Package: comdynamite (via r-universe)

November 7, 2026

Title Process-Based Metacommunity Simulations with Phylogenetic Structure

Version 0.5.2

Description Simulates metacommunity dynamics with the process-based model of Thompson et al. (2020) <doi:10.1111/ele.13568>: density-independent growth that depends on the match between each species' environmental optimum and the local environment, Beverton-Holt competition, Poisson demographic stochasticity, and emigration and immigration among sites. Landscapes can be spatially implicit, with uniform dispersal, or explicit grids with an exponential dispersal kernel and a spatially autocorrelated environment, and the environment can change over time. Species traits (environmental optima, niche breadths and maximum growth rates) can carry phylogenetic signal or evolve under an Ornstein-Uhlenbeck process with one or several optima, and interspecific competition can be structured by relatedness. Rule-based assembly without population dynamics (limiting similarity, habitat filtering and neutral assembly) follows Kraft et al. (2007) <doi:10.1086/519400>, and static metacommunities with unimodal species responses follow Peres-Neto et al. (2012) <doi:10.1890/11-0494.1>. The model was implemented from the published equations.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports ape, graphics, grDevices, parallel, purrr, stats

Suggests adespatial, knitr, picante, rmarkdown, testthat (>= 3.0.0), treeshapesim, vegan

Additional_repositories https://diogoprov.r-universe.dev

Remotes diogoprov/treeshapesim

Config/testthat/edition 3

VignetteBuilder knitr

URL https://provetelab.org/comdynamite/,
https://github.com/diogoprov/comdynamite

BugReports https://github.com/diogoprov/comdynamite/issues

X-schema.org-keywords metacommunity, community-ecology, simulation, species-sorting, mass-effects, dispersal, competition, phylogenetics, community-phylogenetics, process-based-model

Config/roxygen2/version 8.1.1

LazyData true

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

Date/Publication 2026-11-07 00:57:43 UTC

RemoteUrl https://github.com/diogoprov/comdynamite

RemoteRef main

RemoteSha 65b28cc23681a56d217b3ae1983a26870f734b2a

Contents

assemble_community . . . 3
biogeography_matrix . . . 4
competition_matrix . . . 6
daphniids . . . 7
dispersal_kernel . . . 8
env_over_time . . . 8
growth_rates . . . 9
landscape_grid . . . 10
landscape_implicit . . . 11
logistic_gradients . . . 11
mc_dynamics . . . 12
node_contrasts . . . 14
node_test . . . 15
paint_regimes . . . 18
pcc_test . . . 18
phylo_beta . . . 20
phylo_beta_test . . . 21
phylo_eigenvectors . . . 22
phylo_vector . . . 23
plot_node_heatmap . . . 26
plot_node_test . . . 27
plot_trait_partition . . . 28
simulate_logistic . . . 30
simulate_metacommunity . . . 31
simulate_static . . . 33
species_traits . . . 35
torus_permutations . . . 36

trait_bm 37
trait_partition 37
traits_ou 40

Index **42**

`assemble_community` *Assemble a local community from a regional pool by rules*

Description

Static community assembly, without population dynamics, following Kraft et al. (2007). Species are removed from the pool one at a time until the community has `size` species (or until the optional distance end point is reached):

Usage

```
assemble_community(
  traits,
  size,
  rule = c("limiting_similarity", "filtering", "neutral"),
  optimum = "ancestral",
  min_distance = NULL
)
```

Arguments

`traits`	Numeric vector (one trait) or matrix (one column per trait), with species names as names or row names. For example, the output of `traits_ou()`.
`size`	Number of species in the local community.
`rule`	Assembly rule.
`optimum`	For `rule = "filtering"`: the niche optimum, either `"ancestral"` (the root state, 0 by default in `traits_ou()`), `"derived"` (the maximum of each trait in the pool, one extreme corner of trait space), or a numeric vector with one value per trait.
`min_distance`	For `rule = "limiting_similarity"`: optional end point. Removal stops as soon as no two remaining species are closer than this (a limit to similarity), even if more than `size` species remain.

Details

- `"limiting_similarity"`: the GAUSE algorithm of Colwell & Winkler (1984), direct competition followed by diffuse competition. The pair of species closest in trait space is found and, of the two, the species with the smaller sum of distances to all remaining species (the greater "diffuse similarity") is removed. The process is repeated on the remaining species. Kraft et al. (2007) used this algorithm for their limiting similarity model.

- `"filtering"`: habitat filtering as in Kraft et al. (2007). The species farthest from a niche optimum in trait space are removed first.
- `"neutral"`: species are drawn at random, without regard to traits.

Distances are Euclidean, so several traits can be used at once (one column each). Ties are broken at random.

Value

A list with `community` (names of the species kept) and `removed` (names of the species removed, in order of removal).

References

Colwell, R. K., & Winkler, D. W. (1984). A null model for null models in biogeography. In D. R. Strong, D. Simberloff, L. G. Abele, & A. B. Thistle (Eds.), *Ecological Communities: Conceptual Issues and the Evidence* (pp. 344–359). Princeton University Press. doi:10.1515/9781400857081.344

Kraft, N. J. B., Cornwell, W. K., Webb, C. O., & Ackerly, D. D. (2007). Trait evolution, community assembly, and the phylogenetic structure of ecological communities. *The American Naturalist*, 170, 271–283. doi:10.1086/519400

Examples

```
set.seed(1)
tr <- treeshapesim::sim_tree_shape(32)
x <- traits_ou(tr, n_traits = 2)
assemble_community(x, size = 16, rule = "limiting_similarity")$community
assemble_community(x, size = 16, rule = "filtering", optimum = "derived")$community
```

`biogeography_matrix` *Matrix of biogeographic events*

Description

Codes the events of `node_test()` from the area of each site, for example the refugium, basin or island where it lies. Each event gets 1 for sites on one side, -1 for sites on the other and 0 for sites on neither.

Usage

```
biogeography_matrix(
  area,
  contrast = "others",
  outside = c("zero", "other"),
  min_sites = 5
)
```

Arguments

area	Area of each site, with NA for sites outside all areas, or a matrix or data frame with one row per site and one column per area, with 1 or TRUE for the sites of each area. Polygons can give it with sf::st_join() or sf::st_intersects().
contrast	Named list of events, each a list with the areas of the two sides, or "others" for one event per area.
outside	For contrast = "others", sites outside all areas are left out of each event with "zero" or count as the other side with "other".
min_sites	A warning is given for events with fewer sites on one side. This is a rule of thumb, not a value from the literature.

Details

Events should come from a hypothesis stated in advance. Leibold et al. (2010) contrasted areas on the two sides of historical barriers, and Leibold & Chase (2018, chapter 9) a published biogeographic hypothesis. A named list in contrast gives this design, and nested events are written as nested groups of areas. contrast = "others" compares each area with the rest, which does not need that hypothesis, but asks a different question and is better seen as exploratory.

A site on both sides of an event gets 0.

The sites of an area are often neighbours, so their number overstates the information in an event. See site_permutations in node_test().

Value

Matrix with one row per site and one column per event, with values of 1, -1 and 0.

References

Leibold, M. A., & Chase, J. M. (2018). *Metacommunity Ecology*. Monographs in Population Biology 59. Princeton University Press. doi:10.2307/j.ctt1wf4d24

Leibold, M. A., Economo, E. P., & Peres-Neto, P. (2010). Metacommunity phylogenetics: separating the roles of environmental filters and historical biogeography. *Ecology Letters*, 13, 1290–1299. doi:10.1111/j.14610248.2010.01523.x

Examples

```
area <- rep(c("north", "centre", "south", NA), c(8, 6, 6, 10))

# a barrier between the north and the rest, and a later one between centre and south
events <- list(barrier = list("north", c("centre", "south")),
               river = list("centre", "south"))
colSums(biogeography_matrix(area, events) == 1)

# each area against the others
head(biogeography_matrix(area, "others"))
```

`competition_matrix` *Competition matrix*

Description

Intraspecific coefficients are 1 and interspecific coefficients lie in [0, `inter_max`), all multiplied by `scale`. With the default `scale` of 0.05, as in Thompson et al. (2020), a species alone at its optimum with `r_max = 5` reaches an equilibrium of `(5 - 1) / 0.05 = 80` individuals, and `inter_max` is the strength of interspecific relative to intraspecific competition. The matrix is symmetric.

Usage

```
competition_matrix(
  species,
  type = c("none", "random", "phylo"),
  inter_max = 0.1,
  scale = 0.05
)
```

Arguments

`species`	A `phylo` tree, or the number of species (not for `type = "phylo"`).
`type`	Type of interspecific competition.
`inter_max`	Maximum interspecific coefficient, relative to intraspecific competition.
`scale`	Multiplier of all coefficients.

Details

- `"none"`: no interspecific competition.
- `"random"`: interspecific coefficients drawn from U(0, `inter_max`).
- `"phylo"`: the same uniform values, assigned to pairs of species by rank of phylogenetic similarity (1 - d / max(d), with d the patristic distance), so that close relatives compete more strongly while the distribution of coefficients stays the same as with `"random"`.

Value

Square matrix: element [j, k] is the effect of species k on species j.

Examples

```
set.seed(1)
tr <- treeshapesim::sim_tree_shape(5)
round(competition_matrix(tr, "phylo", inter_max = 0.5), 3)
```

daphniids *Daphniids of the northeastern United States*

Description

Incidence of 13 daphniids in lakes of the EMAP Northeastern Lakes Survey of 1991 to 1994 (first visit to each lake), rebuilt to follow Leibold et al. (2010). The tables are public data of the US Environmental Protection Agency, and the script is `data-raw/daphniids.R`.

Usage

```
daphniids
```

Format

List with `comm`, a matrix of lakes by species; `env`, a matrix with colour, conductivity, dissolved organic carbon, total nitrogen, pH, total phosphorus, mean depth, elevation and the ratio of nitrogen to phosphorus; `biogeography`, a matrix of contrasts with values of 1, -1 and 0; and `tree`, a `phylo` object.

Details

The article gives 327 lakes and this rebuilt set has 314, of which 239 have some species. The tree is the topology of Figure 3a of the article, without branch lengths. The survey lists *D. longiremis* and not the *D. longispina* of the figure, which we took to be the same taxon. The article does not give the area of each lake, so rivers in `biogeography` are approximated by longitude and these contrasts differ from the original ones: S1, east and west of the Hudson; S2, Adirondacks and other lakes to the west of the Hudson; S3, east and west of the Connecticut River, east of the Hudson; S4, Maine and other lakes east of the Hudson; S5, New Hampshire and lakes to its west but east of the Hudson; S6, New Hampshire and lakes in Rhode Island and Massachusetts.

Source

https://archive.epa.gov/emap/archive-emap/web/html/nelakes.html

References

Leibold, M. A., Economo, E. P., & Peres-Neto, P. (2010). Metacommunity phylogenetics: separating the roles of environmental filters and historical biogeography. *Ecology Letters*, 13, 1290–1299. doi:10.1111/j.14610248.2010.01523.x

dispersal_kernel *Exponential dispersal kernel*

Description

The probability that an emigrant from site s reaches site t is proportional to exp(-d / `scale`), where d is the distance between the sites. Emigrants do not stay at their own site.

Usage

```
dispersal_kernel(dist, scale)
```

Arguments

`dist`	Square matrix of distances among sites.
`scale`	Scale of the kernel, in the units of `dist`.

Value

Row-stochastic matrix with a zero diagonal: row s gives the probabilities that an emigrant from site s reaches each site.

Examples

```
d <- as.matrix(dist(cbind(1:5, 0)))
round(dispersal_kernel(d, scale = 1), 2)
```

env_over_time *Environment that changes over time*

Description

Adds to the environment of each site a deviation that follows an independent first-order autoregressive process with lag-one autocorrelation `rho` and stationary standard deviation `sd`, starting from its stationary distribution. This is a simple generator of temporal autocorrelation, not the one used by Thompson et al. (2020).

Usage

```
env_over_time(env, n_steps, rho = 0.9, sd = 5)
```

Arguments

`env`	Environment of each site, e.g. the `env` of an `mc_landscape`.
`n_steps`	Number of time steps.
`rho`	Lag-one autocorrelation of the deviations, in [0, 1).
`sd`	Stationary standard deviation of the deviations.

Value

Matrix with one row per site and one column per time step, for the `env_time` argument of `simulate_metacommunity()`.

Examples

```
set.seed(1)
e <- env_over_time(c(20, 50, 80), n_steps = 100, rho = 0.95, sd = 5)
matplot(t(e), type = "l", lty = 1, xlab = "step", ylab = "environment")
```

`growth_rates` *Growth rates of each species at each site*

Description

Density-independent growth rate of species j at site i, `r_max[j] * exp(-(E[i] - optimum[j])^2 / (2 * sigma[j]^2))`: a Gaussian response whose standard deviation is the niche breadth `sigma`. This `sigma` equals `sqrt(2)` times the niche breadth parameter of Thompson et al. (2020).

Usage

```
growth_rates(env, traits)
```

Arguments

`env`	Environment of each site.
`traits`	Data frame from `species_traits()`, or any data frame with columns `optimum`, `sigma` and `r_max`.

Value

Numeric matrix with one row per site and one column per species: the density-independent growth rate of each species at each site.

Examples

```
set.seed(1)
g <- growth_rates(seq(0, 100, 10), species_traits(3))
round(g, 2)
```

`landscape_grid` *Spatially explicit landscape on a square grid*

Description

Sites on a `side` x `side` grid, optionally wrapped as a torus so that no site lies on an edge. The environment is spatially autocorrelated: independent normal deviates are smoothed by a Gaussian kernel with a bandwidth of `env_bandwidth` cells, and then given a uniform marginal distribution on `env_range` by ranks. Emigrants reach the other sites with probabilities from `dispersal_kernel()`.

Usage

```
landscape_grid(
  side = 10,
  env_bandwidth = 2,
  disp_scale = 2,
  torus = TRUE,
  env_range = c(0, 100)
)
```

Arguments

`side`	Number of cells per side.
`env_bandwidth`	Bandwidth (in cells) of the Gaussian kernel that smooths the environment. Larger values give smoother gradients.
`disp_scale`	Scale (in cells) of the exponential dispersal kernel.
`torus`	Wrap the grid as a torus?
`env_range`	Range of the uniform marginal distribution of the environment.

Value

An `mc_landscape` object: a list with `env`, `coords` (data frame with columns `x` and `y`), `dist` (distances among sites, toroidal if `torus = TRUE`) and `dispersal` (row-stochastic matrix with a zero diagonal).

See Also

`torus_permutations()` for site permutations that keep the spatial structure of a torus.

Examples

```
set.seed(1)
ls <- landscape_grid(side = 8)
image(matrix(ls$env, 8, 8), asp = 1)
```

landscape_implicit *Spatially implicit landscape*

Description

Sites without coordinates, with environmental values drawn from a uniform distribution. Dispersal among sites is uniform: emigrants can reach any site, so sites are exchangeable and permuting them is a valid null model.

Usage

```
landscape_implicit(n_sites, env_range = c(0, 100))
```

Arguments

n_sites	Number of sites.
env_range	Range of the uniform distribution of the environment.

Value

An mc_landscape object: a list with env (one value per site), and coords, dist and dispersal set to NULL.

See Also

landscape_grid() for a spatially explicit landscape.

Examples

```
set.seed(1)
landscape_implicit(20)
```

logistic_gradients *Gradients of the seven scenarios of Peres-Neto et al. (2012)*

Description

Two gradients on a transect, as in Appendix C of Peres-Neto et al. (2012). Unless stated, x is evenly spaced on [-15, 15] and y is x in random order, so that the two are independent.

Usage

```
logistic_gradients(scenario, n_sites = 31)
```

Arguments

scenario	Number of the scenario, from 1 to 7.
n_sites	Number of sites.

Details

1. The two gradients are equally important and independent.
2. They are equally important and correlated: y is x plus a N(0, 1) error.
3. They are independent, and y is less important: data come from y, and y_test is y with unit variance plus a N(0, 1) error. On the scale of y this error would have no effect, and with unit variance the results agree with Table 5.
4. Neither is important: data come from scenario 1, and x_test and y_test are N(0, 1) deviates, to estimate the type I error.
5. y has a broad spatial scale, the first Moran's eigenvector map (MEM), and x is in random order along the transect.
6. y has an intermediate spatial scale, the third MEM, and x is as in scenario 5.
7. x is the first MEM and y the third.

MEMs are rescaled to [-15, 15].

Value

Data frame with one row per site: x and y, the gradients that generate the data, and x_test and y_test, the ones to analyse.

References

Peres-Neto, P. R., Leibold, M. A., & Dray, S. (2012). Assessing the effects of spatial contingency and environmental filtering on metacommunity phylogenetics. *Ecology*, 93, S14–S30. doi:10.1890/110494.1

Examples

```
set.seed(1)
head(logistic_gradients(6))
```

---

mc_dynamics *Metacommunity dynamics from growth rates*

Description

The engine of simulate_metacommunity(), for users who want to supply their own growth rates (e.g. responses to several environmental variables).

Usage

```
mc_dynamics(
  r,
  alpha,
  dispersal,
  n_steps = 1000,
  seed_steps = seq(10, 100, 10),
  seed_mean = 0.5,
  disp_matrix = NULL,
  record_steps = NULL,
  sites = NULL,
  species = NULL,
  extirpation = 0
)
```

Arguments

r	Sites x species matrix of density-independent growth rates, or a function of the time step that returns such a matrix.
alpha	Competition matrix from competition_matrix(), with species in the same order as the rows of traits.
dispersal	Emigration probability, one value or one per species.
n_steps	Number of time steps.
seed_steps	Steps at which individuals are added during colonisation.
seed_mean	Mean number of individuals of each species added to each site at each seeding step (and at the start).
disp_matrix	Row-stochastic dispersal matrix (sites x sites), or NULL for uniform dispersal.
record_steps	Optional steps at which abundances are recorded.
sites, species	Optional names of sites and species.
extirpation	Probability that a population is lost at each time step, one value or one per species.

Value

List with abundance and history, as in simulate_metacommunity().

Examples

```
set.seed(1)
# one species alone converges to (r - 1) / alpha = (5 - 1) / 0.05 = 80
out <- mc_dynamics(matrix(5, 20, 1), matrix(0.05), dispersal = 0, n_steps = 200)
mean(out$abundance)
```

`node_contrasts` *Node contrasts of a phylogeny*

Description

Codes each node of a tree as a contrast between its two descendant clades, as in Leibold et al. (2010). One clade starts at -1 and the other at 1, and the value is halved at each node on the way to the tips. Species outside the node get zero.

Usage

```
node_contrasts(tree, cherries = TRUE, branch_lengths = FALSE)
```

Arguments

`tree`	A `phylo` object with dichotomous nodes.
`cherries`	Keep nodes that join two tips? The article tests them (its Figure 3), but the function `CreateNodeContrasts` of the authors leaves them out.
`branch_lengths`	Use the branch lengths of `tree`?

Details

With `branch_lengths = TRUE`, values follow section III of the supplement of Leibold et al. (2010): each branch between a species and the node counts its length divided by the number of species below it, and the sum is divided by the total length of that side of the node. Species that share history get less weight, and each side still sums to 1. The supplement proposes this coding for trees in which all species are equally distant from the root.

Value

Matrix with one row per tip and one column per node, named by the node numbers of `tree`.

Note

Adapted from the MATLAB function `CreateNodeContrasts` in the supplement of Leibold et al. (2010).

References

Leibold, M. A., Economo, E. P., & Peres-Neto, P. (2010). Metacommunity phylogenetics: separating the roles of environmental filters and historical biogeography. *Ecology Letters*, 13, 1290–1299. doi:10.1111/j.14610248.2010.01523.x

Examples

```
node_contrasts(ape::read.tree(text = "((a,b),(c,(d,e)));"))

# Figure S2 of the supplement
tree <- ape::read.tree(text = "((((A:0.2,B:0.2):0.65,C:0.85):0.15,D:1):0.4,(E:0.5,F:0.5):0.9);")
round(node_contrasts(tree, branch_lengths = TRUE), 3)
```

`node_test` *Test clades against the environment, node by node*

Description

Implements the regression of Leibold et al. (2010). For each node, sites without species of that node are removed, the contrast is standardised with species weights and summed within sites, and the result is regressed on the predictors standardised with site weights, using site richness as weights. Each node shows whether the two clades that split there differ in their response to the predictors.

Usage

```
node_test(
  comm,
  nodes,
  env,
  biogeography = NULL,
  n_perm = 9999,
  cores = 1,
  site_permutations = NULL
)
```

Arguments

`comm`	Matrix with one row per site and one column per species. Species are matched to the rows of `nodes` by name, in any order, and those in only one of the two are dropped with a warning.
`nodes`	Matrix of contrasts with one row per species, from `node_contrasts()`.
`env`	Predictor, or matrix or data frame of predictors, with one row per site.
`biogeography`	Optional matrix of biogeographic events with one row per site and values of 1, -1 and 0.
`n_perm`	Number of permutations of sites, of species and of both. With `site_permutations`, only of species.
`cores`	Number of cores, with nodes run in parallel. Not available on Windows, where one core is used. For reproducible p-values with more than one core, call `RNGkind("L'Ecuyer-CMRG")` before `set.seed()`. Results still depend on the number of cores.
`site_permutations`	Optional matrix of site indices with one permutation per row, for example from `torus_permutations()`, to use instead of free permutations of sites.

Details

The test statistic is the explained over the residual sum of squares. It is tested by permuting sites, and by replacing the species of the node with a random set of the same size from all species. Following Peres-Neto et al. (2012) and ter Braak et al. (2012), the p-value to report is the larger of the two tests. With many nodes these p-values need a correction, and `p_adjusted` gives the false discovery rate that the article uses. This correction needs many permutations, as the smallest p-value is 1 / (`n_perm` + 1), so the default is the 9999 of the article.

The article permutes sites and species at once, which is `p_joint`. This tests the null hypothesis that neither the environment nor the phylogeny matters, so it is not a test of their association. In simulations where the environment acts but species responses have no phylogenetic signal, `p_joint` rejected 76% of the nodes at the 5% level, and `p_max` 4.8%.

With few species at a node, permutations of species have little power, so `p_max` is conservative: with the daphniids of the article, of 13 species, no predictor passes it, whereas sites alone recover most of the environmental results of its Figure 3 (see the article on validation).

Each predictor gets a semipartial correlation, the square root of the loss of R2 when it is removed, with the sign of its slope when it is alone in the model.

Biogeographic events follow the article: a column of `biogeography` has 1 for sites in one area, -1 for sites in the other and 0 for sites in neither. Each event is tested only in its sites, controlling for `env`. Other contrasts are allowed, for example a column without zeros, which compares one area with all the others. Events with few sites in one area give unstable values. `biogeography_matrix()` builds the matrix from the area of each site.

Free permutations treat sites as exchangeable. When sites of an area are neighbours and the community has spatial structure of its own, they reject too often: 45% of tests at the 5% level in a simulation with areas placed at random, against 5% with `torus_permutations()` (see the article on analysing clades). `site_permutations` accepts a restricted null model. For events, each row moves the areas across the sites, which keep their community and predictors.

R2, adjusted R2 and the statistic equal those of the MATLAB function `MetacommunityPhylogenetics` of the authors, checked in Octave. Four parts of that function differ from this one:

- In the original, the species permutations shuffle only the first columns of the community matrix, as many as there are species in the node, so they do not draw from all species. Here they do.
- The original keeps the total of the observed data after permuting species. Here it is recalculated.
- The semipartial values of the original use the raw predictors, without an intercept, and multiply by the absolute slope. The article uses standardised predictors and divides by it, to keep only the sign, which is what we do.
- The original compares the permuted statistics as they are. The one of each predictor divides a proportion by a sum of squares, so it depends on the scale of the response, and with one predictor it does not rank draws as the global one does. Neither takes into account that draws of species occupy different numbers of sites. Here draws are ranked on F statistics. `statistic` is still the one of the original, and p-values of sites alone are the same, but those that use species differ.

As in the original, R2 sums squares without site weights, although the fit uses them, so it is not the R2 of a weighted regression, and the adjusted R2 uses the degrees of freedom of an unweighted model.

Nodes for which a predictor has no variance among the remaining sites, with too few sites, or with a clade without individuals get NA. The original stops with an error.

Sites are matched by name when comm, env and biogeography all have row names, and otherwise by position.

Value

List of class mc_node_test with nodes, a data frame with one row per node with node, n_species, n_sites, r2, r2_adjusted, statistic, p_site, p_taxa, p_max, p_joint and p_adjusted; semipartial, a matrix of nodes by predictors; p_partial, with the larger of the two p-values for each of them; p_partial_site, with the p-values of sites alone; p_partial_joint, with those of the permutation of the article; biogeography, a list with the same four matrices for the events, or NULL; and permutations, the numbers of permutations and whether those of sites were free or restricted.

Note

Adapted from the MATLAB function MetacommunityPhylogenetics in the supplement of Leibold et al. (2010), with the differences listed in Details.

References

Leibold, M. A., Economo, E. P., & Peres-Neto, P. (2010). Metacommunity phylogenetics: separating the roles of environmental filters and historical biogeography. *Ecology Letters*, 13, 1290–1299. doi:10.1111/j.14610248.2010.01523.x

Peres-Neto, P. R., Leibold, M. A., & Dray, S. (2012). Assessing the effects of spatial contingency and environmental filtering on metacommunity phylogenetics. *Ecology*, 93, S14–S30. doi:10.1890/110494.1

ter Braak, C. J. F., Cormont, A., & Dray, S. (2012). Improved testing of species traits-environment relationships in the fourth-corner problem. *Ecology*, 93, 1525–1526.

See Also

biogeography_matrix() codes the events. plot_node_test() and plot_node_heatmap() show the results. pcc_test() and phylo_beta_test() test the whole tree at once.

Examples

```
# daphniids of Leibold et al. (2010)
nodes <- node_contrasts(daphniids$tree)
set.seed(1)
out <- node_test(daphniids$comm, nodes, daphniids$env, n_perm = 49)
out
round(out$semipartial, 2)
plot_node_test(out, daphniids$tree, test = "site")
```

paint_regimes *Assign selective regimes to clades*

Description

Paints the branches of a tree with selective regimes for `traits_ou()`: every branch starts in regime `"0"`, and the branches descending from each node in `nodes` take a new regime (`"1"`, `"2"`, ...). Nodes are painted in order, so a later node nested in an earlier one overrides it.

Usage

```
paint_regimes(tree, nodes, stem = FALSE)
```

Arguments

`tree`	A rooted `phylo` tree.
`nodes`	Internal node numbers whose descendant branches get a new regime.
`stem`	Should the branch leading to each node also take the new regime (the shift happens at the start of the stem rather than at the node)?

Value

Character vector with the regime of each branch, in the order of `tree$edge` after reordering it "cladewise" (the order `traits_ou()` uses).

Examples

```
tr <- ape::read.tree(text = "((a:1,b:1):1,(c:1,d:1):1);")
paint_regimes(tr, nodes = 6)
```

pcc_test *Test phylogenetic community composition against the environment*

Description

Implements the phylogenetic community composition (PCC) framework of Peres-Neto & Kembel (2015). Phylogenetic eigenvectors are standardised with species weights and averaged within sites, and each resulting variable is regressed on the predictors with site weights. The test statistic is the variance of the adjusted R2 across eigenvectors. It is tested by permuting sites, which breaks the link between composition and environment, and by permuting species across the tips, which breaks the link between composition and phylogeny. The association requires both, so the p-value to report is the larger of the two.

Usage

```
pcc_test(
  comm,
  tree,
  env,
  n_perm = 999,
  squared = FALSE,
  site_permutations = NULL
)
```

Arguments

comm	Matrix with one row per site and one column per species. Species are matched to the tips of tree by name, in any order, and those in only one of the two are dropped with a warning. Every site needs at least one individual.
tree	A phylo object.
env	Predictor, or matrix or data frame of predictors, with one row per site.
n_perm	Number of permutations of sites and of species.
squared	Passed to phylo_eigenvectors().
site_permutations	Optional matrix of site indices with one permutation per row, for example from torus_permutations(), to use instead of free permutations of sites.

Details

The authors used presences and absences. Abundances are accepted, with species and site totals as weights. Species without individuals leave the analysis.

Value

List of class mc_pcc with r2, the mean of the positive adjusted R2 values, which is the value the authors recommend reporting; statistic, the variance of the adjusted R2 across eigenvectors; p_site, p_taxa and p_max; r2_adjusted, with one value per eigenvector; and permutations.

References

Peres-Neto, P. R., & Kembel, S. W. (2015). Phylogenetic gradient analysis: environmental drivers of phylogenetic variation across ecological communities. *Plant Ecology*, 216, 709–724. doi:10.1007/s1125801404050

Examples

```
set.seed(1)
tree <- treeshapesim::sim_tree_shape(20)
traits <- species_traits(tree, lambda = 1, optima_range = c(-1, 101),
                         rmax_range = c(0, 30))
env <- runif(50, 0, 100)
comm <- simulate_static(traits, env, incidence = TRUE)$abundance
```

```
keep <- rowSums(comm) > 0
pcc_test(comm[keep, ], tree, env[keep], n_perm = 99)
```

`phylo_beta` *Phylogenetic beta diversity between communities*

Description

Mean phylogenetic distance between the species of each pair of communities (COMDIST), the metric used by Peres-Neto & Kembel (2015). Values in `comm` are the weights of the species, so incidence data give the unweighted mean and abundances give the abundance-weighted mean.

Usage

```
phylo_beta(comm, tree)
```

Arguments

`comm`	Matrix with one row per site and one column per species, named as the tips of `tree`. Every site needs at least one individual.
`tree`	A `phylo` object with the same species as `comm`.

Details

Two communities with the same species have a distance equal to their mean pairwise distance, not zero, whereas the distance of a community to itself is set to zero. Peres-Neto & Kembel (2015) describe the matrix as fully Euclidean, but this does not always hold. If sites a and b both have species x and y, site c has only x and site d has only y, then a and b lie halfway between c and d and should coincide, yet their distance is not zero. The pseudo-F of `phylo_beta_test()` does not require a Euclidean matrix (McArdle & Anderson 2001).

Value

A `dist` object.

References

McArdle, B. H., & Anderson, M. J. (2001). Fitting multivariate models to community data: a comment on distance-based redundancy analysis. *Ecology*, 82, 290–297. doi:10.1890/00129658(2001)082[0290:FMMTCD]2.0.C

Peres-Neto, P. R., & Kembel, S. W. (2015). Phylogenetic gradient analysis: environmental drivers of phylogenetic variation across ecological communities. *Plant Ecology*, 216, 709–724. doi:10.1007/s1125801404050

Examples

```
set.seed(1)
tree <- treeshapesim::sim_tree_shape(5)
comm <- matrix(rbinom(20, 1, 0.7) + diag(1, 4, 5), 4,
               dimnames = list(NULL, tree$tip.label))
phylo_beta(comm > 0, tree)
```

---

phylo_beta_test *Test phylogenetic beta diversity against the environment*

Description

Implements the phylogenetic beta-diversity framework of Peres-Neto & Kembel (2015), a distance-based redundancy analysis (Legendre & Anderson 1999; McArdle & Anderson 2001) of `phylo_beta()` on the predictors. The test statistic is the pseudo-F. It is tested by permuting sites, which breaks the link between composition and environment, and by permuting species across the tips, which breaks the link between composition and phylogeny. The association requires both, so the p-value to report is the larger of the two.

Usage

```
phylo_beta_test(comm, tree, env, n_perm = 999, site_permutations = NULL)
```

Arguments

`comm`	Matrix with one row per site and one column per species, named as the tips of `tree`. Every site needs at least one individual.
`tree`	A `phylo` object with the same species as `comm`.
`env`	Predictor, or matrix or data frame of predictors, with one row per site.
`n_perm`	Number of permutations of sites and of species.
`site_permutations`	Optional matrix of site indices with one permutation per row, for example from `torus_permutations()`, to use instead of free permutations of sites.

Details

The equations printed in the article omit the 1 in the adjusted R2, the transpose in the hat matrix and the centring of the matrix that is regressed. Here the predictors and the matrix are centred, and the residual degrees of freedom are `n - p - 1`, with `n - 1` for the total, instead of the printed `n - p` and `n`. The pseudo-F and R2 therefore equal those of a distance-based redundancy analysis in vegan. The printed degrees of freedom would multiply the pseudo-F by a constant, which does not change the permutation tests.

The authors used presences and absences. Abundances are accepted, as weights in `phylo_beta()`.

Value

List of class `mc_phylo_beta` with `r2`, the adjusted R2; `statistic`, the pseudo-F; `p_site`, `p_taxa` and `p_max`; `scores`, the sites on the principal coordinates of the predicted matrix, which the authors suggest using to order sites in a heatmap of the community matrix; and `permutations`.

References

Legendre, P., & Anderson, M. J. (1999). Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. *Ecological Monographs*, 69, 1–24. doi:10.1890/00129615(1999)069[0001:DBRATM]2.0.CO;2

McArdle, B. H., & Anderson, M. J. (2001). Fitting multivariate models to community data: a comment on distance-based redundancy analysis. *Ecology*, 82, 290–297. doi:10.1890/00129658(2001)082[0290:FMMTCD]2.0.C

Peres-Neto, P. R., & Kembel, S. W. (2015). Phylogenetic gradient analysis: environmental drivers of phylogenetic variation across ecological communities. *Plant Ecology*, 216, 709–724. doi:10.1007/s1125801404050

See Also

`pcc_test()`, the other framework of the same article.

Examples

```
set.seed(1)
tree <- treeshapesim::sim_tree_shape(20)
traits <- species_traits(tree, lambda = 1, optima_range = c(-1, 101),
                         rmax_range = c(0, 30))
env <- runif(50, 0, 100)
comm <- simulate_static(traits, env, incidence = TRUE)$abundance
keep <- rowSums(comm) > 0
phylo_beta_test(comm[keep, ], tree, env[keep], n_perm = 99)
```

---

phylo_eigenvectors *Phylogenetic eigenvectors*

Description

Principal coordinates of the patristic distances among species, the first step of the phylogenetic community composition of Peres-Neto & Kembel (2015).

Usage

```
phylo_eigenvectors(tree, squared = FALSE)
```

Arguments

tree	A phylo object.
squared	Use -PD^2 / 2, as in a classic principal coordinate analysis, instead of -PD / 2, the form printed by Peres-Neto & Kembel (2015)?

Value

Matrix with one row per tip and one column per positive eigenvalue.

References

Peres-Neto, P. R., & Kembel, S. W. (2015). Phylogenetic gradient analysis: environmental drivers of phylogenetic variation across ecological communities. *Plant Ecology*, 216, 709–724. doi:10.1007/s1125801404050

Examples

```
set.seed(1)
dim(phylo_eigenvectors(treeshapesim::sim_tree_shape(10)))
```

---

phylo_vector *Represent a phylogeny by one vector related to a predictor*

Description

The phylogenetic representation of Peres-Neto et al. (2012). Eigenvectors of the phylogenetic distances are extracted with species weights. A global test relates all of them to the predictor. Only if it is significant, eigenvectors enter in the order of their correlations until the first that is not significant, and they are summed with their correlations as weights (pev). Brownian vectors are then added to pev, and the sum with the largest correlation is kept (pev_brown).

Usage

```
phylo_vector(
  comm,
  tree,
  predictor,
  n_perm = 999,
  alpha = 0.05,
  n_sim = 1000,
  site_permutations = NULL
)
```

Arguments

comm	Matrix with one row per site and one column per species. Every site needs at least one individual.
tree	A phylo object.
predictor	Numeric vector with the value of each site. The authors estimate one vector for each predictor.
n_perm	Number of permutations of sites and of species.
alpha	Significance level of the global test and of the selection.
n_sim	Number of Brownian vectors.
site_permutations	Optional matrix of site indices with one permutation per row, for example from torus_permutations(), to use instead of free permutations.

Details

The correlations are fourth-corner correlations, and the p-value to report is the larger of the tests that permute sites and species (ter Braak et al. 2012). For a test between a trait and the environment, see ade4::fourthcorner(). The two vectors are chosen to fit the predictor, so a further test of them is not valid on its own. They are meant for trait_partition() after a significant global test.

Species are matched to the tips of tree by name, in any order, and those in only one of the two are dropped with a warning. Species without individuals have no weight and are left out.

Value

List of class mc_phylo_vector with correlation, the correlations of pev and pev_brown; p_site, p_taxa and p_max of the global test; selected, the columns of eigenvectors retained; pev and pev_brown, named vectors or NULL when the global test is not significant or no eigenvector is retained; eigenvectors; eigenvalues; eigenvector_correlation; and permutations. Check pev before combining gradients with cbind(), which drops a NULL without warning.

Article and original code

The functions TraitEnvCor, ModelSelectionStochastic and PCoAWeigthed and the script ScriptDemo of the supplement differ from the article and its appendices in the points below. Here we follow the text, except in the addition of Brownian vectors and where noted.

- Weights of the eigenvectors: the text uses the correlations and the code their absolute values, so eigenvectors with opposite signs cancel.
- Brownian vectors: the text multiplies each one by pev and the code adds them. We follow the code, as the product of a random vector with mean zero would not keep the structure of pev.
- Number of Brownian vectors: 1000 in the text and 100 in the code.
- Distances: the text uses the square root of the path length, and the script one minus the covariance.
- Centring: Q in the text only centres if the weights sum to one, and the script passes the sums of the species. PCoAWeigthed also does not divide the coordinates by the square root of the weights. DoubleCenterWeigth, which the script does not call, does both and agrees with this function.

- Global test: the text regresses each eigenvector on all predictors, and the code accepts one. With one predictor both are the same statistic. We also take one predictor.
- Forward selection: the text tests each eigenvector given the remaining predictors, and the code tests its simple correlation. We follow the code.
- Significance level: Appendix B uses 0.05, and the code uses alpha / 2 for every permutation strategy, although Appendix A gives alpha / 2 only for the stacked test. Here the larger p-value is compared with `alpha`.
- p-value of the global test with row permutations: the code divides by twice the number of permutations, which halves it.
- No significant result: the code returns random normal vectors, which can then enter later analyses. Here `pev` and `pev_brown` are `NULL`.
- Choice of vector: the article recommends the one with the highest R2, and the script always takes the Brownian vector. Here both are returned.
- Combination of permutations: Appendix A stacks the two null distributions, adds one to the count and uses alpha / 2, and `TraitEnvCor` adds two. The stacked p-value is close to the mean of the two p-values, so any test that it rejects at alpha / 2 is also rejected by the larger p-value at alpha, which we use.
- Permuted variables: Appendix A recalculates the statistic, and `TraitEnvCor` keeps the observed mean and standard deviation, so a permuted variable no longer has weighted mean zero. Here it is standardised again, as in the code of the partition.
- The code runs `n_perm` - 1 permutations in each null distribution.
- The supplement calls `eigen`, which it does not include, and the example 2 of `ScriptDemo` removes rows of `PhyloDistance` before creating it.

Note

Adapted from the MATLAB functions `TraitEnvCor`, `ModelSelectionStochastic` and `PCoAWeigthed` in the supplement of Peres-Neto et al. (2012).

References

Peres-Neto, P. R., Leibold, M. A., & Dray, S. (2012). Assessing the effects of spatial contingency and environmental filtering on metacommunity phylogenetics. *Ecology*, 93, S14–S30. doi:10.1890/110494.1

ter Braak, C. J. F., Cormont, A., & Dray, S. (2012). Improved testing of species traits–environment relationships in the fourth-corner problem. *Ecology*, 93, 1525–1526. doi:10.1890/120126.1

Examples

```
set.seed(1)
tree <- treeshapesim::sim_tree_shape(20)
traits <- species_traits(tree, lambda = 1, optima_range = c(0, 100),
                         rmax_range = c(0, 30))
env <- runif(50, 0, 100)
comm <- simulate_static(traits, env, incidence = TRUE)$abundance
keep <- rowSums(comm) > 0
phylo_vector(comm[keep, ], tree, env[keep], n_perm = 99, n_sim = 100)
```

`plot_node_heatmap` *Plot the semipartial correlations of a node test*

Description

Draws a table of nodes by variables with the semipartial correlations of `node_test()`, as in Table S1 of Leibold et al. (2010). Bold values with an asterisk remain significant after the false discovery rate, as in `plot_node_test()`; other bold values are significant only as isolated tests. Blank cells could not be calculated.

Usage

```
plot_node_heatmap(
  x,
  test = c("max", "site", "joint"),
  alpha = 0.05,
  col = c("#0072B2", "#D55E00"),
  cex = 0.8,
  ...
)
```

Arguments

`x`	Result of `node_test()`.
`test`	Permutation test to show: `"max"` for the larger p-value of sites and species, `"site"` for sites alone, or `"joint"` for the permutation of the article. See `node_test()` for what each one tests.
`alpha`	Significance level.
`col`	Colours of negative and positive values.
`cex`	Size of the text.
`...`	Other arguments of `graphics::image()`.

Details

Rows are the node numbers of the tree. To find them, call `ape::nodelabels()` after `plot_node_test()`.

Value

Invisibly, the matrix of semipartial correlations, with predictors followed by biogeographic events.

References

Leibold, M. A., Economo, E. P., & Peres-Neto, P. (2010). Metacommunity phylogenetics: separating the roles of environmental filters and historical biogeography. *Ecology Letters*, 13, 1290–1299. doi:10.1111/j.14610248.2010.01523.x

Examples

```
nodes <- node_contrasts(daphniids$tree)
set.seed(1)
out <- node_test(daphniids$comm, nodes, daphniids$env, daphniids$biogeography,
                 n_perm = 49)
plot_node_heatmap(out, test = "site")
```

`plot_node_test` *Plot the results of a node test on the phylogeny*

Description

Draws the tree with the associations found by `node_test()` at each node, as in Figure 3 of Leibold et al. (2010). Blue circles are predictors and red triangles are biogeographic events, each with the names of the variables. Solid symbols remain significant after the false discovery rate, and open symbols are significant only as isolated tests.

Usage

```
plot_node_test(
  x,
  tree,
  test = c("max", "site", "joint"),
  type = c("phylogram", "fan"),
  alpha = 0.05,
  col = c("#0072B2", "#D55E00"),
  cex = 0.8,
  legend = TRUE,
  ...
)
```

Arguments

`x`	Result of `node_test()`.
`tree`	The `phylo` object that gave the contrasts of `node_contrasts()`.
`test`	Permutation test to show: `"max"` for the larger p-value of sites and species, `"site"` for sites alone, or `"joint"` for the permutation of the article. See `node_test()` for what each one tests.
`type`	`"phylogram"`, as in the article, or `"fan"`, for very large trees, with the names of the variables along the radii so that they stay clear of the species.
`alpha`	Significance level.
`col`	Colours of predictors and biogeographic events.
`cex`	Size of the text.
`legend`	Add a legend?
`...`	Other arguments of `ape::plot.phylo()`.

Details

The article does not say which tests form a family for the correction. Here, the correction of Benjamini & Hochberg (1995) uses all nodes and variables of each set, so predictors and events are corrected separately.

As the smallest p-value is 1 / (`n_perm` + 1), solid symbols need many permutations. With few species, `test = "max"` may leave no symbols at all (see `node_test()`).

Value

Invisibly, a data frame with one row per symbol, with `node`, `set`, `variable`, `p`, `p_adjusted` and `corrected`, which is `TRUE` for solid symbols.

References

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57, 289–300. doi:10.1111/j.25176161.1995.tb02031.x

Leibold, M. A., Economo, E. P., & Peres-Neto, P. (2010). Metacommunity phylogenetics: separating the roles of environmental filters and historical biogeography. *Ecology Letters*, 13, 1290–1299. doi:10.1111/j.14610248.2010.01523.x

Examples

```
nodes <- node_contrasts(daphniids$tree)
set.seed(1)
out <- node_test(daphniids$comm, nodes, daphniids$env, daphniids$biogeography,
                 n_perm = 49)
plot_node_test(out, daphniids$tree, test = "site")

# solid symbols need more permutations
out <- node_test(daphniids$comm, nodes, daphniids$env, daphniids$biogeography,
                 n_perm = 999)
plot_node_test(out, daphniids$tree, test = "site")
```

---

plot_trait_partition *Plot the fractions of a trait partition*

Description

Draws the total variation as a line divided into the fractions of `trait_partition()`, with one bar for the environment and one for space, as in Figure 1 of Desdevises et al. (2003). The two bars overlap in `b`, the variation they share. `a` is explained by the environment alone, `c` by space alone, and `d` is unexplained. Dark segments are significant, and `b` has no test, so it is grey.

Usage

```
plot_trait_partition(
  x,
  value = c("r2_adjusted", "r2", "r2_component"),
  test = c("max", "site", "taxa"),
  alpha = 0.05,
  col = "#0072B2",
  labels = c("Environment", "Space"),
  cex = 0.9,
  ...
)
```

Arguments

x	Result of trait_partition().
value	Column of the fractions to show: "r2_adjusted", "r2", or "r2_component" for the share of the mean or variance component.
test	Permutation test: "max" for the larger p-value of sites and species, "site" or "taxa".
alpha	Significance level.
col	Colour of significant segments.
labels	Names of the two sets of predictors.
cex	Size of the text.
...	Other arguments of graphics::plot.window().

Details

Negative fractions are shown as zero, following Legendre (2008). The result of trait_partition() keeps them, because all fractions sum to one only with them.

Value

Invisibly, the rows of x$fractions for a, b, c and d, with shown, the values drawn.

References

Desdevises, Y., Legendre, P., Azouzi, L., & Morand, S. (2003). Quantifying phylogenetically structured environmental variation. *Evolution*, 57, 2647–2652. doi:10.1111/j.00143820.2003.tb01508.x

Legendre, P. (2008). Studying beta diversity: ecological variation partitioning by multiple regression and canonical analysis. *Journal of Plant Ecology*, 1, 3–8. doi:10.1093/jpe/rtm001

Examples

```
set.seed(1)
trait <- stats::setNames(seq(0, 100, 10), letters[1:11])
env <- runif(40, 0, 100)
space <- cbind(runif(40), runif(40))
```

```
comm <- outer(env, trait, function(e, t) exp(-(e - t)^2 / 800))
colnames(comm) <- names(trait)
out <- trait_partition(comm, trait, env, space, n_perm = 49)
plot_trait_partition(out)
```

`simulate_logistic` *Simulate presences and absences from logistic responses*

Description

The two-gradient simulation of Appendix C of Peres-Neto et al. (2012), after Ives & Helmus (2011). The probability that species `j` occurs at site `t` along gradient `x` is `plogis(a[j] + b[j] * x[t])`, where `a` is a normal deviate and the slope `b` evolves on the tree. With several gradients, each with its own `a` and `b`, the probabilities multiply. A value drawn for each site from `reduction` then reduces them, which adds noise.

Usage

```
simulate_logistic(
  tree,
  gradients,
  intercept_mean = 0,
  intercept_sd = 0.5,
  slope_sd = 1,
  lambda = 1,
  reduction = c(0.5, 1),
  draw = c("uniform", "normal")
)
```

Arguments

`tree`	A `phylo` tree with branch lengths. Appendix C uses a fully balanced tree with 32 species.
`gradients`	Value of the gradient at each site, or a list or data frame with one gradient per element, such as `logistic_gradients()`. Gradients are centred.
`intercept_mean, intercept_sd`	Mean and standard deviation of `a`. Appendix C does not give these values, and the defaults are those of the code of Ives & Helmus (2011).
`slope_sd`	Standard deviation of the slopes `b` among the species of the tree, as in the code of Ives & Helmus (2011). Appendix C does not give this value.
`lambda`	Pagel's lambda of the slopes.
`reduction`	Range of the uniform distribution of the site reduction.
`draw`	Distribution of the value compared with each probability.

Details

Appendix C says that a species is present when a normally generated value is smaller than its probability. Taken literally, a species with a probability near zero is present half of the time, so the default is a uniform value (a Bernoulli draw), and `draw = "normal"` follows the text.

Value

List of class `mc_sim` with `abundance`, a 0/1 matrix with one row per site and one column per species; `probability`, the matrix of probabilities; `env`, the gradients; and `traits`, a list with the matrices `intercept` and `slope`, with one row per species and one column per gradient.

References

Ives, A. R., & Helmus, M. R. (2011). Generalized linear mixed models for phylogenetic analyses of community structure. *Ecological Monographs*, 81, 511–525. doi:10.1890/101264.1

Peres-Neto, P. R., Leibold, M. A., & Dray, S. (2012). Assessing the effects of spatial contingency and environmental filtering on metacommunity phylogenetics. *Ecology*, 93, S14–S30. doi:10.1890/110494.1

Examples

```
set.seed(1)
tr <- ape::compute.brlen(ape::stree(32, "balanced"), method = "Grafen")
simulate_logistic(tr, logistic_gradients(1)[c("x", "y")])
```

`simulate_metacommunity`

Simulate a metacommunity

Description

Runs the process-based model of Thompson et al. (2020) on a landscape. At each time step, abundances are drawn as `N' ~ Poisson(r N / (1 + N alpha'))`, where `r` holds the growth rates of each species at each site (`growth_rates()`) and `alpha` the competition coefficients (`competition_matrix()`). Each individual then emigrates with probability `dispersal`, and emigrants are redistributed among sites:

Usage

```
simulate_metacommunity(
  landscape,
  traits,
  alpha,
  dispersal = 0.01,
  n_steps = 1000,
  env_time = NULL,
  seed_steps = seq(10, 100, 10),
```

```
  seed_mean = 0.5,
  record_steps = NULL,
  extirpation = 0
)
```

Arguments

landscape	An mc_landscape object.
traits	Species traits from species_traits().
alpha	Competition matrix from competition_matrix(), with species in the same order as the rows of traits.
dispersal	Emigration probability, one value or one per species.
n_steps	Number of time steps.
env_time	Optional matrix (sites x n_steps) with the environment at each step, e.g. from env_over_time(). If NULL, the environment of the landscape is constant.
seed_steps	Steps at which individuals are added during colonisation.
seed_mean	Mean number of individuals of each species added to each site at each seeding step (and at the start).
record_steps	Optional steps at which abundances are recorded.
extirpation	Probability that a population is lost at each time step, one value or one per species.

Details

- spatially implicit landscapes (landscape_implicit()): the emigrants of each species are distributed multinomially, each site receiving them with probability proportional to the number of emigrants from the other sites (a uniform kernel; an individual may return to its own site);
- spatially explicit landscapes (landscape_grid()): the number of immigrants of each species at each site is drawn from a Poisson distribution whose mean is the expected flow of emigrants through the dispersal kernel. This approximation does not conserve the number of dispersing individuals exactly, but has the right expectation. The code of Thompson et al. (2020) does the same. Few emigrants may then produce no immigrant, so with extirpation and low dispersal, recolonisation is rarer than in mcomsimr, which conserves dispersers.

After dispersal, each population is lost with probability extirpation, independently of its size. Thompson et al. (2020) used 0.002 in their competition-colonisation scenario and no extirpation in the others.

During colonisation, Poisson(seed_mean) individuals of every species are added to every site at each of the seed_steps. After the last of these, species absent from all sites can never return and are no longer computed.

Value

An mc_sim object: a list with abundance (sites x species matrix at the last step), history (sites x species x recorded steps array, or NULL), and the landscape, traits, alpha, dispersal and extirpation used.

Article and archived code

The article of Thompson et al. (2020) and the code archived with it (doi:10.5281/zenodo.3708350) differ in points that change results. Beta is gamma / alpha in the code and in their figures, but gamma - alpha in their Table 1. Their environment is converted to ranks, so it is uniform between 0 and 1, and the lowest dispersal is 0.00001, not 0.0001. Emigrants are Poisson in the code and binomial in the article, which comdynamite follows. In their competition-colonisation scenario, the code also halves the immigrants. See the article "Validating the model of Thompson et al. (2020)" on the package website for the complete list.

References

Thompson, P. L., Guzman, L. M., De Meester, L., Horváth, Z., Ptacnik, R., Vanschoenwinkel, B., Viana, D. S., & Chase, J. M. (2020). A process-based metacommunity framework linking local and regional scale community ecology. *Ecology Letters*, 23, 1314–1329. doi:10.1111/ele.13568

Examples

```
set.seed(1)
ls <- landscape_implicit(30)
tr <- treeshapesim::sim_tree_shape(20)
sim <- simulate_metacommunity(ls, species_traits(tr),
                              competition_matrix(tr, "phylo"),
                              dispersal = 0.01, n_steps = 200)
sim
```

---

simulate_static *Simulate a static metacommunity*

Description

Draws species abundances directly from unimodal responses to an environmental gradient, without population dynamics, competition or dispersal. The expected abundance of species j at site i is `r_max[j] * exp(-(env[i] - optimum[j])^2 / (2 * sigma[j]^2))`, the Gaussian response of Dray & Legendre (2008), also used by Peres-Neto et al. (2012). By default, observed abundances are Poisson counts with that mean, as in Peres-Neto & Kembel (2015) and Duarte et al. (2016), where `r_max` lies between 0 and 30. In the two earlier studies `r_max` lies between 0.5 and 1, abundances are the expected values (`sampling = "expected"`), and Dray & Legendre (2008) draw presences from a binomial distribution with the expected value as the probability (`sampling = "binomial"`). Because nothing but the gradient structures the data, this is a baseline for `simulate_metacommunity()` and a fast generator for evaluating methods.

Usage

```
simulate_static(
  traits,
  env,
```

```
  incidence = FALSE,
  sampling = c("poisson", "binomial", "expected")
)
```

Arguments

traits	Data frame from species_traits(), or any data frame with columns optimum, sigma and r_max, or a list of them, one per gradient, with the same species. Here r_max is the maximum expected abundance, not a growth rate, so set rmax_range accordingly. With several gradients the maxima multiply, so a range such as c(1, 1) in all but one of them keeps abundances on the scale of the first.
env	Value of the gradient at each site, or a list or data frame with one gradient per element, in the order of traits.
incidence	Return presences and absences instead of Poisson counts? A species is present where its count is positive. Only for sampling = "poisson".
sampling	How to obtain the data from the expected values: "poisson" draws counts, "binomial" draws presences and absences, which needs expected values of at most 1, and "expected" returns the expected values.

Details

With several gradients, for example an environmental and a spatial one, each species has its own response to each gradient and the expected abundance is the product of the responses, as in Peres-Neto et al. (2012).

Value

List of class mc_sim with abundance, a matrix with one row per site and one column per species, plus env and traits. Sites and species with no individuals are kept, so remove them if a method requires it.

References

Dray, S., & Legendre, P. (2008). Testing the species traits-environment relationships: the fourth-corner problem revisited. *Ecology*, 89, 3400–3412. doi:10.1890/080349.1

Duarte, L. D. S., et al. (2016). Dissecting phylogenetic fuzzy weighting: theory and application in metacommunity phylogenetics. *Methods in Ecology and Evolution*, 7, 937–946. doi:10.1111/2041210X.12547

Peres-Neto, P. R., & Kembel, S. W. (2015). Phylogenetic gradient analysis: environmental drivers of phylogenetic variation across ecological communities. *Plant Ecology*, 216, 709–724. doi:10.1007/s1125801404050

Peres-Neto, P. R., Leibold, M. A., & Dray, S. (2012). Assessing the effects of spatial contingency and environmental filtering on metacommunity phylogenetics. *Ecology*, 93, S14–S30. doi:10.1890/110494.1

Examples

```
set.seed(1)
tr <- treeshapesim::sim_tree_shape(20)
traits <- species_traits(tr, lambda = 1, optima_range = c(-1, 101),
                         rmax_range = c(0, 30))
sim <- simulate_static(traits, env = runif(100, 0, 100))
sim

# an environmental and a spatial gradient
space <- species_traits(tr, lambda = 1, optima_range = c(-1, 101),
                        rmax_range = c(1, 1))
gradients <- data.frame(env = runif(100, 0, 100), space = 1:100)
simulate_static(list(traits, space), gradients)
```

---

species_traits *Species traits, with or without phylogenetic signal*

Description

Environmental optima evolve by Brownian motion with Pagel's lambda and are rescaled to `optima_range` (or, with `uniform_optima = TRUE`, replaced by ranks with evenly spaced values on that range, so that only their arrangement on the tree depends on the tree). Niche breadths are drawn from a normal distribution truncated at zero, and maximum growth rates from a uniform distribution. Either can carry phylogenetic signal: the values are then assigned to species by the ranks of a Brownian-motion trait, which keeps their marginal distribution.

Usage

```
species_traits(
  species,
  lambda = 1,
  optima_range = c(0, 100),
  uniform_optima = FALSE,
  sigma_mean = 10,
  sigma_sd = 10,
  rmax_range = c(2, 8),
  sigma_signal = FALSE,
  rmax_signal = FALSE
)
```

Arguments

`species`	A `phylo` tree, or the number of species for traits without phylogenetic structure.
`lambda`	Pagel's lambda of the optima (0 = no signal, 1 = Brownian motion). Ignored when `species` is a number.
`optima_range`	Range of the optima.

`uniform_optima` Give the optima evenly spaced values by rank?

`sigma_mean`, `sigma_sd`
: Mean and standard deviation of the normal distribution (truncated at zero) of niche breadths.

`rmax_range` Range of the uniform distribution of maximum growth rates.

`sigma_signal`, `rmax_signal`
: Should niche breadth or maximum growth rate carry phylogenetic signal (Brownian motion)?

Value

Data frame with columns `optimum`, `sigma` and `r_max`, one row per species, with species names as row names.

Examples

```
set.seed(1)
tr <- treeshapesim::sim_tree_shape(10)
species_traits(tr, lambda = 1, sigma_signal = TRUE)
species_traits(5)
```

`torus_permutations` *Site permutations that keep the structure of a torus*

Description

The translations of a `side` x `side` torus combined with the eight symmetries of the square keep all distances among sites, and therefore the spatial autocorrelation of the environment. They give a restricted null model for spatially explicit metacommunities (see the vignette).

Usage

```
torus_permutations(side)
```

Arguments

`side` Number of cells per side, as in `landscape_grid()`.

Value

Integer matrix with one permutation of the site indices per row, excluding the identity (`8 * side^2 - 1` rows when `side > 2`).

Examples

```
p <- torus_permutations(4)
dim(p)
```

trait_bm *Trait evolution by Brownian motion with Pagel's lambda*

Description

Draws from a multivariate normal distribution with covariance `lambda * C + (1 - lambda) * diag(C)`, where `C` is the phylogenetic covariance matrix of `tree`.

Usage

```
trait_bm(tree, lambda = 1)
```

Arguments

`tree`	A `phylo` tree with branch lengths.
`lambda`	Pagel's lambda, in [0, 1].

Value

Named numeric vector with the simulated trait value of each tip.

Examples

```
set.seed(1)
trait_bm(treeshapesim::sim_tree_shape(5), lambda = 0.5)
```

trait_partition *Partition trait or phylogenetic variation between environment and space*

Description

The phylogenetic gradient analysis of Peres-Neto et al. (2012). Trait values are standardised with species weights, which gives each site a mean and a standard deviation. The chosen component is related to `env` and `space` with site weights, and the variation is split into the fractions of Peres-Neto et al. (2006): `a`, environment alone; `b`, shared; `c`, space alone; and `d`, unexplained.

Usage

```
trait_partition(
  comm,
  traits,
  env,
  space,
  component = c("mean", "variance"),
  n_perm = 999
)
```

Arguments

comm	Matrix with one row per site and one column per species. Every site needs at least one individual.
traits	Numeric vector or matrix with one row per species, which can hold traits or the vectors of phylo_vector(). Species are matched to comm by name, in any order, and those in only one of the two are dropped with a warning.
env, space	Numeric vector, or matrix or data frame, with one row per site. For space, the article uses Moran's eigenvector maps, which adespatial::dbmem() gives.
component	"mean" for the composition of sites or "variance" for their dispersion.
n_perm	Number of permutations of sites and of species.

Details

The total variation includes both the mean and the dispersion within sites, so r2 answers how much of all the variation each fraction holds, and r2_component how much of the component. Fractions b and d cannot be tested.

Value

List of class mc_trait_partition with fractions, a data frame with one row for each of abc, ab, bc, a, b, c and d, and the columns r2, r2_adjusted, r2_component, p_site, p_taxa and p_max; component; component_share, the share of the total variation in the component; semipartial, a matrix of predictors by traits; and p_partial, p_partial_site and p_partial_taxa, with the larger p-value and those of sites and of species, all NA if predictors are collinear; and permutations.

Article and original code

Raw and adjusted fractions of the total for the mean component agree with PartitionRDATwoDataMatricesPhylogeny of the supplement, checked in Octave. The code departs from Appendix A in these points, where we follow the text:

- Fractions relative to the component: Meana and Meanc subtract the fractions of the total from the fraction of the component.
- Adjusted values relative to the component use the number of predictors of both sets in the models of one set, and the result can exceed one.
- Total variation: Appendix A multiplies it by the number of traits, and the code does not.
- Several vectors: the article analyses the vectors of environment and space together in a redundancy analysis, and the script passes only the first.
- Variance component: the explained sum of squares contains the intercept, so b absorbs the mean dispersion. Appendix A asks for the intercept in the models. Here it is removed, and r2 is the variation around the weighted mean.
- Permutations of the variance component: the code takes the square root twice, sets small values to zero in the observed matrix instead of the permuted one, and rescales the permuted values to the observed mean and standard deviation, which the text does not mention. It also calls WeightedMeanVar, which the supplement does not include. None of this is done here.
- The code runs n_perm - 1 permutations in each null distribution.

- The text stacks two null distributions, and here we report the larger of the two p-values, as in `phylo_vector()`.

Moran's eigenvector maps of the function `MEM` of the supplement are the same as those of `adespatial::dbmem()`, checked in Octave. Appendix A recommends centring them with site weights and the code does not, but orthogonality is not needed to partition variation, and here all predictors are standardised with site weights.

Semipartial correlations follow Appendix A: the square root of the loss of R2 when a predictor leaves the model of all predictors, with the sign of its slope when it is alone. The supplement has no code for them, so they were checked against separate regressions and not in Octave. Appendix A calculates the model without the predictor with the slopes of the full model, which it had just fitted again without it, and here we use the second slopes. With several traits it gives no rule, so here each trait has its own value, relative to the variation of that trait. They are tested as the fractions are, with the loss of R2 as the statistic, and with many predictors they need a correction.

This function partitions variation and does not test each trait against each predictor. For that test, ter Braak (2019) proposed multilevel and weighted averaging regressions combined by the max test, in the package TraitEnvMLMWA.

Note

Adapted from the MATLAB function `PartitionRDATwoDataMatricesPhylogeny` in the supplement of Peres-Neto et al. (2012).

References

Peres-Neto, P. R., Legendre, P., Dray, S., & Borcard, D. (2006). Variation partitioning of species data matrices: estimation and comparison of fractions. *Ecology*, 87, 2614–2625. doi:10.1890/0012-9658(2006)87[2614:VPOSDM]2.0.CO;2

Peres-Neto, P. R., Leibold, M. A., & Dray, S. (2012). Assessing the effects of spatial contingency and environmental filtering on metacommunity phylogenetics. *Ecology*, 93, S14–S30. doi:10.1890/110494.1

ter Braak, C. J. F. (2019). New robust weighted averaging- and model-based methods for assessing trait-environment relationships. *Methods in Ecology and Evolution*, 10, 1962–1971. doi:10.1111/2041210X.13278

See Also

`plot_trait_partition()`

Examples

```
set.seed(1)
env <- runif(40, 0, 100)
space <- as.matrix(adespatial::dbmem(cbind(1:40), silent = TRUE))[, 1:3]
trait <- setNames(runif(15, 0, 100), paste0("sp", 1:15))
comm <- outer(env, trait, function(e, t) exp(-(e - t)^2 / 800))
colnames(comm) <- names(trait)
out <- trait_partition(comm, trait, env, space, n_perm = 49)
out
```

```
plot_trait_partition(out)
```

`traits_ou` *Traits evolving by an Ornstein-Uhlenbeck process with one or several optima*

Description

Simulates traits along a tree under the Ornstein-Uhlenbeck (OU) model of Hansen (1997), with the multi-optimum ("multi-peak") extension of Butler & King (2004). Along each branch the trait is pulled towards the optimum `theta` of the selective regime of that branch with strength `alpha`, while drifting with rate `sigma2`. Values are drawn exactly from the transition distribution of the process: over a branch of length t, a trait with value x at the start has, at the end, a normal distribution with mean `theta + (x - theta) * exp(-alpha * t)` and variance `sigma2 / (2 * alpha) * (1 - exp(-2 * alpha * t))`.

Usage

```
traits_ou(
  tree,
  n_traits = 1,
  alpha = 0,
  sigma2 = 1,
  theta = 0,
  regimes = NULL,
  root = 0
)
```

Arguments

`tree`	A rooted `phylo` tree with branch lengths.
`n_traits`	Number of independent traits.
`alpha`	Strength of selection towards the optimum (0 = Brownian motion).
`sigma2`	Rate of the stochastic (drift) component.
`theta`	Optimum of each regime: one value for a single optimum, or a named vector with one value per regime (names as in `regimes`).
`regimes`	Selective regime of each branch, in the order of the edges of `tree` reordered "cladewise" (`ape::reorder.phylo(tree, "cladewise")`), as returned by `paint_regimes()`. `NULL` puts every branch in one regime.
`root`	Trait value at the root.

Details

With `alpha = 0` the process is Brownian motion with rate `sigma2`, and the optima play no role. Each branch belongs to a single regime, so a shift in optimum happens at the start of the branch where the regime changes.

Value

Matrix with one row per tip (tip labels as row names) and one column per trait.

References

Butler, M. A., & King, A. A. (2004). Phylogenetic comparative analysis: a modeling approach for adaptive evolution. *The American Naturalist*, 164, 683–695. doi:10.1086/426002

Hansen, T. F. (1997). Stabilizing selection and the comparative analysis of adaptation. *Evolution*, 51, 1341–1351. doi:10.1111/j.15585646.1997.tb01457.x

See Also

`paint_regimes()` to assign regimes to clades.

Examples

```
set.seed(1)
tr <- treeshapesim::sim_tree_shape(20)
# Brownian motion
traits_ou(tr, alpha = 0)[1:3, ]
# two optima: the clade of node 25 evolves towards 10, the rest towards 0
reg <- paint_regimes(tr, nodes = 25)
x <- traits_ou(tr, alpha = 2, theta = c(`0` = 0, `1` = 10), regimes = reg)
round(x[, 1], 1)
```

Index

∗ **datasets**
    daphniids, 7

ape::plot.phylo(), *27*
assemble_community, 3

biogeography_matrix, 4
biogeography_matrix(), *16, 17*

competition_matrix, 6
competition_matrix(), *13*, *31, 32*

daphniids, 7, *16*
dispersal_kernel, 8
dispersal_kernel(), *10*

env_over_time, 8
env_over_time(), *32*

graphics::image(), *26*
graphics::plot.window(), *29*
growth_rates, 9
growth_rates(), *31*

landscape_grid, 10
landscape_grid(), *11*, *32*, *36*
landscape_implicit, 11
landscape_implicit(), *32*
logistic_gradients, 11
logistic_gradients(), *30*

mc_dynamics, 12

node_contrasts, 14
node_contrasts(), *15*, *27*
node_test, 15
node_test(), *4, 5*, *26–28*

paint_regimes, 18
paint_regimes(), *40, 41*
pcc_test, 18
pcc_test(), *17*, *22*
phylo_beta, 20
phylo_beta(), *21*
phylo_beta_test, 21
phylo_beta_test(), *17*, *20*
phylo_eigenvectors, 22
phylo_eigenvectors(), *19*
phylo_vector, 23
phylo_vector(), *38, 39*
plot_node_heatmap, 26
plot_node_heatmap(), *17*
plot_node_test, 27
plot_node_test(), *17*, *26*
plot_trait_partition, 28
plot_trait_partition(), *39*

simulate_logistic, 30
simulate_metacommunity, 31
simulate_metacommunity(), *9*, *12, 13*, *33*
simulate_static, 33
species_traits, 35
species_traits(), *9*, *32*, *34*

torus_permutations, 36
torus_permutations(), *10*, *15, 16*, *19*, *21*, *24*
trait_bm, 37
trait_partition, 37
trait_partition(), *24*, *28, 29*
traits_ou, 40
traits_ou(), *3*, *18*