Package: DAMOCLES (via r-universe)

September 5, 2024

Type Package

Title Dynamic Assembly Model of Colonization, Local Extinction and Speciation

Version 2.3

Date 2020-08-12

Depends R (>= 3.0)

Imports caper, ape, deSolve, matrixStats, expm, picante, Matrix, DDD (>= 3.4), stats, methods, Hmisc

Suggests testthat (>= 2.1.0)

Author Rampal S. Etienne & Alex L. Pigot

Maintainer Rampal S. Etienne <`r.s.etienne@rug.nl`>

License GPL-2

Description Simulates and computes (maximum) likelihood of a dynamical model of community assembly that takes into account phylogenetic history.

Encoding UTF-8

RoxygenNote 7.1.1

Repository https://rsetienne.r-universe.dev

RemoteUrl https://github.com/rsetienne/damocles

RemoteRef HEAD

RemoteSha 05fbf008d9d80f94d7c16c9fad5e3d07dccae90c

Contents

DAMOCLES_bootstrap *Phylogenetic community structure hypothesis test*

Description

This function computes the maximum likelihood estimates of colonisation and local extinction rate for a given phylogeny and presence-absence data under the DAMOCLES model. These rate estimates are used to simulate null communities under the DAMOCLES model. Standardized effect size of mean nearest taxon distance (mntd), mean phylogentic distance (mpd) and loglikelihood are calculated For comparison, standardised effect sizes are also calculated relative to a "Random-Draw" null model i.e. presence absence randomised across tips

Usage

```
DAMOCLES_bootstrap(
  phy = ape::rcoal(10),
 pa = matrix(c(phy$tip.label, sample(c(0, 1), ape::Ntip(phy), replace = T)), nrow =
    ape::Ntip(phy), ncol = 2),
  initparsopt = c(0.1, 0.1),
  idparsopt = 1:length(initparsopt),
  parsfix = NULL,
  idparsfix = NULL,
  pars2 = c(0.001, 1e-04, 1e-05, 1000),
  pchoice = 0,
  runs = 999,
  estimate_pars = FALSE,
  conf.int = 0.95
)
```

Arguments

`phy`	phylogeny in phylo format
`pa`	presence-absence table. The first column contains the labels of the species (corresponding to the tip labels in the phylogeny. The second column contains the presence (1) or absence (0) of species in the local community.
`initparsopt`	The initial values of the parameters that must be optimized
`idparsopt`	The ids of the parameters that must be optimized, e.g. 1:2 for extinction rate, and offset of immigration rate The ids are defined as follows: id == 1 corresponds to mu (extinction rate) id == 2 corresponds to gamma_0 (offset of immigration rate)
`parsfix`	The values of the parameters that should not be optimized. See idparsfix.

`idparsfix`	The ids of the parameters that should not be optimized, e.g. c(1) if mu should not be optimized, but only gamma_0. In that case idparsopt must be c(2). The default is to fix the parameters not specified in idparsopt.
`pars2`	Vector of settings: `pars2[1]` sets the relative tolerance in the parameters `pars2[2]` sets the relative tolerance in the function `pars2[3]` sets the absolute tolerance in the parameters `pars2[4]` sets the maximum number of iterations
`pchoice`	sets which p-value to optimize and with which root state to simulate (default pchoice = 0) pchoice == 0 correspond to optimizing sum of p_0f + p_1f, and simulating with an equal number of root states being 0 or 1 pchoice == 1 correspond to optimizing p_0f, and simulating with root state being 0 pchoice == 2 correspond to optimizing p_1f, and simulating with root state being 1
`runs`	the number null communities to generate.
`estimate_pars`	Whether to estimate parameters on the simulated datasets (default = FALSE).
`conf.int`	The width of the conifdence intervals calculated on bootstrapped parameter estimates

Details

The output is a list of two dataframes. The first dataframe, summary_table, contains the summary results. The second dataframe, null_community_data, contains decsriptive statistics for each null community.

Value

`summary_table`	`mu` gives the maximum likelihood estimate of mu and confidence intervals in brackets if estimate_pars = TRUE `gamma_0` gives the maximum likelihood estimate of gamma_0 and confidence intervals in brackets if bootstrap=TRUE `loglik` gives the maximum loglikelihood `df` gives the number of estimated parameters, i.e. degrees of feedom `conv` gives a message on convergence of optimization; conv = 0 means convergence `n.obs` gives the number of species locally present in the observed community `mntd.obs` gives the MNTD of the observed community `mpd.obs` gives the MPD of the observed community `runs` gives the number of null communities simulated `mntd.mean.RD` mean of MNTD from null communities generated by a "Random Draw" model `mntd.sd.RD` standard deviation of MNTD from null communities generated by a "Random Draw" model `mntd.obs.z.RD` standardized effect size of MNTD compared to null communities generated by a "Random Draw" model (= -1*(mntd.obs - mntd.mean.RD)/ mntd.sd.RD) `mntd.obs.rank.RD` rank of observed MNTD compared to null communities generated by a "Random Draw" model `mntd.obs.q.RD`

quantile of observed MNTD vs. null communities (= mntd.obs.rank.RD /runs + 1) `mpd.mean.RD` mean of MPD from null communities generated by a "Random Draw" model `mpd.sd.RD` standard deviation of MPD from null communities generated by a "Random Draw" model `mpd.obs.z.RD` standardized effect size of MPD compared to null communities generated by a "Random Draw" model (= -1*(mpd.obs - mpd.mean.RD)/ mpd.sd.RD) `mpd.obs.rank.RD` rank of observed MPD compared to null communities generated by a "Random Draw" model `mpd.obs.q.RD` quantile of observed MPD vs. null communities (= mpd.obs.rank.RD /runs + 1) `n.mean.DAMOCLES` mean number of species locally present in the null communities generated by DAMOCLES `mntd.mean.DAMOCLES` mean of MNTD from null communities generated by DAMOCLES `mntd.sd.DAMOCLES` standard deviation of MNTD from null communities generated by DAMOCLES `mntd.obs.z.DAMOCLES` standardized effect size of MNTD compared to null communities generated by DAMOCLES (= -1*(mntd.obs - mntd.mean.DAMOCLES)/ mntd.sd.DAMOCLES) `mntd.obs.rank.DAMOCLES` rank of observed MNTD compared to null communities generated by DAMOCLES `mntd.obs.q.DAMOCLES` quantile of observed MNTD vs. null communities (= mntd.obs.rank.DAMOCLES /runs + 1) `mpd.mean.DAMOCLES` mean of MPD from null communities generated by DAMOCLES `mpd.sd.DAMOCLES` standard deviation of MPD from null communities generated by DAMOCLES `mpd.obs.z.DAMOCLES` standardized effect size of MPD compared to null communities generated by DAMOCLES (= -1*(mpd.obs - mpd.mean.DAMOCLES)/ mpd.sd.DAMOCLES) `mpd.obs.rank.DAMOCLES` rank of observed MPD compared to null communities generated by DAMOCLES `mpd.obs.q.DAMOCLES` quantile of observed MPD vs. null communities (= mpd.obs.rank.DAMOCLES /runs + 1) `loglik.mean.DAMOCLES` mean of loglikelihoods from null communities generated by DAMOCLES `loglik.sd.DAMOCLES` standard deviation of loglikelihoods from null communities generated by DAMOCLES `loglik.obs.z.DAMOCLES` standardized effect size of loglikelihood compared to null communities generated by DAMOCLES (= -1*(loglik.obs - loglik.mean.DAMOCLES)/ loglik.sd.DAMOCLES) `loglik.obs.rank.DAMOCLES` rank of observed loglikelihood compared to null communities generated by DAMOCLES `loglik.obs.q.DAMOCLES` quantile of observed loglikelihoods vs. null communities (= loglik.obs.rank.DAMOCLES /runs + 1)

`null_community_data`

`run` gives the simulation run `root.state.print` gives the state of the ancestral species in the local community assumed in the simulation, i.e. present (1) or absent (0) `n` gives the number of species locally present in the observed community `n.RD` gives the number of species locally present in the null community generated by a "Random Draw" model `mntd.RD` gives the MNTD of the null community generated by a "Random Draw" model `mpd.RD` gives the MPD of the null community generated by a "Random Draw" model `n.DAMOCLES` gives the number of species locally present in the null community generated by DAMOCLES `mntd.DAMOCLES` gives the MNTD of the null community generated by DAMOCLES `mpd.DAMOCLES` gives the MPD of the null community generated by DAMOCLES `loglik.DAMOCLES` gives the maximum loglikelihood for the null community generated by DAMOCLES `mu.DAMOCLES` gives the maximum likelihood estimate of mu for the null community generated by DAMOCLES `gamma_0.DAMOCLES` gives the maximum likelihood estimate of gamma_0 for the null community generated by DAMOCLES

Author(s)

Rampal S. Etienne

References

Pigot, A.L. & R.S. Etienne (2015). A new dynamic null model for phylogenetic community structure. Ecology Letters 18: 153-163.

See Also

`DAMOCLES_ML` `DAMOCLES_sim`

`DAMOCLES_loglik` *Likelihood for DAMOCLES model*

Description

Computes likelihood for the presence-absence data of species in a local community for a given phylogeny of species in the region.

Usage

```
DAMOCLES_loglik(
  phy,
  pa,
  pars,
  pchoice = 0,
  edgeTList = NULL,
  methode = "analytical",
  model = 0,
  Mlist = NULL,
  verbose = FALSE
)
```

Arguments

`phy`	phylogeny in phylo format
`pa`	presence-absence table with the first column the species labels and the second column the presence (1) or absence (0) of the species
`pars`	Vector of model parameters: `pars[1]` corresponds to mu (extinction rate in local community) `pars[2]` corresponds to gamma_0 in formula gamma(t) = gamma_0/(1 + gamma_1 * t) where gamma(t) is immigration rate into local community) `pars[3]` corresponds to gamma_1 in formula gamma(t) = gamma_0/(1 + gamma_1 * t) where gamma(t) is immigration rate into local community)

`pchoice`	sets the p-value to optimize: pchoice == 0 corresponds to the sum of p_0f + p_1f pchoice == 1 corresponds to p_0f pchoice == 2 corresponds to p_1f
`edgeTList`	list of edge lengths that need to be succesively pruned; if not specified, it will computed using compute_edgeTList
`methode`	method used to solve the ODE. Either 'analytical' for the analytical solution, 'Matrix' for matrix exponentiation using package Matrix or 'expm' using package 'expm' or any of the numerical solvers, used in deSolve.
`model`	model used. Default is 0 (standard null model). Other options are 1 (binary traits) 2 (trinary environmental trait) or 3 (diversity-dependent colonization - beta version)
`Mlist`	list of M matrices that can be specified when methode = 'analytical'. If set at NULL (default) and methode = 'analytical', Mlist will be computed.
`verbose`	Whether intermediate output should be printed. Default is FALSE.

Value

The loglikelihood

Author(s)

Rampal S. Etienne

References

Pigot, A.L. & R.S. Etienne (2015). A new dynamic null model for phylogenetic community structure. Ecology Letters 18: 153-163.

See Also

`DAMOCLES_ML` `DAMOCLES_sim`

Examples

```
#TEST IT WORKS
library(ape)
phy = ape::rcoal(100)
pars = c(0.5,0.1,0.1)
pa = rbinom(100,c(0,1),0.5)
pa = matrix(c(phy$tip.label,pa),nrow = length(phy$tip.label),ncol = 2)

# - without a root edge
loglik = DAMOCLES_loglik(phy,pa,pars)
loglik

# - with a root edge
phy$root.edge = 2
```

```
loglik = DAMOCLES_loglik(phy,pa,pars)
loglik
```

DAMOCLES_ML *Maximization of the loglikelihood under the DAMOCLES model*

Description

This function computes the maximum likelihood estimates of the parameters of the DAMOCLES model for a given phylogeny and presence-absence data. It also outputs the corresponding loglikelihood that can be used in model comparisons.

Usage

```
DAMOCLES_ML(
  phy,
  pa,
  initparsopt,
  idparsopt = 1:length(initparsopt),
  parsfix = NULL,
  idparsfix = NULL,
  idparsequal = NULL,
  pars2 = c(0.001, 1e-04, 1e-05, 1000),
  optimmethod = "subplex",
  pchoice = 0,
  edgeTList = NULL,
  methode = "analytical",
  model = 0,
  verbose = FALSE
)
```

Arguments

`phy`	phylogeny in phylo format
`pa`	presence-absence table. The first column contains the labels of the species (corresponding to the tip labels in the phylogeny. The second column contains the presence (1) or absence (0) of species in the local community.
`initparsopt`	The initial values of the parameters that must be optimized
`idparsopt`	The ids of the parameters that must be optimized, e.g. 1:2 for extinction rate, and offset of immigration rate The ids are defined as follows: id == 1 corresponds to mu (extinction rate) id == 2 corresponds to gamma_0 (offset of immigration rate) id == 3 corresponds to gamma_1 (parameter controlling decline in immigration rate with time)

`parsfix`	The values of the parameters that should not be optimized. See idparsfix.
`idparsfix`	The ids of the parameters that should not be optimized, e.g. c(1,3) if mu and gamma_1 should not be optimized, but only gamma_0. In that case idparsopt must be c(2). The default is to fix all parameters not specified in idparsopt.
`idparsequal`	The ids of the parameters that should be set equal to the first parameter of the same type.
`pars2`	Vector of settings: `pars2[1]` sets the relative tolerance in the parameters `pars2[2]` sets the relative tolerance in the function `pars2[3]` sets the absolute tolerance in the parameters `pars2[4]` sets the maximum number of iterations
`optimmethod`	Method used in optimization of the likelihood. Current default is 'subplex'. Alternative is 'simplex' (default of previous version)
`pchoice`	sets the p-value to optimize: pchoice == 0 corresponds to the sum of p_0f + p_1f pchoice == 1 corresponds to p_0f pchoice == 2 corresponds to p_1f
`edgeTList`	list of edge lengths that need to be succesively pruned; if not specified, it will computed using compute_edgeTList
`methode`	method used to solve the ODE. Either 'analytical' for the analytical solution, 'Matrix' for matrix exponentiation using package Matrix or 'expm' using package 'expm' or any of the numerical solvers, used in deSolve.
`model`	model used. Default is 0 (standard null model). Other options are 1 (binary traits) 2 (trinary environmental trait) or 3 (diversity-dependent colonization - beta version)
`verbose`	Whether intermediate output should be printed. Default is FALSE.

Details

The output is a dataframe containing estimated parameters and maximum loglikelihood.

Value

`mu`	gives the maximum likelihood estimate of mu
`gamma_0`	gives the maximum likelihood estimate of gamma_0
`gamma_1`	gives the maximum likelihood estimate of gamma_1
`loglik`	gives the maximum loglikelihood
`df`	gives the number of estimated parameters, i.e. degrees of feedom
`conv`	gives a message on convergence of optimization; conv = 0 means convergence

Author(s)

Rampal S. Etienne

References

Pigot, A.L. & R.S. Etienne (2015). A new dynamic null model for phylogenetic community structure. Ecology Letters 18: 153-163.

See Also

`DAMOCLES_loglik` `DAMOCLES_sim`

`DAMOCLES_sim` *Simulating DAMOCLES*

Description

Simulates DAMOCLES

Usage

```
DAMOCLES_sim(
  phy,
  gamma_0,
  gamma_td,
  mu,
  sigma,
  psiBranch,
  psiTrait,
  z,
  phi,
  traitOpt,
  br0,
  br_td,
  nTdim,
  root.state,
  root.trait.state,
  plotit = FALSE,
  keepExtinct = FALSE
)
```

Arguments

`phy`	phylogeny in phylo format
`gamma_0`	initial per lineage rate of immigration (gamma)
`gamma_td`	time dependency in gamma

`mu`	per lineage rate of local extinction
`sigma`	probability of local (i.e. in-situ) speciation
`psiBranch`	phylogenetic distance at which gamma is half gamma_0
`psiTrait`	trait distance at which gamma is half gamma_0
`z`	shape of increase in gamma with increasing trait or phylogenetic distance
`phi`	rate of decline in gamma with distance from trait optima
`traitOpt`	trait value at which gamma = gamma_0
`br0`	Brownian rate parameter
`br_td`	rate of temporal decline in Brownian rate parameter
`nTdim`	number of independent trait dimensions
`root.state`	geographic state of ancestor i.e. present (1) or absent(0)
`root.trait.state`	trait value of ancestor
`plotit`	whether to plot the phylogeny and timing of immigration/local extinction events
`keepExtinct`	whether to retain data for extinct lineages

Value

A list of two tables. The first table contains the following columns: The first column contains the vector of tip labels in the phylogeny The last column contains the presence (1) or absence (0) of the species The second table has dimensions d x N where d is the number of trait dimensions and N is the number of species. It contains the trait values.

Author(s)

Alex L. Pigot

References

Pigot, A.L. & R.S. Etienne (2015). A new dynamic null model for phylogenetic community structure. Ecology Letters 18: 153-163.

See Also

`DAMOCLES_ML` `DAMOCLES_loglik`

Examples

```
#create random phylogeny
library(ape)
phy = ape::rcoal(10)

#run DAMOCLES
out = DAMOCLES_sim(
  phy,
  gamma_0 = 1.5,
```

```
    gamma_td =0,
    mu = 0,
    sigma = 0,
    psiBranch = 0,
    psiTrait = 0,
    z = 10,
    phi = 0,
    traitOpt = 1,
    br0 = 0.1,
    br_td = -0.1,
    nTdim = 2,
    root.state = 1,
    root.trait.state = 0,
    plotit = FALSE,
    keepExtinct = FALSE
    )

  #the output consists of a list
  patable = out[[1]] # the first element is the presence absence table
  traits = out[[2]] # this is a matrix of traits values

  #show presence/absence on the tree
  patable$col = rep("black",dim(patable)[1])
  patable$col[which(patable$state == 1)] = "red"
  plot(phy,tip.col = patable$col)
```

`NWPrimates_data` *Dated phylogenetic tree of the New World Primates in nexus format and presence-absence matrix for species in Manu*

Description

A list with two elements.
. phy is a dated molecular phylogeny for 94 species of New World Primates extracted from the maximum likelihood tree (AUTOsoft dated) of Springer et al. (2012). 1 time unit = 100 million years.
pa is the presence-absence matrix of NW Primates in Manu from Solari et al. (2006). The first column indicate the species tip labels and the second column indicates presence (1) and absence (0).

Format

A list with two elements. The first element (phy) is the primate phylogeny in nexus format. The second element (pa) is the presence-absence matrix with 94 rows and 2 columns.

Source

Solari, S., Pacheco, V., Luna, L., Velazco, P.M. & Patterson, B.D. 2006 Mammals of the manu biosphere reserve. Fieldiana Zoology 110, 13-22.
Springer, M.S., Meredith, R.W., Gatesy, J., Emerling, C.A., Park, J., Rabosky, D.L., Stadler, T., Steiner, C., Ryder, O.A., Janecka, J.E., et al. 2012 Macroevolutionary dynamics and historical biogeography of primate diversification inferred from a species supermatrix. Plos One 7. (doi:ARTN e49521 DOI 10.1371/journal.pone.0049521).

See Also

`DAMOCLES_sim`, `DAMOCLES_ML`, `DAMOCLES_loglik`

Index