge.length <- root_age*tree$edge.length/max(nodeHeights(tree))
# here - for this contrived example - I scale the tree so that the root is at 60 Ma

trait <- sim_t_env_ou(tree, sigma=sqrt(sim_sigma2), alpha=sim_alpha, theta0=sim_theta, param=beta,
  env_data=InfTemp, step=0.01, scale=TRUE, plot=FALSE)

## Fit the Environmental model (default)

result1 <- fit_t_env_ou(phylo = tree, data = trait, env_data =InfTemp,
  method = "Nelder-Mead", df=50, scale=TRUE)
plot(result1, lty=2)

result2 <- fit_t_env_ou(phylo = tree, data = trait, env_data =InfTemp,
  method = "Nelder-Mead", df=10, scale=TRUE)
lines(result2, col="red")

}
```

`make_gen.BipartiteEvol`*Compute the genealogies for BipartiteEvol*

Description

Compute the genealogies from a run of BipartiteEvol

Usage

```
make_gen.BipartiteEvol(out, treeP = NULL, treeH = NULL, verbose = T)
```

Arguments

<code>out</code>	The output of a run of <code>sim.BipartiteEvol</code>
<code>treeP</code>	Optional, a previous genealogy for clade P to which the new tree will be grafted (used if <code>out</code> was the continuation of a former run, see in the example)
<code>treeH</code>	Optional, a previous genealogy for clade H to which the new tree will be grafted (used if <code>out</code> was the continuation of a former run, see in the example)
<code>verbose</code>	Should the progression of the computation be printed?

Value

a list object with

<code>P</code>	The genealogy of the clade P
<code>H</code>	The genealogy of the clade H

Author(s)

O. Maliet

References

Maliet, O., Loeuille, N. and Morlon, H. (2020), An individual-based model for the eco-evolutionary emergence of bipartite interaction networks. *Ecol Lett.* doi:10.1111/ele.13592

See Also

[sim.BipartiteEvol](#)

Examples

```

if(test){
# run the model
set.seed(1)
mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 800,
                       D = 3, muP = 0.1 , muH = 0.1,
                       alphaP = 0.12,alphaH = 0.12,
                       rP = 10, rH = 10,
                       verbose = 100, thin = 5)

#build the genealogies
gen = make_gen.BipartiteEvol(mod)
plot(gen$H)

#compute the phylogenies
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#plot the result
plot_div.BipartiteEvol(gen,phy1, 1)

#build the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = nx, spatial = FALSE)

## add time steps to a former run
seed=as.integer(10)
set.seed(seed)

mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 200,
                       D = 3, muP = 0.1 , muH = 0.1,
                       alphaP = 0.12,alphaH = 0.12,
                       rP = 10, rH = 10,
                       verbose = 100, thin = 5,
                       P=mod$P,H=mod$H) # former run output

# update the genealogy
gen = make_gen.BipartiteEvol(mod,
                             treeP=gen$P, treeH=gen$H)

# update the phylogenies...
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#... and the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = 10, spatial = FALSE)

```

}

`mantel_test`*Compute Mantel test*

Description

This function computes a Mantel test between two dissimilarity matrices. The available correlations are Pearson, Spearman, and Kendall.

Usage

```
mantel_test(formula = formula(data), data = sys.parent(),
            correlation = "Pearson", nperm = 1000)
```

Arguments

<code>formula</code>	formula $y \sim x$ describing the test to be conducted where y and x are distance matrices (as "dist" objects).
<code>data</code>	an optional data frame containing the variables in the model as columns of dissimilarities. By default, the variables are taken from the current environment.
<code>correlation</code>	indicates which correlation (R) must be used among Pearson (default), Spearman, and Kendall correlations.
<code>nperm</code>	a number of permutations to evaluate the significance of the correlation. By default, it equals 1000, but this can be very long for the Kendall correlation.

Details

This function is adapted from the function `mantel` in the R-package `ecodist` (Goslee & Urban, 2007).

Value

<code>mantelr</code>	Mantel correlation (R).
<code>pval1</code>	one-tailed p-value (null hypothesis: $R \leq 0$).
<code>pval2</code>	one-tailed p-value (null hypothesis: $R \geq 0$).
<code>pval3</code>	two-tailed p-value (null hypothesis: $R = 0$).

Author(s)

Benoît Perez-Lamarque

References

Perez-Lamarque B, Maliet O, Pichon B, Selosse M-A, Martos F, Morlon H. 2022. Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. bioRxiv, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology. <https://doi.org/10.1101/2021.08.30.458192>

Goslee, S.C. & Urban, D.L. (2007). The ecodist package for dissimilarity-based analysis of ecological data. J. Stat. Softw., 22, 1–19.

Mantel, N. 1967. The detection of disease clustering and a generalized regression approach. Cancer Research 27:209-220.

See Also

[phylosignal_network](#)

[phylosignal_sub_network](#)

Examples

```
# Measuring phylogenetic signal in species interactions using a Mantel test
# (do closely related species interact with similar partners?)

library(RPANDA)

# Load the data
data(mycorrhizal_network)

network <- mycorrhizal_network[[1]] # bipartite interaction matrix
tree_orchids <- mycorrhizal_network[[2]] # phylogenetic tree (phylo object)

network <- network[,tree_orchids$tip.label]

ecological_distances <- as.matrix(vegan::vegdist(t(network), "jaccard", binary=FALSE))

phylogenetic_distances <- cophenetic.phylo(tree_orchids)

mantel_test(as.dist(ecological_distances) ~ as.dist(phylogenetic_distances),
  correlation="Pearson", nperm = 10000)
```

`mantel_test_nbpartners`

Compute Mantel test

Description

This function tests for phylogenetic signal in species interactions in guild A using a Mantel test that keep constant the number of partners per species.

Usage

```
mantel_test_nbpartners(network, tree_A, tree_B = NULL, method="Jaccard_binary",
  nperm = 1000, correlation = "Pearson")
```

Arguments

network	a matrix representing the bipartite interaction network with species from guild A in columns and species from guild B in rows. Row names (resp. column names) must correspond to the tip labels of tree B (resp. tree A).
tree_A	a phylogenetic tree of guild A (the columns of the interaction network). It must be an object of class "phylo".
tree_B	(optional) a phylogenetic tree of guild B (the rows of the interaction network). It must be an object of class "phylo".
method	indicates which method is used to compute the phylogenetic signal in species interactions. If you want to perform a Mantel test between the phylogenetic distances and some ecological distances (do closely related species interact with similar partners?), you can choose "Jaccard_weighted" (default) for computing the ecological distances using Jaccard dissimilarities (or "Jaccard_binary" to not take into account the abundances of the interactions), "Bray-Curtis" for computing the Bray-Curtis dissimilarity, or "GUniFrac" for computing the weighted (or generalized) UniFrac distances ("UniFrac_unweighted" to not take into account the interaction abundances).
correlation	indicates which correlation (R) must be used among Pearson (default) and Spearman correlations.
nperm	a number of permutations to evaluate the significance of the correlation. By default, it equals 1000.

Value

mantelr	Mantel correlation (R).
pval1	one-tailed p-value (null hypothesis: $R \leq 0$).
pval2	one-tailed p-value (null hypothesis: $R \geq 0$).
pval3	two-tailed p-value (null hypothesis: $R = 0$).

Author(s)

Benoît Perez-Lamarque

References

Perez-Lamarque B, Maliet O, Pichon B, Selosse M-A, Martos F, Morlon H. 2022. Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. *bioRxiv*, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology. <https://doi.org/10.1101/2021.08.30.458192>

Mantel, N. 1967. The detection of disease clustering and a generalized regression approach. *Cancer Research* 27:209-220.

See Also

[phylosignal_network](#)
[phylosignal_sub_network](#)
[mantel_test](#)

Examples

```

# Measuring phylogenetic signal in species interactions using a Mantel test
# with permutations keeping constant the number of partners per species

library(RPANDA)

# Load the data
data(mycorrhizal_network)

network <- mycorrhizal_network[[1]] # bipartite interaction matrix
tree_orchids <- mycorrhizal_network[[2]] # phylogenetic tree (phylo object)

# mantel_test_nbpartners(network, tree_orchids, method="Jaccard_weighted",
# correlation="Pearson", nperm = 1000)

```

modelSelection

Phenotypic model selection from tip trait data.

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

modelSelection-methods

~~ *Methods for Function modelSelection* ~~

Description

~~ Methods for function modelSelection ~~

Methods

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

MPhiFFT

A class used internally to compute ClaDS's likelihood

Description

This class represents a matrix $A = (1/\text{rowSums}(\text{Toep})) * \text{Toep}$ where Toep is a Toeplitz matrix.

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS](#)

mycorrhizal_network *Mycorrhizal network from La Réunion island*

Description

Mycorrhizal interconnection network between orchids and mycorrhizal fungi from La Réunion island (Martos et al., 2012) along with the reconstructed phylogenetic trees of the orchids and the fungal OTUs.

Usage

```
data(mycorrhizal_network)
```

Details

These phylogenies were constructed by maximum likelihood inference from four plastid genes for the orchids and one nuclear gene for the fungi. See Martos et al. (2012) for details.

Source

Martos, F., Munoz, F., Pailler, T., Kottke, I., Gonneau, C. & Selosse, M.-A. (2012). The role of epiphytism in architecture and evolutionary constraint within mycorrhizal networks of tropical orchids. *Mol. Ecol.*, 21, 5098–5109.

References

Martos, F., Munoz, F., Pailler, T., Kottke, I., Gonneau, C. & Selosse, M.-A. (2012). The role of epiphytism in architecture and evolutionary constraint within mycorrhizal networks of tropical orchids. *Molecular Ecology*, 21, 5098–5109.

Perez-Lamarque B, Maliet O, Pichon B, Selosse M-A, Martos F, Morlon H. 2022. Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. *bioRxiv*, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology. <https://doi.org/10.1101/2021.08.30.458192>

Examples

```
data(mycorrhizal_network)

network <- mycorrhizal_network[[1]] # interaction matrix
tree_orchids <- mycorrhizal_network[[2]] # phylogenetic tree (phylo object)
tree_fungi <- mycorrhizal_network[[3]] # phylogenetic tree (phylo object)
```

 ostracoda

Ostracod diversity since the Jurassic

Description

Ostracod fossil diversity since the Jurassic

Usage

```
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 two 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

Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832

Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

Examples

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

 paleodiv

Paleodiversity through time

Description

Calculates paleodiversity through time from shift.estimate output with the deterministic approach.

Usage

```
paleodiv(phylo, data, sampling.fractions, shift.res,
         backbone.option = "crown.shift", combi = 1,
         time.interval = 1, split.div = F)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
data	a data.frame containing a database of monophyletic groups for which potential shifts can be investigated. This database should be based on taxonomy, ecology or traits and contain a column named "Species" with species name as in phylo.
sampling.fractions	the output resulting from get.sampling.fractions.
shift.res	the output resulting from shift.estimate.
backbone.option	type of the backbone analysis: <ul style="list-style-type: none"> • "stem.shift": paleodiversity dynamics are calculated from the stem age for subclades. • "crown.shift": paleodiversity dynamics are calculated from the crown age for subclades.
combi	numeric. The combination of shifts defined by its rank in the global comparison.
time.interval	numeric. Define the time interval (in million years) at which paleodiversity values are calculated. Default is 1 for a value at each million year.
split.div	boolean. Specifies if paleodiversity should be plitted by parts of the selected combination (TRUE) or not.

Value

If split.div = FALSE, paleodiversity dynamics are returned in a matrix with as many rows as parts in the selected combination and as many column as million years from the root to the present. If spit.div = TRUE, global paleodiversity dynamic is returned as a vector with a value per million year.

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[shift.estimate](#), [apply_prob_dtt](#)

Examples

```
# loading data
data("Cetacea")
data("taxo_cetacea")
data("shifts_cetacea")
```

```

# no shifts tested at genus level
taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]
f_cetacea <- get.sampling.fractions(phylo = Cetacea, lad = FALSE,
                                  data = taxo_cetacea_no_genus,
                                  plot = TRUE, cex = 0.3)

# use of paleodiv
paleodiversity <- paleodiv(phylo = Cetacea,
                           data = taxo_cetacea_no_genus,
                           sampling.fractions = f_cetacea,
                           shift.res = shifts_cetacea,
                           combi = 1, split.div = FALSE)

```

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", ...)`.

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

Examples

```
showClass("PhenotypicACDC")
```

```
PhenotypicADiag-class Class "PhenotypicADiag"
```

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

Examples

```
showClass("PhenotypicADiag")
```

PhenotypicBM-class *Class* "PhenotypicBM"

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

Examples

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

PhenotypicDD-class *Class "PhenotypicDD"*

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.

Methods

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

Author(s)

Marc Manceau

References

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

Examples

```
showClass("PhenotypicGMM")
```

 PhenotypicModel-class *Class* "PhenotypicModel"

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

Examples

```
showClass("PhenotypicOU")
```

```
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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*, and the associated Supplementary material.

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

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

Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

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

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

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

Revell, L.J., 2009. Size-correction and principal components for intraspecific comparative studies. *Evolution*, 63:3258-3268.

Clavel, J., Aristide, L., Morlon, H., 2019. A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst. Biol.* 68: 93-116.

See Also

[fit_t_pl](#), [ancestral](#), [GIC.fit_pl.rpanda](#), [gic_criterion](#)

Examples

```
if(test){
  if(require(mvMORPH)){
    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

- Bininda-Emonds, O. R., et al. (2007) The delayed rise of present-day mammals *Nature* 446: 507-512
- Rolland, J., Condamine, F. L., Jiguet, F., & Morlon, H. (2014) Faster speciation and reduced extinction in the tropics contribute to the mammalian latitudinal diversity gradient. *PLoS Biol*, 12(1): e1001775.

See Also

[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)
```

phylosignal_network *Compute phylogenetic signal in a bipartite interaction network*

Description

This function computes the phylogenetic signal in a bipartite interaction network, either the phylogenetic signal in species interactions (do closely related species interact with similar partners?) using Mantel tests, or the phylogenetic signal in the number of partners (i.e. degree; do closely related species interact with the same number of partners?) using Mantel tests or using the Phylogenetic bipartite linear model (PBLM) from Ives and Godfray (2006). Mantel tests measuring the phylogenetic signal in species interactions can be computed using quantified or binary networks, with the Jaccard, Bray-Curtis, or UniFrac ecological distances.

Usage

```
phylosignal_network(network, tree_A, tree_B = NULL,
  method = "Jaccard_weighted", nperm = 10000,
  correlation = "Pearson", only_A = FALSE, permutation = "shuffle")
```

Arguments

network	a matrix representing the bipartite interaction network with species from guild A in columns and species from guild B in rows. Row names (resp. column names) must correspond to the tip labels of tree B (resp. tree A).
tree_A	a phylogenetic tree of guild A (the columns of the interaction network). It must be an object of class "phylo".
tree_B	(optional) a phylogenetic tree of guild B (the rows of the interaction network). It must be an object of class "phylo".
method	indicates which method is used to compute the phylogenetic signal in species interactions. If you want to perform a Mantel test between the phylogenetic distances and some ecological distances (do closely related species interact with similar partners?), you can choose "Jaccard_weighted" (default) for computing the ecological distances using Jaccard dissimilarities (or "Jaccard_binary" to not take into account the abundances of the interactions), "Bray-Curtis" for computing the Bray-Curtis dissimilarity, or "GUniFrac" for computing the weighted (or generalized) UniFrac distances ("UniFrac_unweighted" to not take into account the interaction abundances). Conversely, if you want to evaluate the phylogenetic signal in the number of partners (do closely related species interact with the same number of partners?), you can choose "degree". Alternatively (not recommended), you can use the Phylogenetic Bipartite Linear Model "PBLM" (see Ives and Godfray, 2006) or "PBLM_binary" to not consider the abundances of the interactions.
correlation	(optional) indicates which correlation (R) must be used in the Mantel test, among Pearson (default), Spearman, and Kendall correlations. It only applies for the

	methods "Jaccard_weighted", "Jaccard_binary", "Bray-Curtis", "GUniFrac", "UniFrac_unweighted", or "degree".
nperm	(optional) a number of permutations to evaluate the significance of the Mantel test. By default, it equals 10,000, but this can be very long for the Kendall correlation. It only applies for the methods "Jaccard_weighted", "Bray-Curtis", "Jaccard_binary", "GUniFrac", "UniFrac_unweighted", or "degree".
permutation	(optional) indicates which permutations must be performed to evaluate the significance of the Mantel correlation: either "shuffle" (by default - i.e. random shuffling of the distance matrix) or "nbparkers" (i.e. keeping constant the number of partners per species and shuffling at random their identity).
only_A	(optional) indicates whether the signal should be only computed for guild A (and not for guild B). By default, it is computed for both guilds if "tree_B" is provided.

Details

See the tutorial on GitHub (https://github.com/BPerezLamarque/Phylosignal_network).

Value

For Mantel tests, the function outputs a vector of up to 8 values: the number of species in guild A ("nb_A"), the number of species in guild B ("nb_B"), the correlation for guild A ("mantel_cor_A"), its associated upper p-value ("pvalue_upper_A", i.e. the fraction of permutations that led to higher correlation values), its associated lower p-value ("pvalue_lower_A", i.e. the fraction of permutations that led to lower correlation values), and (optional) the correlation for guild B ("mantel_cor_B"), its associated upper p-value ("pvalue_upper_B"), and its associated lower p-value ("pvalue_lower_B"),

"mantel_cor_A" (or "mantel_cor_B") indicates the strength of the phylogenetic signal in guild A (or B). The upper p-value "pvalue_upper_A" (or "pvalue_upper_B") indicates the significance of the phylogenetic signal in guild A (or B). The lower p-value "pvalue_lower_A" (or "pvalue_lower_B") indicates the significance of the anti-phylogenetic signal in guild A (or B). For instance, if "pvalue_upper_A"<0.05, there is a significant phylogenetic signal in guild A.

For the PBLM approach (Ives and Godfray, 2006), the function outputs a vector of 8 values: the number of species in guild A ("nb_A"), the number of species in guild B ("nb_B"), the phylogenetic signals in guilds A ("dA") and B ("dB"), the covariance of interaction matrix ("MSETotal"), the mean square error of the complete model ("MSEFull"), the mean square error of model run on star phylogenies ("MSEStar"), and the mean square error of the model assuming strict Brownian motion evolutions ("MSEBase"). The significance of the phylogenetic signal can be evaluated by comparing "MSEFull" and "MSEStar".

Author(s)

Benoît Perez-Lamarque

References

Perez-Lamarque B, Maliet O, Pichon B, Selosse M-A, Martos F, Morlon H. 2022. Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction

networks. bioRxiv, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology. <https://doi.org/10.1101/2021.08.30.458192>

Goslee, S.C. & Urban, D.L. (2007). The ecodist package for dissimilarity-based analysis of ecological data. *J. Stat. Softw.*, 22, 1–19.

Ives, A.R. & Godfray, H.C.J. (2006). Phylogenetic analysis of trophic associations. *Am. Nat.*, 168, E1–E14.

Kembel, S.W., Cowan, P.D., Helmus, M.R., Cornwell, W.K., Morlon, H., Ackerly, D.D., et al. (2010). Picante: R tools for integrating phylogenies and ecology. *Bioinformatics*, 26, 1463–1464.

Chen, J., Bittinger, K., Charlson, E.S., Hoffmann, C., Lewis, J., Wu, G.D., et al. (2012). Associating microbiome composition with environmental covariates using generalized UniFrac distances. *Bioinformatics*, 28, 2106–2113.

See Also

[phylosignal_sub_network](#)

Examples

```
# Load the data
data(mycorrhizal_network)

network <- mycorrhizal_network[[1]] # interaction matrix
tree_orchids <- mycorrhizal_network[[2]] # phylogenetic tree (phylo object)
tree_fungi <- mycorrhizal_network[[3]] # phylogenetic tree (phylo object)

if(test){

# Using Mantel tests:

# Step 1: Phylogenetic signal in species interactions
# (do closely related species interact with similar partners?)

phylosignal_network(network, tree_A = tree_orchids, tree_B = tree_fungi,
method = "GUniFrac", correlation = "Pearson", nperm = 10000) # measured for both guilds

# Step 2: Phylogenetic signal in species interactions when accounting
# for the signal in the number of partners
# Mantel test with permutations that keep constant the number of partners per species

phylosignal_network(network, tree_A = tree_orchids, tree_B = tree_fungi,
method = "GUniFrac", correlation = "Pearson", nperm = 1000, permutation = "nbpartners")

# Other: Phylogenetic signal in the number of partners
# (do closely related species interact with the same number of partners?)

phylosignal_network(network, tree_A = tree_orchids, method = "degree",
```

```

correlation = "Pearson", nperm = 10000) # for guild A
phylosignal_network(t(network), tree_A = tree_fungi, method = "degree",
correlation = "Pearson", nperm = 10000) # for guild B

# Alternative using PBLM (not recommended) - very slow

# phylosignal_network(network, tree_A = tree_orchids, tree_B = tree_fungi, method = "PBLM")
}

```

```

phylosignal_sub_network

```

Compute clade-specific phylogenetic signals in a bipartite interaction network

Description

This function computes the clade-specific phylogenetic signals in species interactions. For each node of tree A having a certain number of descending species, it computes the phylogenetic signal in the resulting sub-network by performing a Mantel test between the phylogenetic distances and the ecological distances for the given sub-clade of tree A. Mantel tests can be computed using quantified or binary networks, with the Jaccard, Bray-Curtis, or UniFrac ecological distances.

Usage

```

phylosignal_sub_network(network, tree_A, tree_B = NULL,
method = "Jaccard_weighted", nperm = 1000,
correlation = "Pearson", minimum = 10, degree = FALSE,
permutation = "shuffle")

```

Arguments

network	a matrix representing the bipartite interaction network with species from guild A in columns and species from guild B in rows. Row names (resp. column names) must correspond to the tip labels of tree B (resp. tree A).
tree_A	a phylogenetic tree of guild A (the columns of the interaction network). It must be an object of class "phylo".
tree_B	(optional) a phylogenetic tree of guild B (the rows of the interaction network). It must be an object of class "phylo".
method	indicates which method is used to compute the phylogenetic signal in species interactions using Mantel tests. You can choose "Jaccard_weighted" (default) for computing ecological distances using Jaccard dissimilarities (or "Jaccard_binary" to not take into account the abundances of the interactions), "Bray-Curtis" for computing the Bray-Curtis dissimilarity, or "GUniFrac" for computing the weighted (or generalized) UniFrac distances ("UniFrac_unweighted" to not take into account the interaction abundances).

correlation	indicates which correlation (R) must be used in the Mantel test, among Pearson (default), Spearman, and Kendall correlations.
nperm	a number of permutations to evaluate the significance of the Mantel test. By default, it equals 10,000, but this can be very long for the Kendall correlation.
permutation	(optional) indicates which permutations must be performed to evaluate the significance of the Mantel correlation: either "shuffle" (by default - i.e. random shuffling of the distance matrix) or "nbparkers" (i.e. keeping constant the number of partners per species and shuffling at random their identity).
minimum	indicates the minimal number of descending species for a node in tree A to compute its clade-specific phylogenetic signal.
degree	if degree=TRUE, Mantel tests testing for phylogenetic signal in the number of partners are additionally performed in each sub-clade.

Details

See the tutorial on GitHub (https://github.com/BPerezLamarque/Phylosignal_network).

Value

For Mantel tests, the function outputs a table where each line corresponds to a tested clade and which contains at least 8 columns: the name of the node ("node"), the number of species in the sub-clade A ("nb_A"), the number of species in guild B associated with the sub-clade A ("nb_B"), the Mantel correlation for guild A ("mantel_cor"), its associated upper p-value ("pvalue_upper"), its associated lower p-value ("pvalue_lower"), and the associated Bonferroni corrected p-values ("pvalue_upper_corrected" and "pvalue_lower_corrected").

"mantel_cor" indicates the strength of the phylogenetic signal in the sub-clade A. The upper p-value "pvalue_upper" indicates the significance of the phylogenetic signal in the sub-clade A. The lower p-value "pvalue_lower" indicates the significance of the anti-phylogenetic signal in the sub-clade A. Both Bonferroni p-values are corrected using the number of tested nodes. For instance, if "pvalue_upper_corrected"<0.05 for a given node, there is a significant phylogenetic signal in the corresponding sub-clade of A.

If degree=TRUE, it also indicates in each sub-clade, the phylogenetic signal in the number of partners ("degree_mantel_cor") and its significance with or without the Bonferroni correction ("degree_pvalue_upper", "degree_pvalue_lower" and "degree_pvalue_upper_corrected", "degree_pvalue_lower_corrected").

Author(s)

Benoît Perez-Lamarque

References

- Perez-Lamarque B, Maliet O, Pichon B, Selosse M-A, Martos F, Morlon H. 2022. Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. *bioRxiv*, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology. <https://doi.org/10.1101/2021.08.30.458192>
- Goslee, S.C. & Urban, D.L. (2007). The ecodist package for dissimilarity-based analysis of ecological data. *J. Stat. Softw.*, 22, 1–19.

Chen, J., Bittinger, K., Charlson, E.S., Hoffmann, C., Lewis, J., Wu, G.D., et al. (2012). Associating microbiome composition with environmental covariates using generalized UniFrac distances. *Bioinformatics*, 28, 2106–2113.

See Also

[phylosignal_sub_network](#) [plot_phylosignal_sub_network](#)

Examples

```
# Load the data
data(mycorrhizal_network)

network <- mycorrhizal_network[[1]] # interaction matrix
tree_orchids <- mycorrhizal_network[[2]] # phylogenetic tree (phylo object)
tree_fungi <- mycorrhizal_network[[3]] # phylogenetic tree (phylo object)

if(test){

# Clade-specific phylogenetic signal in species interactions in guild A
# (do closely related species interact with similar partners in sub-clades of guild A?)

results_clade_A <- phylosignal_sub_network(network, tree_A = tree_orchids, tree_B = tree_fungi,
method = "GUniFrac", correlation = "Pearson", degree = TRUE)
plot_phylosignal_sub_network(tree_A = tree_orchids, results_clade_A, network)

# Clade-specific phylogenetic signal in species interactions in guild B
# (do closely related species interact with similar partners in sub-clades of guild B?)

results_clade_B <- phylosignal_sub_network(t(network), tree_A = tree_fungi, tree_B = tree_orchids,
method = "GUniFrac", correlation = "Pearson", degree = TRUE)
plot_phylosignal_sub_network(tree_A = tree_fungi, results_clade_B, t(network))
}
```

pi_estimator

Compute nucleotidic diversity (Pi estimator)

Description

This function computes the Pi estimator of genetic diversity (Nei and Li, 1979) while controlling for the presence of gaps in the alignment (Ferretti et al, 2012), frequent in barcoding datasets.

Usage

```
pi_estimator(sequences)
```


Arguments

sequences a matrix representing the nucleotidic alignment of all the sequences present in the phylogenetic tree.

Value

An estimate of genetic diversity

Author(s)

Ana C. Afonso Silva & Benoît Perez-Lamarque

References

Nei M & Li WH, Mathematical model for studying genetic variation in terms of restriction endonucleases, 1979, Proc. Natl. Acad. Sci. USA.

Ferretti L, Raineri E, Ramos-Onsins S. 2012. Neutrality tests for sequences with missing data. Genetics 191: 1397–1401.

Perez-Lamarque B, Öpik M, Maliet O, Silva A, Selosse M-A, Martos F, and Morlon H. 2022. Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont, Molecular Ecology, 31:3496–512.

See Also

[theta_estimator](#) [delineate_phylotypes](#)

Examples

```
data(woodmouse)

alignment <- as.character(woodmouse) # nucleotidic alignment

pi_estimator(alignment)
```

plot.fit_t.env	<i>Plot the phenotypic evolutionary rate through time estimated by the fit_t_env function</i>
----------------	---

Description

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

Usage

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

Arguments

x	an object of class 'fit_t.env' obtained from a <i>fit_t.env</i> fit.
steps	the number of steps from the root to the present used to compute the evolutionary rate σ^2 through time.
...	further arguments to be passed to plot. See ?plot.

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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Science*, 114(16): 4183-4188.

See Also

[lines.fit_t.env](#), [likelihood_t.env](#)

Examples

```
if(test){
  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.fit_t.env.ou *Plot the phenotypic evolutionary optimum through time estimated by the fit_t_env_ou function*

Description

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

Usage

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

Arguments

x	an object of class 'fit_t.env.ou' obtained from a <i>fit_t_env_ou</i> fit.
steps	the number of steps from the root to the present used to compute the optimum $\theta(t)$ through time.
...	further arguments to be passed to plot. See ?plot.

Value

plot.fit_t.env.ou 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.
values	the estimated optimum values 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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Sciences*, 114(16): 4183-4188.

Troyer, E., Betancur-R, R., Hughes, L., Westneat, M., Carnevale, G., White W.T., Pogonoski, J.J., Tyler, J.C., Baldwin, C.C., Orti, G., Brinkworth, A., Clavel, J., Arcila, D., 2022. The impact of paleoclimatic changes on body size evolution in marine fishes. *Proceedings of the National Academy of Sciences*, 119 (29), e2122486119.

Goswami, A. & Clavel, J., 2024. Morphological evolution in a time of Phenomics. *EcoEvoRxiv*, <https://doi.org/10.32942/X22G7Q>

See Also

[lines.fit_t.env](#), [fit_t.env.ou](#), [lines.fit_t.env.ou](#)

Examples

```

if(test){
  data(InfTemp)

  set.seed(9999) # for reproducibility

  # Let's start by simulating a trait under a climatic OU
  beta = 0.6           # relationship to the climate curve
  sim_theta = 4        # value of the optimum if the relationship to the climate curve is 0
  # (this corresponds to an 'intercept' in the linear relationship used below)
  sim_sigma2 = 0.025   # variance of the scatter = sigma^2
  sim_alpha = 0.36     # alpha value = strength of the OU; quite high here...
  delta = 0.001       # time step used for the forward simulations => here its 1000y steps
  tree <- pbtree(n=200, d=0.3) # simulate a bd tree with some extinct lineages
  root_age = 60        # height of the root (almost all the Cenozoic here)
  tree$edge.length <- root_age*tree$edge.length/max(nodeHeights(tree))
  # here - for this contrived example - I scale the tree so that the root is at 60 Ma

  trait <- sim_t_env_ou(tree, sigma=sqrt(sim_sigma2), alpha=sim_alpha,
                       theta0=sim_theta, param=beta, env_data=InfTemp, step=0.01,
                       scale=TRUE, plot=FALSE)

  ## Fit the Environmental model (default)

  result1 <- fit_t_env_ou(phylo = tree, data = trait, env_data =InfTemp,
                        method = "Nelder-Mead", df=50, scale=TRUE)
  plot(result1, lty=2, col="red")
}

```

plot_BICompare	<i>Display modalities on a phylogeny.</i>
----------------	---

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_ClaDS0_chains *Plot the MCMC chains obtained when inferring ClaDS0 parameters*

Description

Plot the MCMC chains obtained with run_ClaDS0.

Usage

```
plot_ClaDS0_chains(sampler, burn = 1/2, thin = 1,  
                  param = c("sigma", "alpha", "l_0", "LP"))
```

Arguments

sampler	The output of a run_ClaDS0 run.
burn	Number of iterations to drop in the beginning of the chains.
thin	Thinning parameter, one iteration out of "thin" is plotted.
param	Either a vector of "character" elements with the name of the parameter to plot, or a vector of integers indicating what parameters to plot.

Author(s)

O. Maliet

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS0](#), [getMAPS_ClaDS0](#), [plot_ClaDS_chains](#)

Examples

```
data("ClaDS0_example")  
  
plot_ClaDS0_chains(ClaDS0_example$Cl0_chains)  
plot_ClaDS0_chains(ClaDS0_example$Cl0_chains, param = paste0("lambda_", c(1,10,5)))
```

plot_ClaDS_chains *Plot the MCMC chains obtained when inferring ClaDS parameters*

Description

Plot the MCMC chains obtained with fit_ClaDS.

Usage

```
plot_ClaDS_chains(sampler, burn = 1/2, thin = 1,  
                  param = c("sigma", "alpha", "mu", "LP"))
```

Arguments

sampler	The output of a fit_ClaDS run.
burn	Number of iterations to drop in the beginning of the chains.
thin	Thinning parameter, one iteration out of "thin" is plotted.
param	Either a vector of "character" elements with the name of the parameter to plot, or a vector of integers indicating what parameters to plot.

Author(s)

O. Maliet

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[fit_ClaDS](#), [getMAPS_ClaDS](#), [plot_ClaDS0_chains](#)

Examples

```
data("Caprimulgidae_ClaDS2")  
  
plot_ClaDS_chains(Caprimulgidae_ClaDS2$sampler)  
  
plot_ClaDS_chains(Caprimulgidae_ClaDS2$sampler, burn = 1/4,  
                  param = c("sigma", "alpha", "l_0", "LP"))  
  
plot_ClaDS_chains(Caprimulgidae_ClaDS2$sampler, burn = 1/5, thin = 5, param = c(1,5,6,15))
```

plot_ClaDS_phylo *Plot a phylogeny with branch-specific values*

Description

Plot a phylogeny with branches colored according to branch-specific rate values

Usage

```
plot_ClaDS_phylo(phylo, rates, rates2 = NULL,
                 same.scale = T, main = NULL, lwd = 2, log = T, show.tip.label = F, ...)
```

Arguments

phylo	An object of class 'phylo'.
rates	A vector containing the branch-specific rates, in the same order as phylo\$edges.
rates2	An optional second vector containing the branch-specific rates, in the same order as phylo\$edges. If NULL (the default), the tree is only plotted once with the rate values from rates. If not, the tree is plotted twice, with the rate values from rates in the left panel and those from rates2 in the right panel.
same.scale	A boolean specifying whether the values from rates and rates2 are plotted with the same colorscale. Default to TRUE.
main	A title for the plot.
lwd	Width of the tree branch lengths. Default to 2.
log	A boolean specifying whether the rates values are plotted on a log scale. Default to TRUE.
show.tip.label	A boolean specifying whether the labels of the phylogeny should be displayed. Default to FALSE.
...	Optional arguments for plot.phylo.

Author(s)

O. Maliet

References

Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0

See Also

[sim_ClaDS](#)

Examples

```

set.seed(1)

obj= sim_ClaDS( lambda_0=0.1,
               mu_0=0.5,
               sigma_lamb=0.7,
               alpha_lamb=0.90,
               condition="taxa",
               taxa_stop = 20,
               prune_extinct = TRUE)

tree = obj$tree
speciation_rates = obj$lamb[obj$rates]
extinction_rates = obj$mu[obj$rates]

par(mar=c(1,1,0,0))
plot_ClaDS_phylo(tree,speciation_rates)

plot_ClaDS_phylo(tree,speciation_rates, lwd = 4, log = FALSE)

```

plot_div.BipartiteEvol

Plot the output of BipartiteEvol

Description

Plot the genealogies and phylogenies simulated with BipartiteEvol

Usage

```

plot_div.BipartiteEvol(gen, spec, trait.id, lwdgen = 1,
                      lwdsp = lwdgen, scale = NULL)

```

Arguments

gen	The output of a run of make_gen.BipartiteEvol
spec	The output of a run of define_species.BipartiteEvol
trait.id	The trait dimension used to color the genealogies, phylogenies an network with trait values
lwdgen	Width of the branches of the genealogies, default to 1
lwdsp	Width of the branches of the phylogenies, default to 1
scale	Optional, used to force the trait scale

Details

The upper line shows the genealogies colored with trait values for both guilds (the number above shows the depth of the respective genealogy).

The second line shows the phylogenies colored with trait values for both guilds (the number above shows the tip number of the respective phylogeny).

On the third line there is, from left to right, the trait distribution within individuals in guild P, trait of the individual in H as a function of the trait of the interacting individual in P, and the trait distribution within individuals in guild H (for the dimension trait.id).

The lower line shows the quantitative interaction network, with species colored according to their mean trait value (for the dimension trait.id).

Author(s)

O. Maliet

References

Maliet, O., Loeuille, N. and Morlon, H. (2020), An individual-based model for the eco-evolutionary emergence of bipartite interaction networks. *Ecol Lett.* doi:10.1111/ele.13592

See Also

[sim.BipartiteEvol](#)

Examples

```
# run the model
set.seed(1)

if(test){
  mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 1000,
    D = 3, muP = 0.1 , muH = 0.1,
    alphaP = 0.12,alphaH = 0.12,
    rP = 10, rH = 10,
    verbose = 100, thin = 5)

  #build the genealogies
  gen = make_gen.BipartiteEvol(mod)
  plot(gen$H)

  #compute the phylogenies
  phy1 = define_species.BipartiteEvol(gen,threshold=1)

  #plot the result
  plot_div.BipartiteEvol(gen,phy1, 1)
}
```

plot_dtt	<i>Plot diversity through time</i>
----------	------------------------------------

Description

Plot the estimated number of species through time

Usage

```
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 <code>max(node.age(phylo)\$ages)</code> .
N0	number of extant species. If all extant species are represented in the phylogeny, N0 is given by <code>length(phylo\$tip.label)</code>

Value

Plot representing how the estimated number of species vary through time

Author(s)

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

Morlon, H. (2014) Phylogenetic approaches for studying diversification, *Eco Lett* 17:508-525

See Also

[fit_bd](#)

Examples

```
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,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	<i>Plot speciation, extinction & net diversification rate functions of a fitted model</i>
-------------	---

Description

Plot estimated speciation, extinction & net diversification rates through time

Usage

```
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

```

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 speciation, 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

```

if(require(pspline)){
data(Balaenopteridae)
tot_time<-max(node.age(Balaenopteridae)$ages)
data(InfTemp)
dof<-smooth.spline(InfTemp[,1], InfTemp[,2])$df

# Fit the pure birth model (no extinction) with exponential variation of the speciation rate
# with 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()
#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_net.BipartiteEvol

Plot the output of BipartiteEvol

Description

Plot the genealogies, phylogenies and interaction network simulated with BipartiteEvol

Usage

```
plot_net.BipartiteEvol(gen, spec, trait.id, link,
  out, lwdgen = 1, lwdsp = lwdgen, scale = NULL,
  nx = NULL, cor = F, network.method = "bipartite",
  spatial = F)
```

Arguments

gen	The output of a run of make_gen.BipartiteEvol
spec	The output of a run of define_species.BipartiteEvol
trait.id	The trait dimension used to color the genealogies, phylogenies an network with trait values
out	The output of a run of sim.BipartiteEvol
link	The output of a run of sim.BipartiteEvol
lwdgen	Width of the branches of the genealogies, default to 1
lwdsp	Width of the branches of the phylogenies, default to 1
scale	Optional, used to force the trait scale
nx	Grid size parameter used in sim.BipartiteEvol. If NULL, sqrt(N) is used, where N is the number of individuals in a guild
cor	If F (the default), the middle panel displays the interraction network with species positionned in trait space. If T, it shows all the individual in trait space
network.method	How should the network be plotted? Can be "bipartite" (the default) or "matrix"
spatial	Should the grid with trait values of the individual of both guilds been shown? Default to F

Details

The upper line shows the genealogies colored with trait values for both guilds (the number above shows the depth of the respective genealogy).

The second line shows the phylogenies colored with trait values for both guilds (the number above shows the tip number of the respective phylogeny).

On the third line there is, from left to right, the trait distribution within individuals in guild P (for the dimension trait.id), the interaction network with species positioned in trait space (if cor = T), and the trait distribution within individuals in guild H (for the dimension trait.id).

The lower line shows the quantitative interaction network, with species colored according to their mean trait value (for the dimension trait.id).

Author(s)

O. Maliet

References

Maliet, O., Loeuille, N. and Morlon, H. (2020), An individual-based model for the eco-evolutionary emergence of bipartite interaction networks. *Ecol Lett.* doi:10.1111/ele.13592

See Also

[sim.BipartiteEvol](#)

Examples

```
# run the model
set.seed(1)

if(test){
  mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 1000,
    D = 3, muP = 0.1 , muH = 0.1,
    alphaP = 0.12,alphaH = 0.12,
    rP = 10, rH = 10,
    verbose = 100, thin = 5)

  #build the genealogies
  gen = make_gen.BipartiteEvol(mod)
  plot(gen$H)

  #compute the phylogenies
  phy1 = define_species.BipartiteEvol(gen,threshold=1)

  #build the network
  net = build_network.BipartiteEvol(gen, phy1)

  trait.id = 1
  plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = 10, spatial = FALSE)
}
```

plot_phylosignal_sub_network

Plot clade-specific phylogenetic signals in a bipartite interaction network

Description

This function plots the clade-specific phylogenetic signals in species interactions. For each node of tree A having a certain number of descending species, it represents the phylogenetic signal in the resulting sub-network by performing a Mantel test between the phylogenetic distances and the ecological distances for the given sub-clade of tree A.

Usage

```
plot_phylosignal_sub_network(tree_A, results_sub_clades, network, legend=TRUE,
show.tip.label=FALSE, where="bottomleft")
```

Arguments

`tree_A` a phylogenetic tree of guild A (the columns of the interaction network). It must be an object of class "phylo".

`results_sub_clades` output of the function `phylosignal_sub_network`.

`network` a matrix representing the bipartite interaction network with species from guild A in columns and species from guild B in rows. Row names (resp. column names) must correspond to the tip labels of tree B (resp. tree A).

`legend` indicates whether the legend should be plotted.

`show.tip.label` indicates whether the tip labels should be plotted.

`where` indicates where to put the legend (default is "bottomleft").

Details

See the tutorial on GitHub (https://github.com/BPerezLamarque/Phylosignal_network).

Value

A phylogenetic tree with nodes colored according to the clade-specific phylogenetic signals. Blue nodes are not significant (Bonferonni correction), whereas orange-red nodes present significant phylogenetic signals and their color indicates the strength of the signal (correlation R of the Mantel test).

Author(s)

Benoît Perez-Lamarque

References

- Perez-Lamarque B, Maliet O, Pichon B, Selosse M-A, Martos F, Morlon H. 2022. Do closely related species interact with similar partners? Testing for phylogenetic signal in bipartite interaction networks. *bioRxiv*, 2021.08.30.458192, ver. 6 peer-reviewed and recommended by Peer Community in Evolutionary Biology. <https://doi.org/10.1101/2021.08.30.458192>
- Goslee, S.C. & Urban, D.L. (2007). The ecodist package for dissimilarity-based analysis of ecological data. *J. Stat. Softw.*, 22, 1–19.
- Chen, J., Bittinger, K., Charlson, E.S., Hoffmann, C., Lewis, J., Wu, G.D., et al. (2012). Associating microbiome composition with environmental covariates using generalized UniFrac distances. *Bioinformatics*, 28, 2106–2113.

See Also

[phylosignal_network](#) [phylosignal_sub_network](#)

Examples

```
# Load the data
data(mycorrhizal_network)

network <- mycorrhizal_network[[1]] # interaction matrix
tree_orchids <- mycorrhizal_network[[2]] # phylogenetic tree (phylo object)
tree_fungi <- mycorrhizal_network[[3]] # phylogenetic tree (phylo object)

if(test){

# Clade-specific phylogenetic signal in species interactions in guild A
# (do closely related species interact with similar partners in sub-clades of guild A?)

results_clade_A <- phylosignal_sub_network(network, tree_A = tree_orchids, tree_B = tree_fungi,
method = "GUniFrac", correlation = "Pearson")
plot_phylosignal_sub_network(tree_A = tree_orchids, results_clade_A, network)

# Clade-specific phylogenetic signal in species interactions in guild B
# (do closely related species interact with similar partners in sub-clades of guild B?)

results_clade_B <- phylosignal_sub_network(t(network), tree_A = tree_fungi, tree_B = tree_orchids,
method = "GUniFrac", correlation = "Pearson")
plot_phylosignal_sub_network(tree_A = tree_fungi, results_clade_B, t(network))
}
```

plot_phylo_comb

Plot shifts of diversification on a phylogeny

Description

Plots the phylogeny with colored branches according to shifts of diversification.

Usage

```
plot_phylo_comb(phylo, data, sampling.fractions, shift.res = NULL,
               combi, backbone.option = "crown.shift",
               main = NULL, col.sub = NULL, col.bck = "black",
               lty.bck = 1, tested_nodes = F, lad = T,
               leg = T, text.cex = 1, pch.cex = 1, ...)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
data	a data.frame containing a database of monophyletic groups for which potential shifts can be tested. This database should be based on taxonomy, ecology or traits and must contain a column named "Species" with species names as in phylo.
sampling.fractions	the output resulting from get.sampling.fractions.
shift.res	the output resulting from shift.estimate or NULL (default). This latter case allows to represent combinations only from the output of <code>get.comb.shift</code> by specifying the combination (see argument <code>combi</code>).
combi	character or numeric. If <code>shift.res</code> is provided, this argument is a numeric and corresponds to the rank of the combination in the global comparison (<code>shift.res\$total</code>). If <code>shift.res</code> is NULL, this argument should be a character giving a combination of node IDs as in <code>get.comb.shift</code> output. This latter manner to specify combination allows to visualize a combination of shifts before having results.
backbone.option	type of the backbone analysis (see <code>backbone.option</code> in <code>shift.estimate</code> for more details): <ul style="list-style-type: none"> • "stem.shift": the stems of subclades are included in subclade analyses; • "crown.shift": the stems of subclades are included in the backbone analysis (Default).
main	Character. The name of the plot. Default is NULL and the combination rank with AICc will be printed if <code>shift.res</code> is not NULL.
col.sub	character. A vector to specify colors of subclade(s). Can be let NULL (see details).
col.bck	character. A vector to specify colors of backbone(s). Default is "black" for simple backbone (see details).
lad	boolean. Allows to ladderize the tree.
leg	boolean. If TRUE, legend of the selected combination is added to the plot with names from data and best model names. Default is TRUE. The position is automatically adjusted in function of <code>lad</code> argument.
lty.bck	numeric. Define lty for the backbone.
tested_nodes	boolean. If TRUE, all the tested nodes are highlighted by a red point.
text.cex	numeric. Define the size of legend text.
pch.cex	numeric. Define the size of points if <code>tested_nodes = TRUE</code>
...	further arguments to be passed to <code>plot</code> or to <code>plot.phylo</code> .

Details

If col.sub is not specified, color vector for subclades is c(c(brewer.pal(8, "Dark2"),brewer.pal(8, "Set1"),"darkmagenta","dodgerblue2", "orange", "forestgreen")). For multiple backbone, default vector is c("blue4", "orange4", "red4", "grey40", "coral4", "deeppink4", "khaki4", "darkolivegreen", "darkslategray","black"). ... allows to set different graphical parameters from plot.phylo such as cex for size of tip labels or edge.width for the thickness of the phylogeny edges.

Value

plot the phylogeny and returns the same invisible object as plot.phylo.

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[shift.estimate](#)

Examples

```
# loading data
data("Cetacea")
data("taxo_cetacea")

taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]

# main procedure
f_cetacea <- get.sampling.fractions(phylo = Cetacea, lad = FALSE,
                                  data = taxo_cetacea_no_genus,
                                  plot = TRUE, cex = 0.3)

comb.shift_cetacea <- get.comb.shift(phylo = Cetacea,
                                    data = taxo_cetacea_no_genus,
                                    sampling.fractions = f_cetacea,
                                    Ncores = 4)

# use of plot_phylo_comb
# without shift.estimate results but with comb.shift_cetacea

plot_phylo_comb(phylo = Cetacea,
                data = taxo_cetacea,
                sampling.fractions = f_cetacea,
                combi = comb.shift_cetacea[15],
                label.offset = 0.3,
```

```
main = "", lad = FALSE ,cex = 0.4)
```

plot_prob_dtt

Plot diversity through time with confidence intervals.

Description

Plots confidence intervals of the estimated number of species through time using a matrix of probabilities given by the function 'prob_dtt'.

Usage

```
plot_prob_dtt(mat, grain = 0.1, plot.prob = TRUE,
              plot.mean = TRUE, int = TRUE, plot.bound=FALSE,
              conf = 0.95, add = FALSE, col.mean = "red", col.bound = "blue",
              lty="solid", lwd=1, lty.bound=1, add.present=T, ...)
```

Arguments

mat	matrix of probabilities, with species numbers as rows and times as columns with rownames and colnames set to the values of each.
grain	the upper limit of a range of probabilities plotted in a gray scale (lower limit is zero). Higher probabilities are plotted in black. Default value is 0.1.
plot.prob	logical: set to TRUE (default value) to plot the probabilities.
plot.mean	logical: set to TRUE (default value) to plot a line for the mean.
plot.bound	logical: set to TRUE to plot the bounds of the confidence interval, int must be set to TRUE.
int	logical: set to TRUE (default value) to plot a confidence interval.
conf	confidence level. The default value is 0.95.
add	logical: set to TRUE to add the plot on an existing graph.
col.mean	color of the line for the mean.
col.bound	color of the confidence interval bounds
lty	style of the line for the mean (if added on a current plot)
lwd	the line width, a positive number (default to 1)
lty.bound	style of the line for the bound (if added on a current plot)
add.present	whether or not to add the present diversity value to the plot. Default is TRUE.
...	further arguments to be passed to plot or to plot.phylo.

Details

The function assumes that the matrix of probabilities 'mat' has species numbers as rows and times as columns with rownames and colnames set to the values of each.

'Grain' must be between 0 and 1. If the plot is too pale 'grain' should be diminished (and inversely if the plot is too dark)

Value

Plot representing how the estimated number of species vary through time with confidence intervals. The darker is the plot, the higher is the probability.

Author(s)

O.Billaud, T.L.Parsons, D.S.Moen, 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.

Billaud, O., Moen, D. S., Parsons, T. L., Morlon, H. (under review) Estimating Diversity Through Time using Molecular Phylogenies: Old and Species-Poor Frog Families are the Remnants of a Diverse Past.

See Also

[fit_bd](#), [plot_dtt](#), [prob_dtt](#)

Examples

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

if(test){
# 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,f=1,
                 expo.lamb = TRUE, fix.mu=TRUE)

# Compute the matrix of probabilities
prob <- prob_dtt(result, tot_time, 1:tot_time, N0=9, type="crown")

# Check that the sums of probabilities are equal to 1
colSums(prob)

# Plot Diversity through time
plot_prob_dtt(prob)
}
```

plot_spectR	<i>Spectral density plot of a phylogeny.</i>
-------------	--

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)
```

Posdef

Positive definite symmetric matrices

Description

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

Usage

```
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

Uyeda J.C., Caetano D.S., Pennell M.W. 2015. Comparative Analysis of Principal Components Can be Misleading. *Syst. Biol.* 64:677-689.

Clavel, J., Aristide, L., Morlon, H., 2019. A Penalized Likelihood framework for high-dimensional phylogenetic comparative methods and an application to new-world monkeys brain evolution. *Syst. Biol.* 68:93-116.

See Also

[GIC.fit_pl.rpanda](#), [fit_t_pl](#) [phyl.pca_pl](#)

Examples

```

if(test){
  if(require(mvMORPH)){
    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)
  }
}

```

prob_dtt

Confidence intervals of diversity through time

Description

Returns a matrix of probabilities to have 'm' species at a given time 't' with 'n' observed extant species (complete sampling or not) and 's' species at the root of the phylogeny (s=1 if the tree has a stem, otherwise s=2)

Usage

```

prob_dtt(fit.bd, tot_time, time, N0, l=N0, f = 1/N0,
         m = seq(N0), method="simple", lin = FALSE,
         prec = 1000, type = "stem", logged = TRUE)

```

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 <code>max(node.age(phylo)\$ages)</code> .
time	vector of times on which the function calculates the probabilities of 'm' species. The function goes forward in time, so that $t = 0$ is the time of the most recent common ancestor.
N0	number of extant species. If all extant species are represented in the phylogeny, N0 is given by <code>length(phylo\$tip.label)</code> .
l	number of extant species sampled. Default value is N0 (complete sampling).
f	the fraction of extant species included in the phylogeny, given by <code>l/N0</code> .

m	a vector of integers for which we want to know the probability of each value.
method	reflects which way of computing is chosen. A 'simple' one (quicker) is used when the number of extant species (N0) is known exactly or when the whole phylogeny is sampled (f==1). A 'hard one', much longer, is used when N0 is not known without doubt and f<1. The default value is "simple" (the other possibility is "hard")
lin	logical: set to TRUE if λ & μ are fitted with a linear model.
prec	precision (number of bits used) of the computation. The default value is 1000.
type	reflects whether the clade has a stem or not. Options are the default "stem" and the alternative "crown", which means the tree starts with two species at time 0.
logged	logical: set to TRUE to log probabilities and factorials as much as possible (required, except perhaps for very small, young clades).

Details

If the sampling fraction is not equal to 1, the function computes with very high numbers. To be sufficiently accurate, the package 'Rmpfr' is used and "prec" is the precision of the computation. Hence, the calculation may take a lot of time. In case of wrong probabilities (negatives or higher than 1 for instance) you should increase the precision.

If the sampling fraction is equal to 1, the function doesn't need the package 'Rmpfr' and simply uses the log of probabilities and factorials (argument "logged"). Thus, computation is faster.

The matrix columns names go backward in time.

Value

Matrix of probabilities to have 'm' species at a given time 't' with 'n' observed extant species (complete sampling or not).

Author(s)

O.Billaud, T.L.Parsons, D.S.Moen, 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.

Billaud, O., Moen, D. S., Parsons, T. L., Morlon, H. (under review) Estimating Diversity Through Time using Molecular Phylogenies: Old and Species-Poor Frog Families are the Remnants of a Diverse Past.

See Also

[fit_bd](#), [plot_dtt](#), [plot_prob_dtt](#)

Examples

```

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()

if(test){
result <- fit_bd(Balaenopteridae,tot_time,f.lamb,f.mu,lamb_par,mu_par,f=1,
                expo.lamb = TRUE, fix.mu=TRUE)

# Compute the matrix of probabilities
prob <- prob_dtt(result, tot_time, 1:tot_time, N0=9, type="crown")

# Check that the sums of probabilities are equal to 1
colSums(prob)
}

```

radiolaria

Radiolaria diversity since the Jurassic

Description

Radiolaria fossil diversity since the Jurassic

Usage

```
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

- Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832
- Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

Examples

```
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

Lazarus, D. (1994) Neptune: A marine micropaleontology database *Mathematical Geology* 26:817–832

Alroy, J. (2010) Geographical, environmental and intrinsic biotic controls on Phanerozoic marine diversification: Controls on phanerozoic marine diversification *Palaeontology* 53:1211–1235

Examples

```
data(redalgae)
plot(redalgae)
```

remove.model	<i>Removing a model from shift.estimate output</i>
--------------	--

Description

Allows to remove a model from the model comparisons of shift.estimate output.

Usage

```
remove.model(shift.res, model)
```

Arguments

shift.res	the output resulting from shift.estimate.
model	character. Specifies the model to remove from the set of model of diversification applied in shift.res.

Details

This function allow to remove model one at a time. The idea is to remove a model without having to reanalyse the phylogeny and all the combinations of shifts if a model (e.g. BVAR_DVAR) behaves strangely on the studied phylogeny.

Value

the same output resulting from shift.estimate but without the chosen model in model comparisons.

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[shift.estimate](#)

Examples

```
# loading data
data("shifts_cetacea")

# Removing "BVAR_DCST" model for the example
shifts_cetacea_noBVAR_DCST <- remove.model(shift.res = shifts_cetacea,
                                           model = "BVAR_DCST")
```

sealevel	<i>Sea level data since the Jurassic</i>
----------	--

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

Miller, K.G., Kominz, M.A., Browning, J.V., Wright, J.D., Mountain, G.S., Katz, M.E., Sugarman, P.J., Cramer, B.S., Christie-Blick, N., Pekar, S.F. (2005) The Phanerozoic Record of Global Sea-Level Change *Science* 310:1293-1298

Examples

```
data(sealevel)
plot(sealevel)
```

shift.estimate *Estimating clade-shifts of diversification*

Description

Applies models of diversification to each part of all combinations of shifts to detect the best combination of subclades and backbone(s).

Usage

```
shift.estimate(phylo, data, sampling.fractions, comb.shift,
               models = c("BCST", "BCST_DCST", "BVAR",
                           "BVAR_DCST", "BCST_DVAR", "BVAR_DVAR"),
               backbone.option = "crown.shift",
               multi.backbone = F, np.sub = 4,
               rate.max = NULL, n.max = NULL, Ncores = 1)
```

Arguments

- phylo** an object of type 'phylo' (see ape documentation)
- data** a data.frame containing a database of monophyletic groups for which potential shifts can be investigated. This database should be based on taxonomy, ecology or traits and contain a column named "Species" with species name as in phylo.
- sampling.fractions** the output resulting from get.sampling.fractions.
- comb.shift** the output resulting from get.comb.shift.
- models** a vector of character that specifies the set of models of diversification to apply. Default is c("BCST", "BCST_DCST", "BVAR", "BVAR_DCST", "BCST_DVAR", "BVAR_DVAR").
- backbone.option** type of the backbone analysis:
- "stem.shift": for every shift, the probability of the speciation event at the stem age of the subclade is included in the likelihood of the backbone thanks to the argument spec_times.
 - "crown.shift": for every shift, both the probability of the speciation event at the stem age of the subclade and the probability that the stem of the subclade survives to the crown age are included in the likelihood of the backbone thanks to the argument branch_times.
- multi.backbone** can be either FALSE (default), TRUE or "all":
- FALSE: only combinations with simple backbone will be analyzed.
 - TRUE: only combinations with multiple backbones will be analyzed.
 - "all": all combinations are analyzed.

np.sub	Defines the set of models to apply to subclade based on the number of parameters. By default np.sub = 4 and all models from argument models will be applied. If np.sub = 3, the more complex model "BVAR_DVAR" is excluded. If np.sub = 2, the set of models is reduced to "BCST", "BCST_DCST" and "BVAR" models. np.sub = "no_extinction" only applies "BCST" and "BVAR" models.
rate.max	numeric. Define a maximum value for diversification rate through time.
n.max	numeric. Define a maximum value for diversity through time.
Ncores	numeric. Define the number of CPU cores to use for parallelizing the computation of combinations.

Details

The output for backbone is a list in which each element corresponds to the backbone model comparisons of a combination. This element contains a list with one table of model comparison per backbone.

We recommend to remove "BVAR_DVAR" model from the models set and to lead the first analysis with multi.backbone = F to limit the number of combination.

clade.size argument should be the same value for the whole procedure (same than for get.sampling.fraction and get.comb.shift).

Value

a list with the following components

whole_tree	a data.frame with the model comparison for the whole tree
subclades	a list of dataframes summarizing the model comparison for all subclades (same format than div.models outputs)
backbones	a list with the model comparison for all backbones (see details)
total	the global comparison of combinations based on AICc

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[get.sampling.fractions](#), [shift.estimate](#), [paleodiv](#)

Examples

```

# loading data
data("Cetacea")
data("taxo_cetacea")

# whole procedure
taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]
f_cetacea <- get.sampling.fractions(phylo = Cetacea, lad = FALSE,
                                  data = taxo_cetacea_no_genus,
                                  plot = TRUE, cex = 0.3)

comb.shift_cetacea <- get.comb.shift(phylo = Cetacea,
                                    data = taxo_cetacea_no_genus,
                                    sampling.fractions = f_cetacea,
                                    Ncores = 4)

shifts_cetacea <- shift.estimate(phylo = Cetacea,
                                 data = taxo_cetacea_no_genus,
                                 sampling.fractions = f_cetacea,
                                 comb.shift = comb.shift_cetacea,
                                 models = c("BCST", "BCST_DCST", "BVAR",
                                           "BVAR_DCST", "BCST_DVAR"),
                                 backbone.option = "crown.shift",
                                 Ncores = 4)

```

shifts_cetacea

Cetacean shift.estimate results

Description

Results of shift.estimate applied to Cetaceans

Usage

```
data(shifts_cetacea)
```

Details

This object is the result of shifts.estimate applied to the Cetacean phylogeny as in the example of shift.estimate function.

Source

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

References

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

Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

Examples

```
data(shifts_cetacea)
print(shifts_cetacea)
```

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

Misra, S., Froelich, P.N. (2012) Lithium isotope history of Cenozoic seawater: Changes in silicate weathering and reverse weathering. *Science* 335(6070):818–823

Cermeno, P., Falkowski, P.G., Romero, O.E., Schaller, M.F., Vallina, S.M. (2015) Continental erosion and the Cenozoic rise of marine diatoms *Proceedings of the National Academy of Sciences* 112:4239-244

Examples

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

sim.BipartiteEvol *Simulation of the BipartiteEvol model*

Description

Simulate of the BipartiteEvol model from Maliet et al. (2020)

Usage

```
sim.BipartiteEvol(nx, ny = nx, NG, dSpace = Inf, D = 1, muP,
muH, alphaP = 0, alphaH = 0, iniP = 0, iniH = 0, nP = 1, nH = 1,
rP = 1, rH = 1, effect = 1, verbose = 100, thin = 1, P = NULL, H = NULL)
```

Arguments

nx	Size of the grid (the grid has size nx * ny)
ny	Size of the grid (default to nx, the grid has size nx * ny)
NG	Number of time step the model is run
dSpace	Size of the dispersal kernel (default to Inf, meaning there are no restrictions on dispersion)
D	Dimension of the trait space (default to 3)
muP	Mutation probability at reproduction for the individuals of clade P
muH	Mutation probability at reproduction for the individuals of clade H
alphaP	alpha parameter for clade P (1/alpha is the niche width)
alphaH	alpha parameter for clade H (1/alpha is the niche width)
iniP	Initial trait value for the individuals in clade P
iniH	Initial trait value for the individuals in clade P
nP	Number of individuals of clade P killed at each time step
nH	Number of individuals of clade H killed at each time step
rP	r parameter for clade P (r is the ratio between the fitness maximum and minimum)
rH	r parameter for clade H (r is the ratio between the fitness maximum and minimum)
effect	Standard deviation of the trait mutation kernel
verbose	The simulation
thin	The number of iterations between two recording of the state of the model (default to 1)
P	Optionnal, used to continue one precedent run: traits of the individuals of clade P at the end of the precedent run
H	Optionnal, used to continue one precedent run: traits of the individuals of clade H at the end of the precedent run

Value

a list with	
Pgenealogy	The genealogy of clade P
Hgenealogy	The genealogy of clade H
xP	The trait values at each time step for clade P
xH	The trait values at each time step for clade H
P	The trait values at present for clade P
H	The trait values at present for clade H
Pmut	The number of new mutations at each time step for clade P
Hmut	The number of new mutations at each time step for clade H
iniP	The initial trait values for the individuals of clade P used in the simulation
iniH	The initial trait values for the individuals of clade H used in the simulation
thin.factor	The thin value used in the simulation

Author(s)

O. Maliet

References

Maliet, O., Loeuille, N. and Morlon, H. (2020), An individual-based model for the eco-evolutionary emergence of bipartite interaction networks. *Ecol Lett.* doi:10.1111/ele.13592

Examples

```
# run the model
set.seed(1)

if(test){
  mod = sim.BipartiteEvol(nx = 8, ny = 4, NG = 500,
                        D = 3, muP = 0.1, muH = 0.1,
                        alphaP = 0.12, alphaH = 0.12,
                        rP = 10, rH = 10,
                        verbose = 100, thin = 5)

  #build the genealogies
  gen = make_gen.BipartiteEvol(mod)
  plot(gen$H)

  #compute the phylogenies
  phy1 = define_species.BipartiteEvol(gen, threshold=1)

  #plot the result
  plot_div.BipartiteEvol(gen, phy1, 1)

  #build the network
```

```

net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = 10, spatial = FALSE)

## add time steps to a former run
seed=as.integer(10)
set.seed(seed)

mod = sim.BipartiteEvol(nx = 8,ny = 4,NG = 500,
                       D = 3, muP = 0.1 , muH = 0.1,
                       alphaP = 0.12,alphaH = 0.12,
                       rP = 10, rH = 10,
                       verbose = 100, thin = 5,
                       P=mod$P,H=mod$H) # former ru output

# update the genealogy
gen = make_gen.BipartiteEvol(mod,
                             treeP=gen$P, treeH=gen$H)

# update the phylogenies...
phy1 = define_species.BipartiteEvol(gen,threshold=1)

#... and the network
net = build_network.BipartiteEvol(gen, phy1)

trait.id = 1
plot_net.BipartiteEvol(gen,phy1,trait.id, net,mod, nx = 10, spatial = FALSE)
}

```

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

```
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 <i>sig2</i> in [, 1], <i>m</i> in [, 2], <i>alpha</i> in [, 3], <i>root.value</i> in [, 4], <i>psi</i> of the OU model for the non-focal, resource use trait in [, 5], and <i>theta</i> 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 ($\max(\text{nodeHeights}(\text{phylo}))/\text{Nsegments}$) should be much smaller than the smallest branch ($\min(\text{phylo}\$edge.length)$).

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

Drury, J., Grether, G., Garland Jr., T., and Morlon, H. 2017. A review of phylogenetic methods for assessing the influence of interspecific interactions on phenotypic evolution. *Systematic Biology*

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)
```

sim.divergence.geo *Simulation of trait data under the model of divergent character displacement described in Drury et al. 2017*

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 <i>sig2</i> in [, 1], <i>m</i> in [, 2], <i>alpha</i> in [, 3], <i>root.value</i> in [, 4], <i>psi</i> of the OU model in [, 5], and <i>theta</i> 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 Nsegments will impact the length of time the simulations take. The length of each segment ($\max(\text{nodeHeights}(\text{phylo}))/\text{Nsegments}$) should be much smaller than the smallest branch ($\min(\text{phylo}\$edge.length)$).

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

Drury, J., Grether, G., Garland Jr., T., and Morlon, H. 2017. A review of phylogenetic methods for assessing the influence of interspecific interactions on phenotypic evolution. *Systematic Biology*

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(0.05,2,1,0,c(2,0),0)
sim.divergence.geo(phylo,pars,Nsegments=2500, plot=FALSE, geo.object)

```

simul.comb.shift	<i>Simulating trees from shift.estimate() results to test model adequacy</i>
------------------	--

Description

Simulates trees with combination of shifts from shifts.estimate() output.

Usage

```

simul.comb.shift(n = 10000, phylo, sampling.fractions,
                shift.res, combi = 1, clade.size = 5)

```

Arguments

n	numeric. Defines the number of simulations to generate (see Details).
phylo	an object of type 'phylo' (see ape documentation).
sampling.fractions	the output resulting from get.sampling.fractions.
shift.res	the output resulting from shift.estimate.
combi	numeric. Corresponds to the rank of the combination in the global comparison (shift.res\$total).
clade.size	numeric. Defines the minimum number of species in a subgroup. Default is 5.

Details

Some combinations of shifts might be complex cases to simulate because the backbone needs to be rich enough to graft subclades. Some simulations will not satisfy this condition and will then be discarded. In consequence, the number of simulated phylogenies in the output will not be equal to n for complex simulations. This is why the value of n is high by default (n = 10000), to ensure to have enough simulations (around 500) to test the presence.

clade.size argument should be the same value for the whole procedure in the empirical case (same than for get.sampling.fraction and get.comb.shift).

Value

a list of simulated phylogenies as object of type phylo. Tips of subclades are named with the letters a, b, c, etc. while tips of backbones are named with letters z, y, etc. The empirical groups are sorted from the more recent to the older one (i.e. group a will be the more recent empirical subclade, etc.).

Author(s)

Nathan Mazet

References

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

See Also

[shift.estimate](#)s

Examples

```
# loading data
data("Cetacea")
data("taxo_cetacea")
data("shifts_cetacea")

# with the results from shifts.estimate()

# no shifts tested at genus level
taxo_cetacea_no_genus <- taxo_cetacea[names(taxo_cetacea) != "Genus"]

f_cetacea <- get.sampling.fractions(phylo = Cetacea,
                                  data = taxo_cetacea_no_genus)

all_posteriors_cetacea <- simul.comb.shift(phylo = Cetacea,
                                           sampling.fractions = f_cetacea,
                                           shift.res = shifts_cetacea)
```

simulateTipData

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

Manceau M., Lambert A., Morlon H. (2017) A unifying comparative phylogenetic framework including traits coevolving across interacting lineages *Systematic Biology*

Examples

```
#Loading an example tree
newick <- "(((A:1,B:0.5):2,(C:3,D:2.5):1):6,E:10.25):2,(F:6.5,G:8.25):3):1;"
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)
```

 simulateTipData-methods

 ~~ *Methods for Function simulateTipData* ~~

Description

~~ Methods for function simulateTipData ~~

Methods

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

 sim_ClaDS

Simulation of the ClaDS model

Description

Simulate a birth-death phyloh-geny with rate shifts happening at speciation events.

Usage

```
sim_ClaDS(lambda_0, mu_0,
           new_lamb_law="lognormal*shift", new_mu_law="turnover",
           condition="time", time_stop = 0, taxa_stop = Inf,
           sigma_lamb=0.1, alpha_lamb=1, lamb_max=1, lamb_min=0,
           sigma_mu=0, alpha_mu=1, mu_min=mu_0, mu_max=mu_0,
           theta=1, nShiftMax=Inf,
           return_all_extinct=FALSE, prune_extinct=TRUE,
           maxRate=Inf)
```

Arguments

lambda_0	Initial speciation rate.
mu_0	Initial extinction rate, or turnover rate if new_mu_law == "turnover".
new_lamb_law	Distribution in which the new speciation rates are drawn at a speciation event. See details.
new_mu_law	Distribution in which the new extinction rates are drawn at a speciation event. See details.
condition	Stopping condition. Can be "time" (the default) or "taxa".
time_stop	Stopping time if condition == "time".

taxa_stop	Final number of species if condition == "taxa". If condition == "time", the process is stopped if the number of species exceeds taxa_stop. This can be useful for some parametrizations of the model for which the number of species can reach very large number very quickly, leading to computation time and memory issues. To disable this option, use taxa_stop = Inf (the default).
sigma_lamb	Parameter of the new speciation rates distribution, see details.
alpha_lamb	Parameter of the new speciation rates distribution, see details.
lamb_max	Parameter of the new speciation rates distribution, see details.
lamb_min	Parameter of the new speciation rates distribution, see details.
sigma_mu	Parameter of the new extinction rates distribution, see details.
alpha_mu	Parameter of the new extinction rates distribution, see details.
mu_min	Parameter of the new extinction rates distribution, see details.
mu_max	Parameter of the new extinction rates distribution, see details.
theta	Probability to have a rate shift at speciation. Default to 1.
nShiftMax	Maximum number of rate shifts. If nShiftMax < Inf, theta is set to 0 as soon as there has been nShiftMax rate shifts. Set nShiftMax = Inf (the default) to disable this option.
return_all_extinct	Boolean specifying whether the function should return extinct phylogenies. Default to FALSE.
prune_extinct	Boolean specifying whether extinct species should be removed from the resulting phylogeny. Default to TRUE.
maxRate	The process is stopped if one of the lineage has a speciation rate that exceeds maxRate. This can be useful for some parametrizations of the model for which the rates can reach very large values, leading to numerical overflows. To disable this option, use maxRate = Inf (the default).

Details

Available options for new_lamb_law are :

- "uniform", the new speciation rates are drawn uniformly in [lamb_min, lamb_max].
- "normal", the new speciation rates are drawn in a normal distribution with parameters (sigma_lamb², parent_lambda), truncated in 0.
- "lognormal", the new speciation rates are drawn in a lognormal distribution with parameters (sigma_lamb², parent_lambda).
- "lognormal*shift", the new speciation rates are drawn in a lognormal distribution with parameters (sigma_lamb², parent_lambda * alpha_lamb). This is the default option as it corresponds to the ClaDS model.
- "lognormal*t", the new speciation rates are drawn in a lognormal distribution with parameters (sigma_lamb² * t², parent_lambda), where t is the age of the mother species.

- "logbrownian", the new speciation rates are drawn in a lognormal distribution with parameters ($\sigma_{\text{lamb}}^2 * t$, parent_lambda), where t is the age of the mother species. This is used to approximate the case where speciation rates are evolving as the log of a brownian motion, as is done in Beaulieu, J. M. and B. C. O'Meara. (2015).
- "normal+shift", the new speciation rates are drawn in a normal distribution with parameters (σ_{lamb}^2 , $\text{parent_lambda} + \alpha_{\text{lamb}}$), truncated in 0.
- "normal*shift", the new speciation rates are drawn in a normal distribution with parameters (σ_{lamb}^2 , $\text{parent_lambda} * \alpha_{\text{lamb}}$), truncated in 0.

Available options for `new_mu_law` are :

- "uniform", the new extinction rates are drawn uniformly in [μ_{min} , μ_{max}].
- "normal", the new extinction rates are drawn in a normal distribution with parameters (σ_{mu}^2 , parent_mu), truncated in 0.
- "lognormal", the new extinction rates are drawn in a lognormal distribution with parameters (σ_{mu}^2 , parent_mu).
- "lognormal*shift", the new extinction rates are drawn in a lognormal distribution with parameters (σ_{mu}^2 , $\text{parent_mu} * \alpha_{\text{mu}}$).
- "normal*t", the new speciation rates are drawn in a normal distribution with parameters ($\sigma_{\text{lamb}}^2 * t^2$, parent_lambda), where t is the age of the mother species.
- "turnover", the turnover rate is constant (in that case μ_0 is the turnover rate), so the new extinction rates are μ_0 times the new speciation rates. This is the default option, corresponding to ClaDS2.

Value

A list with :

<code>tree</code>	The resulting phylogeny.
<code>times</code>	A vector with the times of all speciation and extinction events.
<code>nblineages</code>	A vector in which <code>nblineages[i]</code> is the number of species in the clade after the event happening at time <code>times[i]</code> .
<code>lamb</code>	A vector with all the different speciation rates resulting from the simulation.
<code>mu</code>	A vector with all the different extinction rates resulting from the simulation.
<code>rates</code>	A vector of integer mapping the elements of <code>.\$lamb</code> and <code>.\$mu</code> to the branches of <code>.\$tree</code> .
<code>maxRate</code>	A boolean indicating whether the process was ended before reaching the specified stopping criterion because one of the speciation rates exceeded <code>maxRate</code> (see the "arguments" section).
<code>root_length</code>	The time before the first speciation event.

Author(s)

O. Maliet

References

- Maliet O., Hartig F. and Morlon H. 2019, A model with many small shifts for estimating species-specific diversification rates, *Nature Ecology and Evolution*, doi 10.1038/s41559-019-0908-0
- Beaulieu, J. M. and B. C. O'Meara. 2015. Extinction can be estimated from moderately sized molecular phylogenies. *Evolution* 69:1036-1043.

See Also

[plot_ClaDS_phylo](#)

Examples

```
# Simulation of a ClaDS2 phylogeny
set.seed(1)

obj= sim_ClaDS( lambda_0=0.1,
                mu_0=0.5,
                sigma_lamb=0.7,
                alpha_lamb=0.90,
                condition="taxa",
                taxa_stop = 20,
                prune_extinct = TRUE)

tree = obj$tree
speciation_rates = obj$lamb[obj$rates]
extinction_rates = obj$mu[obj$rates]

plot_ClaDS_phylo(tree,speciation_rates)

# Simulation of a phylogeny with constant extinction rate and speciation
# rates evolving as a logbrownian
set.seed(4321)

obj= sim_ClaDS( lambda_0=0.1,
                mu_0=0.2,
                new_mu_law = "uniform",
                new_lamb_law = "logbrownian",
                sigma_lamb=0.4,
                condition="taxa",
                taxa_stop = 20,
                prune_extinct = FALSE)

tree = obj$tree
speciation_rates = obj$lamb[obj$rates]
extinction_rates = obj$mu[obj$rates]

par(mar=c(1,1,0,0))
plot_ClaDS_phylo(tree,speciation_rates)
```

```

# Simulation of a phylogeny with constant extinction rate and at most one shift
# in speciation rates
set.seed(1221)

obj= sim_ClaDS( lambda_0=0.1,
               mu_0=0.05,
               new_mu_law = "uniform",
               new_lamb_law = "uniform",
               lamb_max = 0.5, lamb_min = 0,
               theta = 0.1, nShiftMax = 1,
               condition="taxa",
               taxa_stop = 100,
               prune_extinct = TRUE)

tree = obj$tree
speciation_rates = obj$lamb[obj$rates]
extinction_rates = obj$mu[obj$rates]

plot_ClaDS_phylo(tree,speciation_rates)

```

sim_env_bd

Simulate birth-death tree dependent on an environmental curve

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 λ 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 μ 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).
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 <i>sm.spline</i> 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
- Condamine, F.L., Rolland, J., and Morlon, H. (2013) Macroevolutionary perspectives to environmental 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){y[1] * exp(y[2] * x)}
f.mu<-function(t,x,y){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)
```

sim_MCBD

Simulation of macroevolutionary diversification under the integrated model described in Aristide & Morlon 2019

Description

Simulates the joint diversification of species and a continuous trait, where changes in both dimensions are interlinked through competitive interactions.

Usage

```
sim_MCBD(pars, root.value = 0, age.max = 50, step.size = 0.01, bounds = c(-Inf,Inf),
          plot = TRUE, ylims=NULL, full.sim = FALSE)
```

Arguments

pars Vector of simulation parameters:

- pars[1] corresponds to *lambda*1, the speciation initiation rate
- pars[2] corresponds to *tau*0, the basal speciation completion rate
- pars[3] corresponds to *beta*, the effect of trait differences on the speciation completion rate
- pars[4] corresponds to *mu*0, the competitive extinction parameter for good species
- pars[5] corresponds to *mubg*, the background good species extinction rate
- pars[6] corresponds to *mui*0, the competitive extinction parameter for incipient species
- pars[7] corresponds to *muibg*, the background incipient species extinction rate
- pars[8] corresponds to *alpha*1, the competition effect on extinction (competition strength)
- pars[9] corresponds to *alpha*1, the competition effect on trait evolution (competition strength)

	pars[10] corresponds to <i>sig2</i> , the variance (rate) of the Brownian motion
	pars[11] corresponds to <i>m</i> , the relative contribution of character displacement (competition) with respect to stochastic (brownian) evolution
root.value	the starting trait value
age.max	maximum time for the simulation (if the process doesn't go extinct)
step.size	size of each simulation step
bounds	lower and upper value for bounds in trait space
plot	logical indicating whether to plot the simulation
ylims	y axis (trait values) limits for the simulation plot
full.sim	logical indicating whether to return the full simulation (see details)

Details

It might be difficult to find parameter combinations that are sensitive. It is recommended to use the parameter settings of the examples as a starting point and from there modify them to understand the behaviour of the model. If trees produced are too big, simulation can become too slow to ever finish.

Value

returns a list with the following elements:

all contains the complete tree of the process (extant and extinct good and incipient lineages) and trait values for each tip in the tree

gsp_fossil contains the extant and extinct good species tree and trait values for each tip in the tree

gsp_extant contains the reconstructed (extant only) good species tree and trait values for each tip in the tree

If full.sim = TRUE, two additional elements are returned inside **all**:

note: both elements are used internally to keep track of the simulation and are dynamically updated, so returned elements only reflect the last state

lin_mat a matrix with information about the diversification process. Each row represents a new lineage in the process with the following elements: - Parental node, descendent node (0 if a tip), starting time, ending time, status at end (extinct(-2); incipient(-1); good(1)), speciation completion or extinction time; speciation completion time (NA if still incipient).

trait_mat a list with trait values for each lineage at each time step throughout the simulation. Each element is a vector composed of the following: Lineage number (same as row number in lin_mat), status (as in lin_mat), sister lineage number, trait values (NA if lineage didn't exist yet at that time step)

Author(s)

Leandro Aristide (leandroaristi@gmail.com)

References

Aristide, L., and Morlon, H. 2019. Understanding the effect of competition during evolutionary radiations: an integrated model of phenotypic and species diversification

Examples

```

lambda1 = 0.25
tau0 = 0.01
beta = 0.6
mu0 = 0.5
mubg = 0.01
mui0 = 0.8
muibg = 0.02
alpha1 = alpha2 = 0.04
sig2 = 0.5
m = 20

pars <- c(lambda1, tau0, beta, mu0, mubg, mui0, muibg, alpha1, alpha2, sig2, m)

if(test){

#1000 steps, unbounded
res <- sim_MCBD(pars, age.max=10, step.size=0.01)

#asymmetric bounds
res <- sim_MCBD(pars, age.max=10, step.size=0.01, bounds=c(-10,Inf))

#only deterministic component
pars <- c(lambda1, tau0, beta, mu0, mubg, mui0, muibg, alpha1, alpha2, sig2=0, m)
res <- sim_MCBD(pars, age.max=10)

plot(res$gsp_extant$tree)

}

```

sim_sgd

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

```
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

Manceau M., Lambert A., Morlon H. (2015) Phylogenies support out-of-equilibrium models of biodiversity Ecology Letters 18: 347-356

Examples

```
tau <- 10
b <- 1e6
d <- b-0.5
nu <- 0.6
tree <- sim_sgd(tau,b,d,nu)
plot(tree)
```

 sim_t_comp

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

```
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 Nuismer & 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 ($\max(\text{nodeHeights}(\text{phylo}))/\text{Nsegments}$) should be much smaller than the smallest branch ($\min(\text{phylo}\$edge.length)$).

Value

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

Author(s)

J Drury jonathan.p.drury@gmail.com

References

- Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020
- Nuismer, S. & Harmon, L. 2015. Predicting rates of interspecific interaction from phylogenetic trees. *Ecology Letters* 18:17-27.
- Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

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, ...)
```

Arguments

phylo	An object of class 'phylo' (see ape documentation)
param	A numeric vector of parameters for the user-defined climatic model. For the <i>EnvExp</i> and <i>EnvLin</i> , 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 σ^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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Science*, 114(16): 4183-4188.

See Also

[plot.fit_t.env](#), [likelihood_t.env](#)

Examples

```

if(test){
  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))
  fit

  # 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))
  fit2

  # When providing the environmental function:
  if(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
}
}

```

sim_t_env_ou

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

Description

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

Usage

```

sim_t_env_ou(phylo, param, env_data, model, step=0.01,
             plot=FALSE, sigma, alpha, theta0, ...)

```

Arguments

phylo	An object of class 'phylo' (see ape documentation)
param	A numeric vector of parameters for the user-defined climatic model. For the OU-environmental model, there is only one parameters (beta). If a model fit object of class 'fit_t_env.ou' is provided, the ML parameters are used to generate new datasets.
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 trajectory of the optimum "theta(t)" with time and the environmental variable (see details for default model). An user defined function of any functional form can be used (forward in time). This function has four 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), and the fourth is the theta_0 value. See the example below.
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.
sigma	The "sigma" parameter of the OU process.
alpha	The "alpha" parameter of the OU process.
theta0	The "theta" parameter at the root of the tree (t=0).
...	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

Clavel, J. & Morlon, H., 2017. Accelerated body size evolution during cold climatic periods in the Cenozoic. *Proceedings of the National Academy of Sciences*, 114(16): 4183-4188.

Troyer, E., Betancur-R, R., Hughes, L., Westneat, M., Carnevale, G., White W.T., Pogonoski, J.J., Tyler, J.C., Baldwin, C.C., Orti, G., Brinkworth, A., Clavel, J., Arcila, D., 2022. The impact of paleoclimatic changes on body size evolution in marine fishes. *Proceedings of the National Academy of Sciences*, 119 (29), e2122486119.

Goswami, A. & Clavel, J., 2024. Morphological evolution in a time of Phenomics. *EcoEvoRxiv*, <https://doi.org/10.32942/X22G7Q>

See Also

[plot.fit_t.env](#), [fit_t.env](#), [fit_t.env_ou](#), [plot.fit_t.env_ou](#)

Examples

```
if(test){

data(InfTemp)
set.seed(9999) # for reproducibility

# Let's start by simulating a trait under a climatic OU
beta = 0.6          # relationship to the climate curve
sim_theta = 4       # value of the optimum if the relationship to the climate
# curve is 0 (this corresponds to an 'intercept' in the linear relationship used below)
sim_sigma2 = 0.025 # variance of the scatter = sigma^2
sim_alpha = 0.36    # alpha value = strength of the OU; quite high here...
delta = 0.001       # time step used for the forward simulations => here its 1000y steps
tree <- pbtree(n=200, d=0.3) # simulate a bd tree with some extinct lineages
root_age = 60       # height of the root (almost all the Cenozoic here)
tree$edge.length <- root_age*tree$edge.length/max(nodeHeights(tree))
# here - for this contrived example - I scale the tree so that the root is at 60 Ma

# define a model - here we replicate the default model used in fit_t_env_ou
```



```

my_model <- function(t, env, param, theta0) theta0 + param[1]*env(t)

# simulate the traits
trait <- sim_t_env_ou(tree, sigma=sqrt(sim_sigma2), alpha=sim_alpha,
                    theta0=sim_theta, param=beta, model=my_model,
                    env_data=InfTemp, step=0.01, scale=TRUE, plot=TRUE)

## Fit the Environmental model (default)

result_fit <- fit_t_env_ou(phylo = tree, data = trait, env_data =InfTemp,
                        method = "Nelder-Mead", df=50, scale=TRUE)
plot(result_fit)

# We can also use the results from the previous fit to simulate a new dataset
trait2 <- sim_t_env_ou(tree, param=result_fit, step=0.001, plot=TRUE)

result_fit2 <- fit_t_env_ou(phylo = tree, data = trait2, env_data =InfTemp,
                          method = "Nelder-Mead", df=50, scale=TRUE)
result_fit2
}

```

sim_t_tworegime

Recursive simulation (root-to-tip) of two-regime models

Description

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

Usage

```

sim_t_tworegime(regime.map, pars, root.value, Nsegments=2500,
               model=c("MC", "DDexp", "DDlin", "EB"),
               verbose=TRUE, rnd=6)

```

Arguments

regime.map	a stochastic map of the two regimes stored as a simmap object output from make.simmap
pars	a vector containing the three parameters for the chosen model; all models require sig2, and additionally, the MC model requires S1 and S2, specifying the level of competition in regime 1 and 2, respectively (larger negative values correspond to higher levels of competition), the DDlin model requires b1 and b2, the DDexp model requires r1, 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, "DDlin" is the diversity-dependent linear model, "DDexp" is the diversity-dependent exponential model, and "EB" is the early burst model.
verbose	if TRUE, prints the identity of regimes corresponding to each parameter value
rnd	number of digits to round timings to (see round (see Details))

Details

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

Adjusting rnd may help if function crashes.

Value

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

Author(s)

J Drury jonathan.p.drury@gmail.com

References

- Drury, J., Clavel, J., Manceau, M., and Morlon, H. 2016. Estimating the effect of competition on trait evolution using maximum likelihood inference. *Systematic Biology* doi 10.1093/sysbio/syw020
- Nuismer, S. & Harmon, L. 2015. Predicting rates of interspecific interaction from phylogenetic trees. *Ecology Letters* 18:17-27.
- Weir, J. & Mursleen, S. 2012. Diversity-dependent cladogenesis and trait evolution in the adaptive radiation of the auks (Aves: Alcidae). *Evolution* 67:403-416.

See Also

[fit_t_comp](#)

Examples

```
data(Cetacea_clades)

# Simulate data under the matching competition model
MC_tworegime.data<-sim_t_tworegime(Cetacea_clades,pars=c(sig2=0.01,S1=-0.1,S2=-0.01),
  root.value=0,Nsegments=1000,model="MC")

# Simulate data under the diversity dependent linear model
```

```

DDlin_tworegime.data<-sim_t_tworegime(Cetacea_clades,pars=c(sig2=0.01,b1=-0.0001,b2=-0.000001),
  root.value=0,Nsegments=1000,model="DDlin")

# Simulate data under the diversity dependent linear model
DDexp_tworegime.data<-sim_t_tworegime(Cetacea_clades,pars=c(sig2=0.01,r1=-0.01,r2=-0.02),
  root.value=0,Nsegments=1000,model="DDexp")

# Simulate data under the diversity dependent linear model
EB.data_tworegime<-sim_t_tworegime(Cetacea_clades,pars=c(sig2=0.01,r1=-0.01,r2=-0.02),
  root.value=0,Nsegments=1000,model="EB")

```

 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, meth=c("standard"), zero_bound=F)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
meth	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)
zero_bound	if false, eigenvalues less than one are discarded

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](#), [BICcompare](#)

Examples

```
data(Cetacea)
spectR(Cetacea, meth="standard", zero_bound=FALSE)
```

spectR_t

Spectral density plot of phylogenetic trait data

Description

Computes the spectra of eigenvalues for the modified graph Laplacian of a phylogenetic tree with associated tip data, convolves the eigenvalues with a Gaussian kernel and plots the density profile of eigenvalues, and estimates the summary statistics of the profile.

Usage

```
spectR_t(phylo, dat, draw=F)
```

Arguments

phylo	an object of type 'phylo' (see ape documentation)
dat	a vector of trait data associated with the tips of the phylo object; tips and trait data should be aligned
draw	if true, the spectral density profile of the phylogenetic trait data is plotted

Value

a list with the following components:

eigenvalues	the vector of eigenvalues
splitter	the largest (or principal) eigenvalue of the spectral density profile
fragmenter	the skewness of the spectral density profile
tracer	the largest y-axis value of the spectral density profile

Author(s)

E Lewitus

References

Lewitus, E., Morlon, H. (2019) Characterizing and comparing phylogenetic trait data from their normalized Laplacian spectrum, bioRxiv doi: <https://doi.org/10.1101/654087>

Examples

```
tr<-rtree(10)
dat<-runif(10,1,2)
spectR_t(tr,dat,draw=TRUE)
```

taxo_cetacea	<i>Cetacean taxonomy</i>
--------------	--------------------------

Description

Taxonomy of Cetaceans

Usage

```
data(taxo_cetacea)
```

Details

This taxonomy lists all species of Cetaceans to properly calculate sampling fractions by clades. It corresponds to the phylogeny of Steeman et al. (2009).

Source

Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

References

- Steeman ME et al.(2009) Radiation of extant cetaceans driven by restructuring of the oceans *Syst Biol* 58:573-585
- Morlon, H., Parsons, T.L., Plotkin, J.B. (2011) Reconciling molecular phylogenies with the fossil record *Proc Nat Acad Sci* 108: 16327-16332
- Mazet, N., Morlon, H., Fabre, P., Condamine, F.L., (2023). Estimating clade-specific diversification rates and palaeodiversity dynamics from reconstructed phylogenies. *Methods in Ecology and in Evolution* 14, 2575–2591. <https://doi.org/10.1111/2041-210X.14195>

Examples

```
data(taxo_cetacea)
print(taxo_cetacea)
```

theta_estimator	<i>Compute Watterson genetic diversity (Theta estimator)</i>
-----------------	--

Description

This function computes the Theta estimator of genetic diversity (Watterson, 1975) while controlling for the presence of gaps in the alignment (Ferretti et al, 2012), frequent in barcoding datasets.

Usage

```
theta_estimator(sequences)
```

Arguments

sequences a matrix representing the nucleotidic alignment of all the sequences present in the phylogenetic tree.

Value

An estimate of genetic diversity.

Author(s)

Ana C. Afonso Silva & Benoît Perez-Lamarque

References

- Watterson GA , On the number of segregating sites in genetical models without recombination, 1975, *Theor. Popul. Biol.*
- Ferretti L, Raineri E, Ramos-Onsins S. 2012. Neutrality tests for sequences with missing data. *Genetics* 191: 1397–1401.
- Perez-Lamarque B, Öpik M, Maliet O, Silva A, Selosse M-A, Martos F, and Morlon H. 2022. Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont, *Molecular Ecology*, 31:3496–512.

See Also

[pi_estimator](#) [delineate_phylotypes](#)

Examples

```
data(woodmouse)

alignment <- as.character(woodmouse) # nucleotidic alignment

theta_estimator(alignment)
```

Index

* classes

PhenotypicACDC-class, [142](#)
PhenotypicADiag-class, [143](#)
PhenotypicBM-class, [144](#)
PhenotypicDD-class, [145](#)
PhenotypicGMM-class, [146](#)
PhenotypicModel-class, [148](#)
PhenotypicOU-class, [149](#)
PhenotypicPM-class, [150](#)

* datasets

Balaenopteridae, [13](#)
Calomys, [17](#)
Cetacea, [20](#)
Cetacea_clades, [21](#)
co2, [23](#)
co2_res, [23](#)
coccolithophore, [24](#)
d13c, [34](#)
foraminifera, [93](#)
greenalgae, [106](#)
InfTemp, [107](#)
landplant, [111](#)
mycorrhizal_network, [139](#)
ostracoda, [140](#)
Phocoenidae, [151](#)
Phyllostomidae, [153](#)
Phyllostomidae_genera, [154](#)
radiolaria, [186](#)
redalgae, [187](#)
sealevel, [189](#)
shifts_cetacea, [192](#)
silica, [193](#)
taxo_cetacea, [221](#)

* methods

fitTipData-methods, [43](#)
getDataLikelihood-methods, [98](#)
getTipDistribution-methods, [102](#)
modelSelection-methods, [138](#)
simulateTipData-methods, [202](#)

[,PhenotypicModel,ANY,ANY,ANY-method
(PhenotypicModel-class), [148](#)
[<- ,PhenotypicModel,ANY,ANY,ANY-method
(PhenotypicModel-class), [148](#)

add.gts, [7](#)
ancestral, [9](#), [90](#), [153](#)
Anolis.data, [10](#)
apply_prob_dtt, [11](#), [141](#)

Balaenopteridae, [13](#)
BGB.examples, [14](#)
BICompare, [15](#), [108](#), [165](#), [220](#)
build_network.BipartiteEvol, [16](#)

Calomys, [17](#)
Caprimulgidae, [18](#)
Caprimulgidae_ClaDS2, [19](#)
Cetacea, [20](#), [21](#)
Cetacea_clades, [21](#)
ClaDS0_example, [22](#)
co2, [23](#)
co2_res, [23](#)
coccolithophore, [24](#)
createClassObject, [25](#), [28](#)
CreateGeobyClassObject, [25](#), [26](#), [26](#), [79](#),
[119](#)
CreateGeoObject, [11](#), [28](#), [31](#), [74](#), [76](#), [119](#),
[120](#), [122](#), [123](#), [127](#), [128](#), [197–199](#)
CreateGeoObject_BioGeoBEARS, [14](#), [28](#), [30](#)
createModel, [32](#)
createModelCoevolution, [33](#)

d13c, [34](#)
define_species.BipartiteEvol, [35](#)
delineate_phylotypes, [37](#), [161](#), [223](#)
div.models, [39](#)
div.rates, [40](#)
fit_bd, [13](#), [43](#), [56](#), [68](#), [171](#), [172](#), [181](#), [185](#), [208](#)
fit_bd_backbone, [47](#)

- fit_bd_backbone_c, 50
- fit_bd_in_past, 54, 71
- fit_ClaDS, 19, 57, 58, 61, 99, 138, 167
- fit_ClaDS0, 22, 59, 60, 100, 166
- fit_coal_cst, 62, 65
- fit_coal_var, 63, 63
- fit_env, 45, 49, 53, 66, 71, 173, 208
- fit_env_in_past, 56, 69
- fit_sgd, 72
- fit_t_comp, 28–31, 74, 78, 79, 92, 120, 122, 123, 127, 128, 212, 218
- fit_t_comp_subgroup, 25–28, 76, 77
- fit_t_env, 10, 80, 125, 216
- fit_t_env_ou, 84, 131, 164, 216
- fit_t_pl, 10, 89, 103, 105, 153, 183
- fit_t_standard, 91
- fitTipData, 42
- fitTipData, PhenotypicModel-method (fitTipData-methods), 43
- fitTipData-methods, 43
- foraminifera, 93
- get.comb.shift, 94, 96, 178
- get.sampling.fractions, 95, 95, 191
- getDataLikelihood, 97
- getDataLikelihood, PhenotypicModel-method (getDataLikelihood-methods), 98
- getDataLikelihood-methods, 98
- getMAPS_ClaDS, 98, 100, 167
- getMAPS_ClaDS0, 19, 61, 99, 99, 166
- getTipDistribution, 101
- getTipDistribution, PhenotypicACDC-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicADiag-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicBM-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicDD-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicGMM-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicModel-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicOU-method (getTipDistribution-methods), 102
- getTipDistribution, PhenotypicPM-method (getTipDistribution-methods), 102
- getTipDistribution-methods, 102
- GIC, 10
- GIC.fit_pl.rpanda, 90, 103, 105, 153, 183
- gic_criterion, 10, 90, 103, 104, 153
- greenalgae, 106
- InfTemp, 107
- JSDt_cluster, 109
- JSDtree, 15, 107, 109, 220
- JSDtree_cluster, 108, 108
- landplant, 111
- likelihood_bd, 45, 49, 53, 68, 111
- likelihood_bd_backbone, 113
- likelihood_coal_cst, 63, 115
- likelihood_coal_var, 65, 116
- likelihood_sgd, 73, 118
- likelihood_subgroup_model, 79, 119
- likelihood_t_DD, 76, 120, 121, 123
- likelihood_t_DD_geog, 76, 122, 122
- likelihood_t_env, 83, 124, 129, 162, 214
- likelihood_t_MC, 76, 126, 128
- likelihood_t_MC_geog, 76, 127, 127
- lines.fit_t.env, 128, 162, 164
- lines.fit_t.env.ou, 130, 164
- make.simmmap, 25, 27–29, 74, 77, 91, 217
- make_gen.BipartiteEvol, 132
- mantel_test, 134, 137
- mantel_test_nbpartners, 135
- modelSelection, 137
- modelSelection, PhenotypicModel-method (modelSelection-methods), 138
- modelSelection-methods, 138
- MPhiFFT, 138
- mvgl, 90, 103
- mycorrhizal_network, 139
- optim, 75, 78, 92
- ostracoda, 140
- paleodiv, 140, 191
- par, 129, 130, 162, 163

- PhenotypicACDC-class, 142
- PhenotypicADiag-class, 143
- PhenotypicBM-class, 144
- PhenotypicDD-class, 145
- PhenotypicGMM-class, 146
- PhenotypicModel, 142, 144–147, 149, 151
- PhenotypicModel-class, 148
- PhenotypicOU-class, 149
- PhenotypicPM-class, 150
- Phocoenidae, 151
- phyl.pca_pl, 10, 90, 152, 183
- Phyllostomidae, 153, 154
- Phyllostomidae_genera, 154, 154
- phylosignal_network, 135, 137, 155, 177
- phylosignal_sub_network, 135, 137, 157, 158, 160, 177
- pi_estimator, 38, 160, 223
- plot.fit_t.env, 83, 129, 161, 214, 216
- plot.fit_t.env.ou, 87, 131, 163, 216
- plot_BICompare, 15, 165
- plot_ClaDS0_chains, 61, 100, 166, 167
- plot_ClaDS_chains, 19, 59, 99, 166, 167
- plot_ClaDS_phylo, 168, 205
- plot_div.BipartiteEvol, 169
- plot_dtt, 45, 49, 53, 56, 171, 181, 185
- plot_fit_bd, 45, 49, 53, 56, 172
- plot_fit_env, 68, 71, 173
- plot_net.BipartiteEvol, 174
- plot_phylo_comb, 177
- plot_phylosignal_sub_network, 160, 176
- plot_prob_dtt, 13, 180, 185
- plot_spectR, 182, 220
- Posdef, 183
- print,PhenotypicModel-method
(PhenotypicModel-class), 148
- prob_dtt, 13, 181, 184

- radiolaria, 186
- redalgae, 187
- remove.model, 188
- round, 25, 218
- RPANDA (RPANDA-package), 5
- RPANDA-package, 5

- sealevel, 189
- shift.estimates, 41, 95, 96, 141, 179, 188, 190, 191, 200
- shifts_cetacea, 192

- show,PhenotypicModel-method
(PhenotypicModel-class), 148
- silica, 193
- sim.BipartiteEvol, 16, 36, 132, 170, 175, 194
- sim.convergence.geo, 196
- sim.divergence.geo, 198
- sim_ClaDS, 168, 202
- sim_env_bd, 206
- sim_MCBD, 208
- sim_sgd, 210
- sim_t_comp, 76, 211
- sim_t_env, 213
- sim_t_env_ou, 87, 215
- sim_t_tworegime, 92, 217
- simul.comb.shift, 199
- simulateTipData, 200
- simulateTipData,PhenotypicModel-method
(simulateTipData-methods), 202
- simulateTipData-methods, 202
- spectR, 15, 108, 182, 219
- spectR_t, 110, 220
- spg, 78

- taxo_cetacea, 221
- theta_estimator, 38, 161, 222