Package 'idar'

February 12, 2022

Version 1.3

Date 2022-02-12

Title Individual Diversity-Area Relationships

Author Marcelino de la Cruz

Depends FD, picante, spatstat (>= 2.0-0)

Imports ape, spatstat.geom, spatstat.core, spatstat.random

Suggests ecespa, vegan

Maintainer Marcelino de la Cruz <marcelino.delacruz@urjc.es>

Description Computes and tests individual (species, phylogenetic and functional) diversity-area relationships, i.e., how species-, phylogenetic- and functional-diversity varies with spatial scale around the individuals of some species in a community. See applications of these methods in Wiegand et al. (2007) <doi:10.1073/pnas.0705621104> or Chacon-Labella et al. (2016) <doi:10.1007/s00442-016-3547-z>.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-02-12 18:10:02 UTC

R topics documented:

checktree . . . 2
envelope4idar . . . 3
fdis . . . 8
ipsim . . . 9
isar . . . 13
LF.gof . . . 16
localdar . . . 18
midar . . . 22
mitable . . . 24
pisar . . . 25
proportion.idar . . . 29
SF . . . 32

Index **34**

checktree *Check Trait and Phylo Data*

Description

Checks consistency of names of species in spatial, phylogenetic and traits data.

Usage

```
checktree(tree,  mippp, idar, correct.phylo)
checktraits(traits, mippp, idar, correct.trait.na, correct.trait)
```

Arguments

`tree`	A phylogenetic tree in `phylo` format (**ape**) or a phylogenetic covariance matrix
`traits`	A data.frame of traits, or a distance matrix among species (in dist or matrix format) computed on a data.frame of traits.
`mippp`	A multitype (a.k.a. multivariate) marked point pattern . An object with the ppp format of **spatstat**, with the names of species as marks.
`idar`	Character. The name of the idar fucntion to be computed. Either "isar", "ipsvar", "ipsrar", "ipsear", "ipscar", or "imntdar"
`correct.phylo`	Character. Either `"mean"` meaning *"include missing species in the tree with a constant mean phylogenetic covariance"* or `"exclude"`, meaning *"exclude missing species in the tree from the analysis"*
`correct.trait.na`	Logical flag indicating whether `NA` values in the matrix of traits should be "corrected": NA values will be assigned the mean trait value.
`correct.trait`	Character. Either `"mean"` or `"exclude"`. Species missing in the data.frame of traits will be assigned mean trait values or will be excluded from the analysis, respectively.

Details

This functions check for the coincidence of species in the point pattern and in the phylogenetic or trait data and for the existence of missing data. If `correct.phylo="mean"`, species which are absent in the phylogenetic tree or covariance matrix will be included and assigned mean phylogenetic covariance. If `correct.phylo="exclude"`, missing species in the tree will be excluded from the analysis (i.e., will not be considered in the computation of the local phylogenetic indices). If `correct.trait.na="TRUE"`, NA values for traits in the data.frame of traits will be assigned the mean trait value. If `correct.trait="mean"`, missing species in the data.frame of traits will be included and assigned mean trait values. If `correct.trait="exclude"`, missing species in the data.frame of traits will be excluded from the analysis (i.e., will not be considered in the computation of the local functional indices).

Value

checktree returns a covariance matrix with the appropriate species corrections. checktraits returns a data.frame of traits with the appropriate corrections or a Gower distance matrix among the species from the corrected data.frame of traits if idar="ifdar".

Warning

The transcription of species names in the multivariate mippp, in the row names of the data.frame of traits (or in the names or dimnames of the distance matrices) should be identical. The same applies to the tiplabels of the phylogenetic tree.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

---

envelope4idar *Compute Simulation Envelopes for IDAR(r) Functions*

Description

Compute simulation envelopes for IDAR(r) functions.

Usage

```
  envelope4idar(mippp, mippp.sp.sim, mippp.sp, mimark=NULL,
    namesmark=NULL, r, idar="isar", buffer="adapt", bfw=NULL,
    nsim=NULL, nrank=1, tree = NULL, traits = NULL,
    cross.idar=FALSE, savefuns=TRUE, correct.phylo="exclude",
    correct.trait.na=FALSE, correct.trait="mean" )
idar2(mippp.sp, mippp, mimark, idar="isar", buffer,bfw, r,
    cross.idar=FALSE,  tree = NULL, traits = NULL)
    raoDmod(comm, phy = NULL)
```

Arguments

mippp	A multitype (a.k.a. multivariate) marked point pattern. An object with the ppp format of **spatstat**.
mippp.sp	Univariate point pattern of the focal species. An object with the ppp format of **spatstat**.
mippp.sp.sim	A list with simulations of the focal species point pattern created with simulador2
mimark	Character. Name of the focal species in the multitype mippp.
namesmark	Character. If the marks in mippp are within a data.frame, the name of the column with the species names
r	Vector of distances to compute IDAR(r) functions

idar	Character. The name of the idar fucntion to be computed. Either "isar", "ipsvar", "ipsrar", "ipsear", "ipscar", "icwmar", "icwmar.O", "iraodar"or "imntdar"
buffer	Character or numeric. Either "adapt" (i.e., compute an adaptive buffer), or a number indicating de width of a fixed buffer area around the plot border
bfw	An owin object indicating the limits of the buffer area.
nsim	The number of simulations.
nrank	Integer. Rank of the envelope value amongst the nsim simulated values. A rank of 1 means that the minimum and maximum simulated values will be used.
tree	A phylogenetic tree in phylo format (**ape**) or a phylogenetic covariance matrix
traits	A data.frame of traits, or a distance matrix among species (in dist or matrix format) computed on a data.frame of traits.
cross.idar	Logical. If TRUE, the focal pattern be excluded form the community being measured.
savefuns	Logical flag indicating whether to save all the simulated function values.
correct.phylo	Character. Either "mean" meaning *"include missing species in the tree with a constant mean phylogenetic covariance"* or "exclude", meaning *"exclude missing species in the tree from the analysis"*
correct.trait.na	Logical flag indicating whether NA values in the matrix of traits should be "corrected": NA values will be assigned the mean trait value.
correct.trait	Character. Either "mean" or "exclude". Species missing in the data.frame of traits will be assigned mean trait values or will be excluded from the analysis, respectively.
comm	A data.frame with community data
phy	a phylogenetic tree in phylo format (**ape**) or more probably, a distance matrix.

Details

In 2007, Wiegand et al. developed the concept of Individual Species-Area Relationship. Basically, this consist in computing species accumulation curves by samping areas with varying radius r around the individual trees of a *focal* species. Here we provide a version of *ISAR(r)* (idar="isar"), but we also extend this concept to other diversity-area relationships and provide functions to compute individual phylogenetic diversity-area and individual functional diversity-area relationships. The individual phylogenetic functions are based in Helmus et al. (2007) measures, i.e., phylogenetic species variability (idar="ipsvar"), phylogenetic species richness (idar="ipsrar"), phylogenetic species evenness (idar="ipsear"), and phylogenetic species clustering (idar="ipscar"). Also, an individual version of the mean nearest taxon distance of Webb et al. (2002) (idar="imntd"). The individual functional-diversity function (idar="ifdar") is based in the functional dispersion measure (FDis) of Laliberté and Legendre (2010). Other available functions are based in a weighted community mean of traits (idar="icwmar"), on a verison of Rao quadratic entropy (idar="iraodar"), and on the same functions but computed in "rings" defined by the neighbour r values (idar="icwmar.O" and idar="iraodar.O"), which would eliminate the "accumulative" or "memory" effect on *ICWMAR(r)* and *IRAODAR(r)*.

Although recent litterature (e.g., Wiegand and Moloney 2014) suggest that buffer correction is not necessary for this type of statistics, and by default all functions are estimated without buffer (e.g.,

buffer=0), several edge correction coould be employed. For example, an adaptative buffer correction could be used (buffer="adapt"), i.e., for each radius r, only individuals of the focal species that are placed at a distance $>= r$ from the border of the plot are considered in the computation of the different meassures. It is also possible to set a fixed buffer width (e.g., buffer=30), which will accelerate te computations but will discard many useful trees. It is also possible to provide also a fixed window (in the argument bfw) to indicate the limits of the buffer area. This could be useful to computing the IDAR(r) functions in different subsets of the original plot (e.g., in different "habitats").

In general, computing envelopes with envelope4idar is a little faster than using envelope and the individual functions (e.g., ipsvar, ifdar, etc). In addition, envelope4idar has the possibility of computing "crossed" individual functions, i.e., using a focal species that is not part of the community whose diversities are being measured. This allows evaluating the diversity of e.g., young trees around older trees, etc. This is accomplished by seting the argument cross.idar=TRUE.

While envelope4idar manages data and results, idar2 actually computes the individual functions (both observed and simulated) . In general, idar2 would not be called directly by the user.

raoDmod is a modification of the function raoD of **picante** to accept distance matrices instead of phylogenetoc trees. It would not be called directly by the user.

Value

An object of class "fv", see fv.object, and envelope,which can be plotted directly using plot.fv. Essentially a data frame containing columns

r the vector of values of the argument r at which the idar(r) function has been estimated

obs values of the summary function for the data point pattern

lo lower envelope of simulations

hi upper envelope of simulations

nmean estimated theoretical value of the summary function under the considered null model, computed by averaging simulated values

Warning

The transcription of species names in the multivariate mippp, in the row names of the data.frame of traits (or in the names or dimnames of the distance matrices) should be identical. The same applies to the tiplabels of the phylogenetic tree.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

References

Helmus M.R., Bland T.J., Williams C.K. and Ives A.R. (2007) Phylogenetic measures of biodiversity. *American Naturalist*, 169, E68-E83.

Laliberté, E. and Legendre, P. (2010) A distance-based framework for measuring functional diversity from multiple traits. *Ecology* 91, 299-305.

Wiegand,T., Gunatilleke, C.V.S., Gunatilleke, I.A.U.N. and Huth, A. (2007) How individual species structure diversity in tropical forests. *PNAS* 104, 19029-19033.

Webb, C., Ackerly, D., McPeek, M., and Donoghue M. 2002. Phylogenies and community ecology. *Annual Review of Ecology and Systematics* 33:475-505

See Also

psd for a description of the phylogenetic measures of Helmus et al. (2007).

fdisp for a description of the functional dispersion measure (FDis) of Laliberté and Legendre (2010).

Examples

```
# compute envelope for isar around sp_44
data(SF)
data(SFtraits)
data(SFphylotree)


# 1) Create a list with simulations of the focal species point pattern  with simulador2()
#     or by hand. # Beware that each simulated ppp should be marked (with the mark
#      of the focal species)

#  Example for simulations of an inomogenous PP
# Adjust an IPP:

    sp_44.ipp<- density.ppp(unmark(SF[SF$marks$species=="sp_44"]))

# simulate 19 realizations of the adjusted IPP
# (BEWARE: in real tests you should use 199 or higher)
    sp_44.ipp.sim <- vector(mode="list", le=19)
    sp_44.ipp.sim <- lapply(sp_44.ipp.sim, function(x) x=rpoispp(sp_44.ipp))
   # mark each simulated pattern
    sp_44.ipp.sim.m <- lapply(sp_44.ipp.sim, function(x)
                                      {marks(x) = factor(rep("sp_44", x$n)); return(x)})

# 2) compute.envelopes
     # ISAR
  isar.sp_44.ipp.env<- envelope4idar(mippp=SF, mippp.sp.sim= sp_44.ipp.sim.m,
                mimark="sp_44", namesmark="species",r=1:30, buffer=0)
## Not run:
     # IFDAR
 ifdar.sp_44.ipp.env<- envelope4idar(mippp=SF, mippp.sp.sim= sp_44.ipp.sim.m,
                mimark="sp_44",  namesmark="species", r=1:30, idar="ifdar", buffer=0,
 traits=SFtraits, correct.trait.na=TRUE)

    #IPSVAR
   ipsvar.sp_44.ipp.env<- envelope4idar(mippp=SF, mippp.sp.sim= sp_44.ipp.sim.m,
                 mimark="sp_44", namesmark="species", r=1:30, idar="ipsvar", buffer=0,
```

```
  tree=SFphylotree)

####################################
# Computing CROSS_IDAR
#-------------------------------------------------

# You need a focal pattern that its not part of the multivariate pattern which is "measured"
# to estimate diversity.
# For example, let's measure diversity of small trees  around large trees of the focal species.

# First, obtain the pattern of large and small trees
# the pattern of small trees will be the "measured" one, i.e., the argument "mippp"

SFlarge<- SF[SF$marks$dbh>=10]
SFsmall <- SF[SF$marks$dbh<10]

# pattern of the focal species (this will be the argument "mippp.sp")
sp_44.large<- SFlarge[SFlarge$marks$species=="sp_44"]

# list of simulated patterns of the focal species (e.g., from an IPP)
sp_44.large.ipp<- density.ppp(sp_44.large)
sp_44.large.ipp.sim <- vector(mode="list", le=99)
sp_44.large.ipp.sim <- lapply(sp_44.large.ipp.sim, function(x) x=rpoispp(sp_44.large.ipp))

# COMPUTE envelopes for cross-ISAR(r)
 isar.sp_44.large.cross.ipp.env<- envelope4idar(mippp=SFsmall, r=1:30, buffer=0,
mippp.sp.sim= sp_44.large.ipp.sim, mippp.sp=sp_44.large,
                        namesmark="species", cross.idar =TRUE)

# COMPUTE envelopes for cross-IFDAR(r)
ifdar.sp_44.large.cross.ipp.env<- envelope4idar(mippp=SFsmall, r=1:30, idar="ifdar",
                     buffer=0, mippp.sp.sim= sp_44.large.ipp.sim,  mippp.sp=sp_44.large,
                       namesmark="species", traits=SFtraits, correct.trait.na=TRUE,
                       cross.idar =TRUE)

# COMPUTE envelopes for cross-IPSVAR(r)
ipsvar.sp_44.large.cross.ipp.env<- envelope4idar(mippp=SFsmall, r=1:30, idar="ipsvar",
                     buffer=0, mippp.sp.sim= sp_44.large.ipp.sim,  mippp.sp=sp_44.large,
                      namesmark="species",  tree=SFphylotree, cross.idar =TRUE)


####################################
# Comparing the performance of envelope() and envelope4idar()
#
#-------------------------------------------------
require(ecespa) # for the ipc.estK() function
data(SF)
SFsp<- unmark(SF)
marks(SFsp)<- SF$marks$species
sp_44.ppp<-unmark(SF[SF$marks$species=="sp_44"])
sp_44.pc<- ipc.estK(sp_44.ppp)
```

```
# use multifocalsimulator() to use the same simulations with both functions.
sp_44.pc.sim.mf0<-multifocalsimulator(pp=SFsp, mimark="sp_44",
                        simulate=expression(rIPCP(sp_44.pc)), nsim=99, nmin=sp_44.ppp$n)

# envelopes with function envelope()
gc()
t0<- Sys.time()
  ifdar.sp_44.pc.env.e2<- envelope(SFsp, fun=ifdar, mimark="sp_44", traits=SFtraits,
                       correct.trait.na=TRUE, nsim=99, simulate=sp_44.pc.sim.mf0, r=1:30,
                            savefuns=TRUE, buffer=0)
Sys.time()-t0

# envelopes with function envelope4idar()

# Here you should input simulated patterns only for the focal species so, first,
# extract it fom the list of simulated multivariate ppp

sp_44.pc.sim.mf00<- lapply(sp_44.pc.sim.mf0, function(x) {x=x[x$marks=="sp_44"];return(x)})

gc()
t0<- Sys.time()
ifdar.sp_44.pc.env2<- envelope4idar(mippp=SF, mippp.sp.sim= sp_44.pc.sim.mf00, mimark="sp_44",
namesmark="species", r=1:30, idar="ifdar", buffer=0,
                                nsim=99, traits=SFtraits, correct.trait.na=TRUE)
Sys.time()-t0



## End(Not run)
```

`fdis` *Average Functional Dispersion*

Description

Computes average Functional Dispersion for several communities.

Usage

```
fdis(x, traits)
```

Arguments

`x`	A community data matrix containing the abundances of the species in the different communities. Rows are sites and species are columns.
`traits`	A distance matrix among species in `matrix` format, i.e. obtained using `as.matrix`(*d*), where *d* is a `dist` object from `gowdis`, `dist` or similar functions.

Details

This function is a wrap to fdisp in package **FD**. It manages some of the possible problems that could appear when computing automatically functional dispersion for local communities in ifdar (mainly "empty" communities, common when computing ifdar(r) for very small *r*'s). It is a kind of internal function that wouldn't be usually called by the user.

Value

Numeric. The average functional dispersion of the communities in matrix x

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

References

Laliberté, E. and Legendre, P. (2010) A distance-based framework for measuring functional diversity from multiple traits. *Ecology* 91, 299-305.

See Also

fdisp for a description of the functional dispersion measure (FDis) of Laliberté and Legendre (2010).

---

ipsim *Simulate Multivariate Point Patterns*

Description

These functions simulate multivariate point patterns from a variety of null models, in the way required to test IDAR(r) functions.

Usage

```
ipsim(pp, mimark,sigma=0, lambda=NULL, namesmark=NULL)
ipsimlist(pp, mimark, listsim)
simulador2(mimark, milambda, nsim=99)
multifocalsimulator(pp, mimark, simulate,nsim=99,nmin=NULL)
```

Arguments

pp	A multitype (a.k.a. multivariate) marked point pattern. An object with the ppp format of **spatstat**
mimark	Character. Name of the focal species in the multitype pp.
namesmark	Character. If the marks in pp are within a data.frame, the name of the column with the species names

sigma	Sigma for the Gaussian kernel to estimate the intensity of the point pattern to simulate
listsim	List with simulated point paterns from simulador2
lambda	intensity surface, e.g., an image from density.ppp or predict.ppm
milambda	intensity surface, e.g., an image from density.ppp or predict.ppm
nsim	number of simulations
nmin	expected minimum number of points in each simulated pattern
simulate	either a list of pre-computed univariate point patterns or an expression in the R language indicating how to simulate the patterns.

Details

This functions produce simulated point patterns appropriate to to compute envelopes of IDAR(r) functions. The usual tests of IDAR(r) functions require that the multivariate (i.e. multispecies) pattern remains fixed, except for the focal species, that is simulated acording to, e.g., an (inhomogeneous) Poisson process.

ipsim returns the multivariate pp pattern with all species in the same locations except the "focal" one (i.e., the one indicated by the argument mimark) that is simulated using rpoispp. If an intensity surface (argument lambda) is provided, the focal species will be simulated from this surface. If no lambda is provided but the argument sigma is >0, an intensity surface will be estimated with a Gaussian kernel with the sigma provided (using density.ppp) and the simulation will be made form this surface. If no lambda is provided and sigma=0, a homogeneos Poisson process will be simulated.

simulador2 generates a list (length = *nsim*) of marked (with mark = *mimark*) univariate point patterns from an intensity surface using rpoispp.

ipsimlist uses the results of simulador2 and the multivariate pp pattern to generate a list of multivariate point patterns with all species in the same locations except the "focal" one (i.e., the one indicated by the argument mimark) that has the locations simulated with simulador2.

multifocalsimulator is more flexible and allows the simulation of whichever null model of the focal species that could be described by an R expression.

Value

ipsim produces a multivariate point pattern (with the ppp format of **spatstat**); ipsimlist and multifocalsimulator produce a list of multivariate point patterns; simulador2 produces a list of univariate marked patterns.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

Examples

```
# Build a multivariate point pattern where maple is simulated according to a Poisson process
# and where all the other species are keep fixed in their original coordinates.
```

```
# (The warnings are because in the original lansing point pattern there is a dupplicated
#  point)

data(lansing)
ipsim(pp=lansing, mimark="maple")

# Build a multivariate point pattern where maple is simulated according to an Inhomogeneous
# Poisson process from an intensity surface estimated "on the fly" with a Gaussian kernel with
# sd = "sigma", and where all the other species are keep fixed in their original coordinates.

ipsim(pp=lansing, mimark="maple", sigma=0.1)

# Build a multivariate point pattern where maple is simulated according to an Inhomogeneous
# Poisson process from a predefined  intensity surface "lambda" and where all the other
# species are keep fixed in their original coordinates. "Lambda" is an im object resulting
# from density.ppp(), from predict.ppm() or converted from any other rasterized image.

maple.lambda<- density.ppp(lansing[lansing$marks=="maple"])
ipsim(pp=lansing, mimark="maple", lambda=maple.lambda)

# Build a list of 19 multivariate point pattern where maple is simulated according to an
# Inhomogeneous Poisson process from a predefined  intensity surface "lambda" and where all
# the other species are keep fixed in their original coordinates. "Lambda" is an im object
# resulting  from density.ppp(), # from predict.ppm() or converted from any other rasterized
# image.

  # Estimate the intensity of maple
   maple.lambda<- density(unmark(lansing[lansing$marks=="maple"]))

  # first simulate the individual maple patterns
   maple.sim<- simulador2(mimark="maple", milambda=maple.lambda, nsim=19)

  # Then, mix the simulated maple patterns with the rest of the multivariate pattern
  # (which remains "fixed")
   multi.maple.sim<- ipsimlist(pp=lansing, mimark="maple", listsim=maple.sim)


## Use of multifocalsimulator() ##

# The same but in a single step with multifocalsimulator(): Build a list of 19 multivariate
# point pattern where maple is simulated according to an Inhomogeneous Poisson process
# from a predefined  intensity surface "lambda" and where all the other species are keep
# fixed in their original coordinates. "Lambda" is an im object resulting from density.ppp(),
# from predict.ppm() or converted from any other rasterized image.

   # Estimate the intensity of maple
   maple.lambda<- density(unmark(lansing[lansing$marks=="maple"]))

   # get 99 simulated multivariate point patterns where only maple varies,
   # according to an inhomogeneous Poisson process
   multi.maple.sim <- multifocalsimulator(lansing, "maple", nsim=99,
                                          simulate=expression(rpoispp(maple.lambda)))
```

```
## Not run:
   # Use the simulated multivariate patterns to compute  envelopes for the ISAR against
   # a null model of IPP for maple
   isar.maple.env<- envelope(lansing, fun=isar, mimark="maple", nsim=99, savefuns=TRUE,
                                    r=seq(0.01, 0.25, le=100), simulate=multi.maple.sim)
   plot( isar.maple.env)



# Use multifocalsimulator()  to compute  envelopes for the ISAR against a null model of
# Poisson cluster for maple.

   # First, adjust a Poisson Cluster process to maple
   require(ecespa)
   maple.pc<- ipc.estK(unmark(lansing[lansing$marks=="maple"]))

  # generate list of simulated multivariate partterns (all other species fixed and maple
  # simulated according to the adjusted PC process):
   maple.pc.sim<-multifocalsimulator(pp=lansing, mimark="maple", nsim=99,nmin=NULL,
                                                 simulate=expression(rIPCP(maple.pc)))

  # compute envelopes
  isar.maple.pc.env<- envelope(lansing, fun=isar, mimark="maple", nsim=99, savefuns=TRUE,
                                     simulate=maple.pc.sim, r=seq(0.01, 0.25, le=100))
   plot( isar.maple.pc.env)


# Compute  envelopes for the IFDAR against a null model of Poisson cluster for sp_44 in
# San Francisco forest.
   data(SF)
   data(SFtraits)

  # first, get the original point pattern but with marks only for the species (i.e., dsicard
   # the data.frame of marks and keep only the vector of species names)
   SFsp<- unmark(SF)
   marks(SFsp)<- SF$marks$species

   # second, adjust a PCP to sp_44
   sp_44.pc<- ipc.estK(unmark(SFsp[SFsp$marks=="sp_44"]))

  # third, generate multivariate simulated patterns with only sp_44 varying according
  # to the adjusted PCP
   sp_44.pc.sim<-multifocalsimulator(pp=SFsp, mimark="sp_44",nsim=99,nmin=NULL,
                                                 simulate=expression(rIPCP(sp_44.pc)))

   # finally, compute envelopes
  ifdar.sp_44.pc.env<- envelope(SFsp, fun=ifdar, mimark="sp_44", traits=SFtraits, r=1:30,
                  correct.trait.na=TRUE, nsim=99, simulate=sp_44.pc.sim,  savefuns=TRUE)

   plot( ifdar.sp_44.pc.env)


## End(Not run)
```

isar *Individual Diversity Area Relationships*

Description

Estimate different Individual Diversity-Area Relationships from a multivariate point pattern.

Usage

```
isar(mippp, mippp.sp=NULL, mimark=NULL,  namesmark=NULL, r=NULL,
 buffer=0, bfw=NULL)
ipscar(mippp, mippp.sp=NULL, mimark=NULL,  namesmark=NULL,
 tree=NULL, r=NULL, buffer=0, bfw=NULL, correct.phylo="mean")
ipsear(mippp, mippp.sp=NULL, mimark=NULL,  namesmark=NULL,
 tree=NULL, r=NULL, buffer=0, bfw=NULL, correct.phylo="mean")
ipsvar(mippp, mippp.sp=NULL, mimark=NULL,  namesmark=NULL,
 tree=NULL, r=NULL, buffer=0, bfw=NULL, correct.phylo="mean")
ipsrar(mippp, mippp.sp=NULL, mimark=NULL,  namesmark=NULL,
 tree=NULL, r=NULL, buffer=0, bfw=NULL, correct.phylo="mean")
ifdar(mippp, mippp.sp=NULL, mimark=NULL,  namesmark=NULL,
 traits=NULL, r=NULL, buffer=0, bfw=NULL, correct.trait.na=FALSE,
   correct.trait="mean")
```

Arguments

`mippp`	A multitype (a.k.a. multivariate) marked point pattern. An object with the ppp format of **spatstat**.
`mippp.sp`	Univariate point pattern of the focal species. An object with the ppp format of **spatstat**.
`mimark`	Character. Name of the focal species in the multitype `mippp`.
`namesmark`	Character. If the marks in `mippp` are within a `data.frame`, the name of the column with the species names
`buffer`	One of "adapt", i.e., compute an adaptive buffer, or a number indicating de width of a fixed buffer area around the plot border
`bfw`	An owin object indicating the limits of the buffer area.
`r`	Vector of distances to compute IDAR(r) functions
`tree`	A phylogenetic tree in `phylo` format (**ape**) or a phylogenetic covariance matrix
`traits`	A data.frame of traits, or a distance matrix among species (in `dist` or `matrix` format) computed on a data.frame of traits.
`correct.phylo`	Character. Either `"mean"` meaning *"include missing species in the tree with a constant mean phylogenetic covariance"* or `"exclude"`, meaning *"exclude missing species in the tree from the analysis"*
`correct.trait.na`	Logical flag indicating whether `NA` values in the matrix of traits should be "corrected": NA values will be assigned the mean trait value.

`correct.trait` Character. Either `"mean"` or `"exclude"`. Species missing in the data.frame of traits will be assigned mean trait values or will be excluded from the analysis, respectively.

Details

In 2007, Wiegand et al. developed the concept of Individual Species-Area Relationship. Basically, this consist in computing species accumulation curves by samping areas with varying radius r around the individual trees of a *focal* species. Here we extend this concept to other diversity-area relationships and provide functions to compute individual phylogenetic diversity-area and individual functional diversity-area relationships. The individual phylogenetic functions are based in Helmus et al. (2007) measures, i.e., phylogenetic species variability (`ipsvar`), phylogenetic species richness (`ipsrar`), phylogenetic species evenness (`ipsear`), and phylogenetic species clustering (`ipscar`). The individual functional-diversity function (`ifdar`) is based in the functional dispersion measure (FDis) of Laliberté and Legendre (2010).

Although recent litterature (e.g., Wiegand and Moloney 2014) suggest that buffer correction is not necessary for this type of statistics, and by default all functionsare estimated without buffer (e.g., `buffer=0`), several edge correction coould be employed. For example, an adaptative buffer correction could be used (`buffer="adapt"`), i.e., for each radius r, only individuals of the focal species that are placed at a distance $>= r$ from the border of the plot are considered in the computation of the different meassures. It is also possible to set a fixed buffer width (e.g., `buffer=30`), which will accelerate te computations but will discard many useful trees. It is also possible to provide also a fixed window (in the argument `bfw`) to indicate the limits of the buffer area. This could be useful to computing the IDAR(r) functions in different subsets of the original plot (e.g., in different "habitats").

Value

An object of class `"fv"`, see fv.object, which can be plotted directly using plot.fv.

Essentially a data frame containing a column named *r* with the vector of values of the argument r at which the IDAR(r) function has been estimated and aonther column, named *"isar", "ipsvar", "ipsrar", "ipsear", "ipscar"* or *"ifdar"*, according to the selected idar argment. This column contains an estimate of the selected IDAR(R) function.

Warning

The transcription of species names in the multivariate `mippp`, in the row names of the `data.frame` of traits (or in the names or `dimnames` of the distance matrices) should be identical. The same applies to the `tiplabels` of the phylogenetic tree.

Simulation envelopes

To compute simulation envelopes for the IDAR(r) functions, use `envelope`. See the examples in this help page and in `ipsim` to know how to compute simulation envelopes from appropriate null models.

To compute envelopes for "crossed" IDAR(r) functions or to accelerate the computation of "single" IDAR(r) functions, use `envelope4idar`.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

References

Helmus M.R., Bland T.J., Williams C.K. and Ives A.R. (2007) Phylogenetic measures of biodiversity. *American Naturalist*, 169, E68-E83.

Laliberté, E. and Legendre, P. (2010) A distance-based framework for measuring functional diversity from multiple traits. *Ecology* 91, 299-305.

Wiegand,T., Gunatilleke, C.V.S., Gunatilleke, I.A.U.N. and Huth, A. (2007) How individual species structure diversity in tropical forests. *PNAS* 104, 19029-19033.

See Also

psd for a description of the phylogenetic measures of Helmus et al. (2007).

fdisp for a description of the functional dispersion measure (FDis) of Laliberté and Legendre (2010).

Examples

```
# ISAR
   # Point pattern with a data.frame of marks
   data(SF)
   isar.sp_44 <- isar(mippp = SF, mimark="sp_44", namesmark="species", r=1:40)
   plot(isar.sp_44)

   # Point pattern with just a vector of marks
    data(lansing)
    isar.blackoak <- isar(mippp = lansing, mimark="blackoak",  r=seq(0.01, 0.25, le=100))
    plot(isar.blackoak)

   # Examples of the use of different buffers
     # No  buffer at all (by deffault, buffer = 0)
     isar.sp_44.0 <- isar(mippp = SF, mimark="sp_44", namesmark="species", r=1:18)

     # Adaptive buffer (for each r, use only points within a r distance form the border)
     isar.sp_44.a <- isar(mippp = SF, mimark="sp_44", namesmark="species", r=1:18,
                                buffer="adapt")

     # Predefined window, for example with a buffer of 7 m within plot limits
     mibfw<- erosion(SF$win, r=7)
    isar.sp_44.w <- isar(mippp = SF, mimark="sp_44", namesmark="species", r=1:18, bfw=mibfw)


   ######################
   ### Phylogenetic functions ###
   ######################

data(SFphylotree)
```

```
# IPSCAR
    ipscar.sp_44 <- ipscar(mippp = SF, mimark="sp_44", namesmark="species", r=1:40,
                                    tree=SFphylotree)
    plot(ipscar.sp_44)

# IPSEAR
    ipsear.sp_44 <- ipsear(mippp = SF, mimark="sp_44", namesmark="species", r=1:40,
                                   tree=SFphylotree)
    plot(ipsear.sp_44)

# IPSVAR
    ipsvar.sp_44 <- ipsvar(mippp = SF, mimark="sp_44", namesmark="species", r=1:40,
                                    tree=SFphylotree)
    plot(ipsvar.sp_44)

# IPSRAR
    ipsrar.sp_44 <- ipsrar(mippp = SF, mimark="sp_44", namesmark="species", r=1:40,
                                   tree=SFphylotree)
    plot(ipsrar.sp_44)

  ######################
  ###  Functional functions ###
  ######################

data(SFtraits)

# IFDAR
   # this will cause an error becuse some species have NA's in the vector of trait values
## Not run:
  # ifdar.sp_44 <- ifdar(mippp = SF, mimark="sp_44", namesmark="species", traits=SFtraits,
  #                           r=1:40, correct.trait="exclude")

## End(Not run)
  # "correct" NA's in trait values  by assigning tospecies without traits the average of the trait
   # for  all the other species
   ifdar.sp_44 <- ifdar(mippp = SF, mimark="sp_44", namesmark="species",  traits=SFtraits,
       r=1:40, correct.trait.na=TRUE)

   #"correct"  the existence of NA's in trait values  by excluding species without traits from the
    # analysis
    ifdar.sp_44 <- ifdar(mippp = SF, mimark="sp_44", namesmark="species", traits=SFtraits,
                                 r=1:40, correct.trait.na=TRUE,correct.trait="exclude")

     plot(ifdar.sp_44)

# For examples of envelopes for these functions  see the help page of ipsim() or envelope4idar()
```

LF.gof *Loosmore and Ford Goodness of Fit Test*

Description

Performs the Loosmore and Ford (2006) test or the Maximum Absolute Deviation test for a spatial point pattern.

Usage

```
LF.gof(X, rmin=NULL, rmax=NULL, na.rm=TRUE)
```

Arguments

X	An object resulting from the function envelope, i.e., with an attribute "*simfuns*" (obtained using the argument savefuns=TRUE in envelope) ,which is an object of class "fv" containing the summary functions computed for each of the simulated patterns.
rmin	Minimum value of the function argument r over which the maximum absolute deviation, or the integral, will be computed for the test.
rmax	Maximum value of the function argument r over which the maximum absolute deviation, or the integral, will be computed for the test.
na.rm	Should NA's be removed to compute the integral?

Details

These function perform a tests for goodness-of-fit of a point pattern dataset to a point process model, based on Monte Carlo simulation from the model. The simulations should have been previously computed with the function envelope, applied with the argument savefuns=TRUE in order to save all the simulated functions, required for the computation of the test.

The test, popularized in the ecological field by Loosmore and Ford (2006) is also described in Diggle (2003, page 14), and according to Baddeley and Turner (2005) also in Diggle (1986) and Cressie (1991, page 667, equation (8.5.42)). If the arguments rmin and rmax are set to NULL, the integral of the GoF statistics will be computed over the complete range of r values.

Value

A list with the following components:

u The GoF statistic, i.e., the value of the integral over the range of *r*'s

p The p-value of the test

na.count.by.r Number of NA values for each r. It helps to evaluate the reliability of the computed u's, specially for small *r*'s

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

References

Cressie, N.A.C. (1991) *Statistics for spatial data*. John Wiley and Sons, 1991.

Diggle, P. J. (1986). Displaced amacrine cells in the retina of a rabbit : analysis of a bivariate spatial point pattern. *J. Neuroscience Methods* 18, 115-125.

Diggle, P.J. (2003) *Statistical analysis of spatial point patterns*, Second edition. Arnold.

Loosmore, N.B. and Ford, E.D. (2006) Statistical inference using the G or K point pattern spatial statistics. *Ecology* 87, 1925-1931.

See Also

`dclf.test` for an alternative implementation of the test in **spatstat**.

Examples

```
# some envelopesfor some idar function
# The argument "savefuns" must be set to "TRUE"
# BEWARE: in real tests nsim shpuld be 199 or higher
data(lansing)
maple.lambda<- density(unmark(lansing[lansing$marks=="maple"]))
multi.maple.sim <- multifocalsimulator(lansing, "maple", nsim=10,
                                simulate=expression(rpoispp(maple.lambda)))

isar.maple.env<- envelope(lansing, fun=isar, mimark="maple", nsim=10,
                                     simulate=multi.maple.sim, r=seq(0.01, 0.25, le=100),
                                       savefuns=TRUE)
# Estimate GoF test
LF.gof(isar.maple.env)
```

`localdar` *Map Local Diversity Area Relationships*

Description

Estimates and maps local diversity-area relationships.

Usage

```
localdar(mippp, mippp.sp = NULL, nx = NULL, ny = NULL, mimark = NULL, idar = "isar",
         buffer = 0, bfw = NULL, r, cross.idar = FALSE, tree = NULL, traits = NULL,
            namesmark = NULL, correct.trait.na = TRUE, correct.trait = "mean",
            correct.phylo="mean")
fdismap(comm, traits)
raoDmap(comm, phy = NULL)
```

Arguments

mippp	A multitype (a.k.a. multivariate) marked point pattern. An object with the ppp format of **spatstat**.
mippp.sp	Univariate point pattern of the focal species. An object with the ppp format of **spatstat**.
nx	Number of points of the grid along the x axis.
ny	Number of points of the grid along the y axis.
mimark	Character. Name of the focal species in the multitype mippp.
idar	Character. The name of the idar function to be computed. Either "isar", "ipsvar", "ipsrar", "ipsear", "ipscar", "icwmar", "icwmar.O", "iraodar"or "imntdar"
buffer	One of "adapt", i.e., compute an adaptive buffer, or a number indicating the width of a fixed buffer area around the plot border
bfw	An owin object indicating the limits of the buffer area.
r	Vector of distances to compute IDAR(r) functions
cross.idar	Logical. If TRUE, the focal pattern will be excluded form the community being measured.
tree	A phylogenetic tree in phylo format (**ape**) or a phylogenetic covariance matrix
traits	A data.frame of traits, or a distance matrix among species (in dist or matrix format) computed on a data.frame of traits.
namesmark	Character. If the marks in mippp are within a data.frame, the name of the column with the species names
correct.trait.na	Logical flag indicating whether NA values in the matrix of traits should be "corrected": NA values will be assigned the mean trait value.
correct.trait	Character. Either "mean" or "exclude". Species missing in the data.frame of traits will be assigned mean trait values or will be excluded from the analysis, respectively.
correct.phylo	Character. Either "mean" meaning *"include missing species in the tree with a constant mean phylogenetic covariance"* or "exclude", meaning *"exclude missing species in the tree from the analysis"*
comm	A comunity data table (sites x species).
phy	A comunity data table (sites x species).

Details

localdar estimates any of the individual diversity area indices (isar, ipsvar,ipsrar, ipsear, ipscar, icwmar, icwmar.O, iraodar or imntdar) at specific locations, such as the locations of trees of a "focal" point partern or at some grid of points (i.e., "mapping" local diversity area relationships). If no predefined focal point pattern is provided (by the arguments mippp.sp or mimark), localdar will, by default, estimate the selected idar function in a 30 x 30 point-grid and return a map (the size of the grid can be modified by the arguments nx and ny). fdismap and raoDmap are internal functions used by localdar to get the individual componentes of iraoD and FDis, instead of the averages obtained by envelope4idar.

Value

If a focal point patternn has been provided by the arguments `mippp.sp` or `mimark`, `localdar` will return a list of marked point patterns (as long as the vector `r`) with the marks showing the estimation of the selected diversity index for the local community defined by a circle of radius *r* around each of the points of the focal pattern. If no focal point pattern is provided, it will return a list of maps (as long as the vector `r`) each with the format im of **spatstat**, each pixel showing the estimation of the selected diversity index for the local community defined by a circle of radius *r* around the pixel center.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

See Also

envelope4idar

Examples

```
# Map of local species area-relationship [ISAR(r)] in San Francisco plot at r=5 and r= 6 m
  data(SF)
  data(SFphylotree)
  data(SFtraits)

  isarSF <- localdar(SF, r=5:6, namesmark="species")

  isarSF

  plot(isarSF[[1]])


# Map of local species area-relationship [ISAR(r)] in San Francisco plot at r=5 and r= 6 m,
# with an adaptive buffer

isarSF <- localdar(SF, r=5:6, namesmark="species",buffer="adapt")



# Map of local species area-relationship [ISAR(r)] in lansing woods at different radii,
# with an fixed  buffer (only for rectangular windows).

data(lansing)
lansing.bfw<- owin(c(0.2,0.8),c(0.2,0.8))
lansing.lsar.bf<-localdar(lansing, r=seq(0.05,0.2,by=0.05), bfw=lansing.bfw)


# Map of local species area-relationship [ISAR(r)] in San Francisco plot at r=5 and r= 6 m,
# with a buffer of 7 m within plot limits

  mibfw<- erosion(SF$win, r=7)
```

```
 isarSF <- localdar(SF, r=5:6, namesmark="species", bfw=mibfw)



# Estimate local species area-relationship [ISAR(r)] in the local communities
#  in circles with radius  r=5 and r= 6 m around the individuals of sp_44

  sp_44_sar <- localdar(SF, r=5:6, namesmark="species", mimark="sp_44")
  sp_44_sar
  plot(sp_44_sar[[1]])
  sp_44_sar[[1]]$marks



# Estimate local species area-relationship [ISAR(r)] in the local communities
#  in circles with radius  r=5 and r= 6 m around the individuals of sp_44
# EXCLUDING the focal species from species counts.

  sp_44_sar <- localdar(SF, r=5:6, namesmark="species", mimark="sp_44", cross.idar=TRUE)
  sp_44_sar
  plot(sp_44_sar[[1]])
  sp_44_sar[[1]]$marks




# Map and estimation of local Phylogenetic Species Variety
local_psvar<-localdar(SF, r=5:6, idar="ipsvar", tree=SFphylotree, namesmark="species")
sp44_psvar <- localdar(SF, r=5:6, idar="ipsvar", tree=SFphylotree, namesmark="species",
                                    mimark="sp_44", buffer="adapt")

# Map and estimation of local Functional Dispersion
local_fdar <- localdar(SF, nx=50, ny=25, r=5:6,  idar="ifdar", traits=SFtraits,
                                  namesmark="species", correct.trait.na=TRUE)
sp44_fdar <- localdar(SF, nx=50, ny=25, r=5:6,  idar="ifdar", traits=SFtraits,
                              namesmark="species", mimark="sp_44", correct.trait.na=TRUE)


# Map of a local community weighted mean of wood density
# first, put the wood desnsity data as a named vector
 wood.density.vec<-unlist(SFtraits[,  "wood.density", drop=FALSE])
    names(wood.density.vec)<- rownames(SFtraits)

local_cwd <- localdar(SF, r=5:6,  idar="icwmar", traits=wood.density.vec,
                                   namesmark="species", correct.trait.na=TRUE)
local_O.cwm <-localdar(SF, r=5:6, idar="icwmar.O", traits=wood.density.vec,
                                      namesmark="species", correct.trait.na=TRUE)

# Map of Rao's phylogenetic diversity
local_rao<- localdar(SF,  r=5:6, idar="iraodar",  tree=SFphylotree, namesmark="species")
local_O.rao <- localdar(SF, r=5:6, idar="iraodar.O",  tree=SFphylotree, namesmark="species")
```

```
# Map of local mean nearest taxon distance
local_mntd <- localdar(SF, r=5:6, idar="imntdar",  tree=SFphylotree, namesmark="species")
```

`midar` *Customize the Individual Diversity-Area Relationship Function*

Description

A wrapper to develop new Individual Diversity-Area Relationship functions on the fly.

Usage

```
midar(mippp, mippp.sp = NULL, mimark = NULL, namesmark = NULL, traits = NULL,
tree = NULL, r = NULL, buffer = 0, bfw = NULL, what = NULL)
```

Arguments

`mippp`	A multitype (a.k.a. multivariate) marked point pattern. An object with the ppp format of **spatstat**.
`mippp.sp`	Univariate point pattern of the focal species. An object with the ppp format of **spatstat**.
`mimark`	Character. Name of the focal species in the multitype `mippp`.
`namesmark`	Character. If the marks in `mippp` are within a `data.frame`, the name of the column with the species names
`buffer`	One of "adapt", i.e., compute an adaptive buffer, or a number indicating the width of a fixed buffer area around the plot border
`bfw`	An owin object indicating the limits of the buffer area.
`r`	Vector of distances to compute IDAR(r) functions
`tree`	A phylogenetic tree in `phylo` format (**ape**) or a phylogenetic covariance matrix
`traits`	A data.frame of traits, or a distance matrix among species (in `dist` or `matrix` format) computed on a data.frame of traits.
`what`	A valid R expression or function that would accept a community matrix (sites x species) and return a unique value

Details

`midar` allows computing new *IDAR*(*r*) functions. The basis of all idar functions are the local communities defined around each point (e.g., each tree) of a focal species (`mimark`) for a certain circular neighborhood of radius *r*. Some diversity measure is computed on each community an the average of all of them is returned as the idar value for this *r*, i.e., as *IDAR*(*r*). The function `midar` applies the R function or expression defined by the argument `what` to each of the "community data tables" (matrices) generated by `mitable` (one for each *r* interval defined by the argument `r`) and return the result as a spatial summary function like all the others in the **idar** package (e.g., `isar` or `pisar`). The R expression or function should accept a community matrix (sites x species) as input and return a unique numeric value.

Value

midar return an object of class fv, see fv.object, which can be plotted directly using plot.fv. Essentially, a data frame containing a column named *r* with the vector of values of the argument r at which the proposed function had been estimated and another column, named *"midar"* which contains an estimate of the selected function.

Simulation envelopes

To compute simulation envelopes for midar functions, use envelope. See the examples in this help page and in multifocalsimulator to know how to compute simulation envelopes from appropriate null models.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

See Also

isar or pisar in this package.

Examples

```
data(SF)
data(SFphylotree)
 # Discard the size mark and keep the species mark in SF ppp:
  sfsp<- ppp(SF$x, SF$y, window=SF$window, marks=SF$marks$species)

# compute "individual Simpsom diversity-area relationship" around sp_44
# using function diversity from package vegan
require(vegan)
simpsom_sp_44<- midar(sfsp, mimark="sp_44", what =function(x) mean(diversity(x, "simpson")),
                                  r=1:15)


# test "individual Simpsom diversity-area relationship" against an inhomogeneous Poisson
# null modellfor sp_44.
  # estimate intensityh surface for sp_44
   lambda<- density.ppp(unmark(sfsp[sfsp$marks=="sp_44"]), positive=TRUE)

  # generate 19 realizations of the null model, keeping all the other species fixed in their
  #  original coordinates.
  simulados<- multifocalsimulator(sfsp, mimark="sp_44", simulate=expression(rpoispp(lambda)),
                      nsim=19, nmin=15)

  # for simplicity define the function that we want to apply to each "local community"
  #  around each tree of sp_44
  mean_simpsom <- function(x) mean(diversity(x, "simpson"))

  # compute envelopes and plot them.
  simpsom_sp_44.env<-envelope(sfsp, midar, mimark="sp_44", what =mean_simpsom ,r=1:15,
```

```
                    nsim=19, simulate=simulados)
  plot(simpsom_sp_44.env)


# compute IPSVAR(r) "by hand"
   # first, check tree as would check it ipsvar
   arbol <- checktree(SFphylotree,  SF, "ipsvar", correct.phylo="exclude")

    # define function to obtain the averge psv from the set of local communities
    #  at each neigborhood radii r
     mipsv <- expression(mean(psv(x, tree=tree, compute.var=FALSE)$PSVs, na.rm=TRUE))
     # compute ipsvar "by hand"
     sp_44_psv<- midar(sfsp, mimark="sp_44", tree=arbol, what =mipsv ,r=1:15)
     plot(sp_44_psv)

# compare it with the result of the built-in function
     plot(ipsvar(sfsp, mimark="sp_44", tree=arbol, r=1:15), add=TRUE, col="blue")
```

`mitable` *Tabulate Marks in Neighbourhood of Every Point in a Point Pattern*

Description

For a sequence of radii defining different neighbourhood sizes, visit each point in a focal point pattern, find the neighbouring points in a target point pattern, and compile a frequency table of the marks of these neighbour points for each radii.

Usage

```
mitable(ppp1,ppp2,r)
```

Arguments

ppp1	"Focal" point pattern (an ppp object of **spatstat**) or a data.frame with columns named "x" and "y"
ppp2	"Target" multivariate point pattern (an ppp object of **spatstat**) or a data.frame with columns named "x", "y", and "marks". The column "marks" wears the (species) name of each point
r	Vector with the sequence of radii ($r>0$) that define different neiborhood sizes.

Details

Given both a "focal" and a "target" point patterns, `mitable` visits each point in the focal point pattern, finds the neighbouring points in the target point pattern, and compile a frequency table of the marks of these neighbour points, where the neibourhood is defined by circles orf radius r around the focal points. From an ecological point of view, it provides a *relevé* of the local community around each focal tree. It repeats the process for each provided r.

Value

A list, with length = *length(r)*. Within each element of the list, a matrix with dimensions *np* X *nsp*, where *np* is the number of points of the focal point pattern and *nsp* is the number of unique species (i.e., unique marks) in the target point pattern. Cell values in the matrix represent number of neighbours of the *j* species for individual *i* at the considered neibourhood size.

Author(s)

Marcelino de la Cruz <marcelino.delacruz@urjc.es>

See Also

marktable for a similar function.

Examples

```
# Compute number of trees of different species  within circles of several  radii r
# around each individual maple tree in lansing woods

data(lansing)

maple<- unmark(lansing[lansing$marks=="maple"])
maple
r= c(0.05, 0.08, 0.1)

counts <- mitable(ppp1=maple, ppp2=lansing, r=r)
dim(counts[[1]])
head(counts[[1]])
```

---

pisar *Phylogenetic Individual Species Area Relationship*

Description

Compute Phylogenetic Individual Species Area Relationship function, i.e., PISAR(r) and its normalized version rISAR(r).

Usage

```
pisar(mippp, mippp.sp = NULL, mimark = NULL, namesmark = NULL, d = NULL, r = NULL,
         buffer = 0, bfw = NULL)
risar(mippp, mippp.sp = NULL, mimark = NULL, namesmark = NULL, d = NULL, d0 = NULL,
        r = NULL, buffer = 0, bfw = NULL)
controldis(d, m, mimark)
```

Arguments

`mippp`	A multitype (a.k.a. multivariate) marked point pattern. An object with the ppp format of **spatstat**.
`mippp.sp`	Univariate point pattern of the focal species. An object with the ppp format of **spatstat**.
`mimark`	Character. Name of the focal species in the multitype `mippp`.
`namesmark`	Character. If the marks in `mippp` are within a `data.frame`, the name of the column with the species names.
`buffer`	One of "adapt", i.e., compute an adaptive buffer, or a number indicating the width of a fixed buffer area around the plot border.
`bfw`	An owin object indicating the limits of the buffer area.
`r`	Vector of distances to compute PIDAR(r) functions.
`d`	A matrix expresing relationships (usually functional or phylogenetic) between species present in the multivariate point pattern.
`d0`	Another matrix expresing relationships (usually functional or phylogenetic) between species present in the multivariate point pattern.
`m`	A community (sites x species) data table.

Details

The original definition of *ISAR*(*r*) (Wiegand et al. 2007) was reformulated as:

$$ISAR_f(r) = \sum_{m=1}^{S} \delta_{fm} D_{fm}(r)$$

(Wiegand and Moloney 2014; Wang et al. 2016), where $D_{fm}(r)$ describe the probabilities that the nearest species *m* neighbor of the typical individual of the focal species *f* is located within distance *r*, and δ_{fm} yields a value of zero if *f* = *m* and a value of one otherwise (note that in their original proposal ISAR was formulated as if the value assigned to δ_{fm} were 1 for all species pairs, including *f* = *m*). Based in this re-formulation, they defined the *Phylogenetic Individual Species Area Relationship*, i.e., *PISAR*(*r*), as:

$$PISAR_f(r) = \sum_{m=1}^{S} \delta_{fm}^{phy} D_{fm}(r)$$

where δ_{fm}^{phy} is an index of phylogenetic (or functional) dissimilarity between species *f* and *m*. *PISAR*(*r*) quantifies the expected phylogenetic (or functional) diversity of species within the neighborhood with radius *r* around the typical individual of the focal species *f*.

They also defined *rISAR*(*r*) as a a function that is independent of local species richness within the neighborhood *r*; for this, they divided the PISAR function by the ISAR function:

$$rISAR_f(r) = \frac{\sum_{m=1}^{S} \delta_{fm}^{phy} D_{fm}(r)}{\sum_{m=1}^{S} \delta_{fm} D_{fm}(r)}$$

If the placement of the focal species f is unrelated with functional or phylogenetic relationships with their neighbors, the $rISAR_f(r)$ will approximate the mean pairwise functional (or phylogenetic) dissimilarity $\Delta_f^P = \sum_m \delta_{fm}^{phy}/(S-1)$ between an individual of the focal species f and all other species in the plot.

The function `controldis` controls that the order of species in the phylogenetic distance matrix matches the order of species amomng the levels of species marks in the point pattern, and extracts the vector of distances from all species to the focal one (`mimark`).

Value

`pisar` and `risar` return an object of class `fv`, see fv.object, which can be plotted directly using plot.fv.

Essentially a data frame containing a column named *r* with the vector of values of the argument r at which the PISAR(r) or rISAR(r) function had been estimated and another column, named *"risar"* or *"pidar"*,which contains an estimate of the selected function.

`controldis` returns either the vector of distances between the focal and the rest of species or a vector of 1's if there is not phyloegenetic distance provided.

Simulation envelopes

To compute simulation envelopes for `pisar` or `risar` functions, use `envelope`. See the examples in this help page and in `multifocalsimulator` to know how to compute simulation envelopes from appropriate null models.

NOTE

When computing risar it is necessary to provide a phylogenetic or functional distance matrix to the argument `d`. By default, argument `d0` will be set to a vector of 1's. It is however possible to provide a different matrix to `d0` and compute instead, e.g., a ratio of phylogenetic to functional diversity.

Author(s)

Marcelino de la Cruz `<marcelino.delacruz@urjc.es>`

References

Wang, X., et al. (2016). Stochastic dilution effects weaken deterministic effects of niche-based processes in species rich forests. *Ecology, in press*

Wiegand,T., Gunatilleke, C.V.S., Gunatilleke, I.A.U.N. and Huth, A. (2007) How individual species structure diversity in tropical forests. *PNAS* 104, 19029-19033.

Wiegand, T., and K.A. Moloney. (2014). *A handbook of spatial point pattern analysis in ecology*. Chapman and Hall/CRC press, Boca Raton, FL

See Also

See also `isar` for other individual diversity area functions.

Examples

```
data(SF)
data(SFphylotree)

# Discard the size mark and keep the species mark in SF ppp:
sfsp<- ppp(SF$x, SF$y, window=SF$window, marks=SF$marks$species)

# compute phylogenetic distance among species
dphy <- cophenetic(SFphylotree)

# compute and plot PISAR function for sp_44
pisar_44 <- pisar(sfsp, mimark="sp_44", r=1:15,  d=dphy)
plot(pisar_44)

## Not run:

# Compute rISAR and plot envelopes for an inhomogeneous Poisson model
#  of each species in San Francisco plot
# BEWARE: THIS TAKES QUITE A FEW MINUTES !!!

# Split sfsp point pattern ppp by species
sfsp.sp<- split(sfsp)

#  Species with >= 10 individuals
sfsp10 <- sapply(sfsp.sp, function(x) x$n>=10)
#names of those species
nombressf<- names(sfsp10[sfsp10])

# parameters for the simulations, estimation of intensity, etc.
nsim<-199
nmin<-10
sigma <- 8
r<- seq(1,15, by=0.5)

# list to store results
risar.sf<- list()
# start computation
for( sp in nombressf){
   print(sp)
   # estimate intensity of the focal species
   lambda<- density(unmark(sfsp[sfsp$marks==sp]), sigma=sigma, positive=TRUE)
   # obtain simulated patterns were all species ecept the focal remain fixed
   #  and the focal varies according to an inhomomgeneous Poiisson process
   simulados<- multifocalsimulator(sfsp, mimark=sp,
               simulate=expression(rpoispp(lambda)), nsim=nsim,nmin=nmin)
   # compute risar
   risar.sf[[sp]] <- envelope(sfsp, risar, mimark=sp, d=dphy, r=r,
        simulate=simulados,nsim=nsim, savefuns=T, buffer=0)
}

# plot the results
```

```
dev.new(height=7, width=16)
par(mfrow=c(3,9))
for(i in 1:27) plot(risar.sf[[i]], legend=F, main=nombressf[i])


## End(Not run)
```

---

proportion.idar *Determine the Proportion of Accumulator and Repeller Species*

Description

This function determines the proportion of species in a community that "accumulate" or "repell" diversity, in the sense of Wiegand et al. (2007).

Usage

```
  proportion.idar(envlist, alfa = 0.05)
     ## S3 method for class 'pidar'
plot(x,cols=c(1,2,3),type=c("l","o","o"), pch=c(NA,19,19),
    lty=c(1,1,1),  legend=TRUE, p.legend="topleft",...)
```

Arguments

envlist	A list whose elements are the result of applying envelope to a set of point patterns.
alfa	alpha value to calculate deviations from null model
x	The result of proportion.idar
cols	A vector (length =3) with the color names or codes for the curves of each of the 3 categories: "neutral", "accumulator", "repeller".
type	A vector (length =3), with the type of plot for each category. See plot.default.
pch	A vector (length =3), with the point type for each category. See points
lty	A vector (length =3), with the line type for each category. See par
legend	Logical. Should a legend be added to the plot?
p.legend	Position of the legend. It can be a vector wit the xycoordinates of the upperleft corner of the legend box or a keyword accepted by legend
...	Additional graphical parameters passed both to funtions plot and lines

Details

This function determines the proportion of accumulator, repeller and neutral species at each scale *r*, following the approach of Wiegand et al (2007). A species is classified as an accumulator at scale *r* if there are less than $(nsim + 1) * alpha/2$ simulated values greater than the observed *idar(r)*. On the contrary, a species is classified as repeller at scale if there are less than $(nsim + 1) * alpha/2$ simulated values smaller than the observed *idar(r)*. The percentage is computed over the total of species in `envlist`. It is necessary that the objects in envlist (i.e. the envelope objects) had been computed with the argument "`simfuns=TRUE`".

Value

`proportion.idar` produces an object of class "`pidar`", basically a list with components

percentage A `data.frame` with 3 columns ("p.accumulators", "p.repellers", "p.neutrals") indicating the percentage in each category for each *r* (rows)

nsp The total number of species for which the percentage has been computed.

nsim The number of simulations

alfa The alpha value employed to calculate the deviations from null model and the assignation to each of the categories

r The vector of r values at which the idar(r) functions have been estimated

behaviour A data.frame with the behaviour ("A"= accumulator, "R" = repeller) of each species at each radius r

`plot.pidar` plots the result.

Author(s)

Marcelino de la Cruz <`marcelino.delacruz@urjc.es`>

References

Wiegand,T., Gunatilleke, C.V.S., Gunatilleke, I.A.U.N. and Huth, A. (2007) How individual species structure diversity in tropical forests. *PNAS* 104, 19029-19033.

Examples

```
# Compute percentage of accumulator, neutral an repeller species (for ISAR) in Lansing woods
# In this example, against a null model of IPP.
data(lansing)

# Compute rISAR and plot envelopes for an inhomogeneous Poisson model
# of each species in San Francisco plot
# Split sfsp point pattern ppp by species
lansing.sp<- split(lansing)

# Set parameters for the simulations, estimation of intensity surface for IPP, etc.
# BEWARE: THIS is for R-TESTING ALONE. REAL TESTS SHOULD EMPLOY 199 simulations or higher
nsim<-8
```

```
r<- seq(0.01,0.25, by=0.01)
# Create list to store the results
isar.lansing<- list()
# start computation

for( i in 1: length(lansing.sp)){
   print(i)
   # estimate intensity of the focal species
   lambda<- density(lansing.sp[[i]])
   # obtain simulated patterns were all species except the focal remain fixed
   # and the focal varies according to an inhomomgeneous Poiisson process
   simulados<- multifocalsimulator(lansing, mimark=levels(lansing$marks)[i],
                            simulate=expression(rpoispp(lambda)), nsim=nsim)
  # compute isar and envelopes for each species
  # It is COMPULSORY that the argument "savefuns" it is set to TRUE
   isar.lansing[[i]] <- envelope(lansing, isar, mimark=levels(lansing$marks)[i],  r=r,
         simulate=simulados,nsim=nsim, savefuns=TRUE, buffer=0)
}


prop.isar.lansing <- proportion.idar(isar.lansing)
head(prop.isar.lansing $percentage)
head(prop.isar.lansing $behaviour)
plot(prop.isar.lansing, p.legend=c(0.15,60))

## Not run:

# Compute percentage of accumulator, neutral an repeller species (for ISAR) in San Francisco
# forest.  In this example, against a null model of IPP.
data(SF)

# Discard the size mark and keep the species mark in SF ppp:
sfsp<- ppp(SF$x, SF$y, window=SF$window, marks=SF$marks$species)

# Compute ISAR and plot envelopes for an inhomogeneous Poisson model
# of each species in San Francisco plot
# Split sfsp point pattern ppp by species
sfsp.sp<- split(sfsp)
# Select species with >= 10 individuals (to get some statistical power)
sfsp10 <- sapply(sfsp.sp, function(x) x$n>=10)
#Get names of those species
nombressf<- names(sfsp10[sfsp10])
# Set parameters for the simulations, estimation of intensity surface for IPP, etc.
nsim<-199
nmin<-10
sigma <- 8
r<- seq(1,15, by=0.5)
# Create list to store the results
isar.sf<- list()
# start computation
# BEWARE: THIS TAKES QUITE A FEW MINUTES!!!
for( sp in nombressf){
   print(sp)
```

```
   # estimate intensity of the focal species
   lambda<- density(unmark(sfsp[sfsp$marks==sp]), sigma=sigma, positive=TRUE)
   # obtain simulated patterns were all species except the focal remain fixed
   # and the focal varies according to an inhomomgeneous Poiisson process
   simulados<- multifocalsimulator(sfsp, mimark=sp,
                          simulate=expression(rpoispp(lambda)), nsim=nsim,nmin=nmin)
  # compute isar and envelopes for each species
  # It is COMPULSORY that the argument "savefuns" it is set to TRUE
   isar.sf[[sp]] <- envelope(sfsp, isar, mimark=sp,  r=r,
         simulate=simulados,nsim=nsim, savefuns=TRUE, buffer=0)
}


prop.isar.SF <- proportion.idar(isar.sf)
head(prop.isar.SF$percentage)
head(prop.isar.SF$behaviour)
plot(prop.isar.SF)


## End(Not run)
```

SF *San Francisco forest plot.*

Description

Point pattern describing the locations of 822 trees (belonging to 113 species) in the San Francisco forest (southern Ecuador). Accompanied by a phylogenetic tree and a file with wood density of some of the species.

Usage

```
data(SF)
data(SFphylotree)
data(SFtraits)
```

Format

SF is an object of class ppp of **spatstat** representing the point pattern of trees locations, with a data.frame of marks. See ppp.object for details of the format.The dataset has 822 points with the following marks:

species Species to which each tree belongs

dbh Diameter at breast height of each tree

SFphylotree is a phylogenetic tree of the class phylo of **ape**, showing the phylogenetic relationshios among 296 tree species.

SFtraits is a data.frame with just one column, giving the estimated wood density for some of the 296 tree species.

Details

This dataset represents the locations of trees with *dbh* >= 5 cm in a polygonal plot of approx. 100 x 70 m in the montane forest near to Reserva Biológica San Francisco (Zamora-Chinchipe, southern Ecuador). These are part of the data collected by Vicuña (2016) and other Ecuadorian botanists and have been analyzed several times (e.g., Chacón et al. 2014). The coordinates of trees are given in meters and the *dbh* in cm. The phylogenetic tree has been extracted from Phylomatic and calibrated with the BLADJ algorithm of Phylocom. The data about wood density have been compiled from several sources and is expressed in g/cm^3.

Source

Vicuña, R. (2016) *Estructura espacial y dinámica del bosque montano del Sur del Ecuador. Interacciones bióticas y limitaciones abióticas*. Tesis Doctoral. Universidad Politécnica de Madrid.

References

Chacón-Labella et al. (2014) Negative density dependence and environmental heterogeneity effects on tree ferns across succession in a tropical montane forest. *Perspectives in Plant Ecology, Evolution and Systematics* 16(2): 52-63.

Index

∗ **datasets**
SF, 32
∗ **spatial**
checktree, 2
envelope4idar, 3
fdis, 8
ipsim, 9
isar, 13
LF.gof, 16
localdar, 18
midar, 22
mitable, 24
pisar, 25
proportion.idar, 29

as.matrix, *8*

checktraits *(*checktree*)*, 2
checktree, 2
controldis *(*pisar*)*, 25

data.frame, *2*, *4*, *13*, *19*, *22*, *24*
dclf.test, *18*
density.ppp, *10*
dist, *2*, *4*, *8*, *13*, *19*, *22*

envelope, *5*, *14*, *17*, *23*, *27*, *29*
envelope4idar, 3, *14*, *19, 20*

fdis, 8
fdismap *(*localdar*)*, 18
fdisp, *6*, *9*, *15*
fv, *17*
fv.object, *5*, *14*, *23*, *27*

gowdis, *8*

idar2 *(*envelope4idar*)*, 3
ifdar, *5*, *9*
ifdar *(*isar*)*, 13
im, *20*
ipscar *(*isar*)*, 13
ipsear *(*isar*)*, 13
ipsim, 9, *14*
ipsimlist *(*ipsim*)*, 9
ipsrar *(*isar*)*, 13
ipsvar, *5*
ipsvar *(*isar*)*, 13
isar, 13, *22, 23*, *27*

legend, *29*
LF.gof, 16
localdar, 18

marktable, *25*
matrix, *2*, *4*, *8*, *13*, *19*, *22*
midar, 22
mitable, *22*, 24
multifocalsimulator, *23*, *27*
multifocalsimulator *(*ipsim*)*, 9

owin, *4*, *13*, *19*, *22*, *26*

par, *29*
phylo, *32*
pisar, *22, 23*, 25
plot.default, *29*
plot.fv, *5*, *14*, *23*, *27*
plot.pidar *(*proportion.idar*)*, 29
points, *29*
ppp, *2, 3*, *9, 10*, *13*, *19*, *22*, *24*, *26*
ppp.object, *32*
proportion.idar, 29
psd, *6*, *15*

raoD, *5*
raoDmap *(*localdar*)*, 18
raoDmod *(*envelope4idar*)*, 3
risar *(*pisar*)*, 25
rpoispp, *10*

SF, 32

SFphylotree *(SF)*, 32
SFtraits *(SF)*, 32
simulador2, *3*
simulador2 *(ipsim)*, 9