Package ‘bamm’

December 20, 2022

Type    Package
Title   Species Distribution Models as a Function of Biotic, Abiotic and Movement Factors (BAM)
Version 0.4.3
Maintainer Luis Osorio-Olvera <luismurao@gmail.com>
URL     https://luismurao.github.io/bamm/
BugReports https://github.com/luismurao/bamm/issues

Description Species Distribution Modeling (SDM) is a practical methodology that aims to estimate the area of distribution of a species. However, most of the work has focused on estimating static expressions of the correlation between environmental variables. The outputs of correlative species distribution models can be interpreted as maps of the suitable environment for a species but not generally as maps of its actual distribution.
Soberón and Peterson (2005) <doi:10.17161/bi.v2i0.4> presented the BAM scheme, a heuristic framework that states that the occupied area of a species occurs on sites that have been accessible through dispersal (M) and have both favorable biotic (B) and abiotic conditions (A). The 'bamm' package implements classes and functions to operate on each element of the BAM and by using a cellular automata model where the occupied area of a species at time t is estimated by the multiplication of three binary matrices: one matrix represents movements (M), another abiotic -niche- tolerances (A), and a third, biotic interactions (B). The theoretical background of the package can be found in Soberón and Osorio-Olvera (2022) <arXiv:2212.06308>.

SystemRequirements C++11, GDAL (>= 2.2.3): gdal-bin (deb), libgdal-dev (deb) or gdal-devel (rpm), GEOS (>= 3.4.0), PROJ (>= 4.9.3): libproj-dev (deb), sqlite3, ImageMagick++: imagemagick (deb), libmagic++-dev (deb) or ImageMagick-c++-devel (rpm) ImageMagick (http://imagemagick.org) or GraphicsMagick (http://www.graphicsmagick.org) or LyX (http://www.lyx.org) for saveGIF(); (PDF)LaTeX for saveLatex(); SWF Tools
(http://swftools.org) for saveSWF(); FFmpeg (http://ffmpeg.org) or avconv (https://libav.org/avconv.html) for saveVideo()

**NeedsCompilation**  yes

**License**  GPL (>= 3)

**Encoding**  UTF-8

**Depends**  R (>= 3.5.0)

**Imports**  raster (>= 3.4-13), Matrix (>= 1.2.14), RSpectra (>= 0.13.1), methods (>= 3.3), magrittr (>= 1.2), dplyr (>= 0.8.0), purrr (>= 0.2), igraph (>= 1.2), leaflet (>= 2.0), Rdpack (>= 0.11.0), animation (>= 2.3), future (>= 1.18.0), furrr (>= 0.1.0), Rcpp (>= 0.12.18), sp(>= 1.3.0)

**RdMacros**  Rdpack

**RoxygenNote**  7.2.2

**Suggests**  knitr, crosstalk, plotly, rmarkdown, testthat (>= 3.0.0), covr

**VignetteBuilder**  knitr

**LinkingTo**  Rcpp, RcppArmadillo

**Config/testthat/edition**  3

**Author**  Luis Osorio-Olvera [aut, cre] (<https://orcid.org/0000-0003-0701-5398>), Jorge Soberón [aut] (<https://orcid.org/0000-0003-2160-4148>)

**Repository**  CRAN

**Date/Publication**  2022-12-20 11:10:05 UTC

**R topics documented:**

.nonzero .............................................. 3
adj_mat ................................................ 4
bam-class ............................................. 5
bam_clusters .......................................... 6
bam_sim ................................................ 7
bam_ssim .............................................. 9
bioindex-class ...................................... 11
bioindex_sparse-class ............................... 11
community_sim ...................................... 12
community_sim-class ............................... 13
csd-class ........................................... 14
csd_estimate ......................................... 14
csim2pam ............................................ 15
diversity_range-class .............................. 16
diversity_range_analysis ......................... 17
eigen_bam ........................................... 19
g_area-class ......................................... 20
jaccard ................................................ 21
Helper function to compute the elements in g0 that have no zero values. The function is taken from the Ringo package

**Description**

Helper function to compute the elements in g0 that have no zero values. The function is taken from the Ringo package.

**Usage**

`.nonzero(x)`

**Arguments**

- `x` A matrix of class "dgCMatrix"
adj_mat

adj_mat: Function to compute the adjacency matrix of an area.

Description

Creates an adjacency matrix of an area of interest. This could be the accessible area (M) of a species or any geographic region of interest.

Usage

adj_mat(modelsparse, ngbs = 1, eigen_sys = FALSE, which_eigs = 1)

Arguments

- modelsparse: A setA object returned by the function model2sparse.
- ngbs: Numeric. Number of neighbors (see details).
- eigen_sys: Logical. If TRUE the eigen analyses of the adjacency matrix will be returned.
- which_eigs: Numeric. Which eigen value and eigen vector will be returned.

Details

The model is a raster object of the area where the dispersal process will occur. The number of neighbors depends on the dispersal abilities of the species and the spatial resolution of the niche model; for example, a species's with big dispersal abilities will move throughout more than 1 km^2 per day, so the idea is to give an approximate number of moving neighbors (pixels) per unit of time. For more information about see adjacency matrices in the context of the theory of area of distribution (Soberon and Osorio-Olvera, 2022).

Value

Returns an object of class setM with 7 slots. The first contains the adjacency matrix. A n x n sparse matrix (n=number of non-NA cells of the niche model) where connected cells are represented by 1. The second slot has the adjacency list. It is a list of matrices with four columns (FromRasCell -from cell ID of the raster-, -to cell ID of the raster-, -from non-NA cell-, -to non-NA cell-). Other slots contain information about initial coordinates where dispersal occurs (initial_points), number of cells used to define the neighborhood (ngbs), non-NA coordinates (coordinates), and a matrix of eigen vectors (eigen_vec).

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References

**Examples**

```r
x_coord <- c(-106.5699, -111.3737, -113.9332, -110.8913, -106.4262, -106.5699)
xy <- cbind(x_coord, y_coord)
p <- sp::Polygon(xy)
ps <- sp::Polygons(list(p),1)
sps <- sp::SpatialPolygons(list(ps))
mx_grid <- bamm::shape2Grid(sps,resolution = 0.25,ones = TRUE)
mx_sparse <- bamm::model2sparse(model=mx_grid, threshold = 0.1)
adj_mx <- bamm::adj_mat(modelsparse=mx_sparse,
                         ngbs=1,eigen_sys=TRUE,which_eigs=1)
print(adj_mx)
mx_grid_eigen <- mx_grid
mx_grid_eigen[mx_sparse@cellIDs] <- adj_mx@eigen_vec
raster::plot(mx_grid_eigen)
```

---

**Description**

Class bam digram

**Value**

An object of class bam

**Slots**

- `sdm_sim` A list of sparse vectors representing the area occupied
- `palatable_matrices` A list of sparse vectors representing palatable sites.
- `sim_steps` Number of simulation steps by the species

**Author(s)**

Luis Osorio-Olvera & Jorge Soberón
bam_clusters  

Function to estimate the connectivity of suitable areas given an adjacency matrix.

Usage

```r
bam_clusters(model, ngbs = 1, plot_model = FALSE)
```

Arguments

- `model`: A niche model in raster format or a `setA` object (see `model2sparse`).
- `ngbs`: Numeric. Number of neighbors (see details).
- `plot_model`: Logical. Indicates whether to plot the niche model using a leaflet map, connected suitable cells shown in the same color.

Details

The main result of the function is the Connectivity-Suitability-Diagram (CSD). In this diagram connected suitable cells make clusters of pixels. For more details about the CSD see (Soberon and Osorio-Olvera, 2022).

Value

An object of class `csd`. It contains three slots. 1) `connections`: a data.frame with three columns where first and the second represent (x and y) centroid coordinates of the niche model and the third column with the cluster ID where they belong. 2) `interactive_map`: a leaflet map of connected suitable pixels shown in the same color. 3) A RasterLayer of connected suitable pixels.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

```r
set.seed(891)
model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
model <- model > 0.7
clusterin <- bamm::bam_clusters(model, ngbs=1, plot_model=TRUE)
```
bam_sim

raster::plot(clusterin@raster_map)
clusterin@interactive_map

bam_sim

bam_sim: Simulate dispersal dynamics using the set B of the BAM framework.

description

bam_sim: Simulate dispersal dynamics using the set B of the BAM framework.

usage

bam_sim(
  sp1,
  sp2,
  set_M,
  initial_points,
  periods_toxic,
  periods_suitable,
  nsteps,
  progress_bar = TRUE
)

arguments

sp1 Niche model of the focal species (the one that disperses).
sp2 Niche model of the species with whom sp1 interacts (currently no dispersal dynamics for this species).
set_M A setM object containing the adjacency matrix for sp1. See adj_mat
initial_points A sparse vector returned by the function occs2sparse
periods_toxic Time periods that sp2 takes to develop defense mechanisms (i.e. toxic).
periods_suitable This is the time that sp2 takes to become non-toxic
nsteps Number of steps to run the simulation
progress_bar Show progress bar

details

The returned object inherits from setA, setM classes. Details about the dynamic model can be found in Soberon and Osorio-Olvera (2022). The model is cellular automata where the occupied area of a species at time \( t + 1 \) is estimated by the multiplication of three binary matrices: one matrix represents movements (M), another abiotic -niche- tolerances (A), and a third, biotic interactions (B) (Soberon and Osorio-Olvera, 2022).

\[
G_j(t + 1) = B_j(t)A_j(t)M_jG_j(t)
\]
Value

An object of class bam. The object contains 12 slots of information (see details) from which simulation results are stored in sdm_sim object, a list of sparse matrices with results of each simulation step.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

```r
# Compute dispersal dynamics of Urania boisduvalii as a function of
# palatable Omphalea
urap <- system.file("extdata/urania_omph/urania_guanahacabibes.tif",
                  package = "bamm")
ura <- raster::raster(urap)
ompp <- system.file("extdata/urania_omph/omphalea_guanahacabibes.tif",
                    package = "bamm")
omp <- raster::raster(ompp)
msparse <- bamm::model2sparse(ura)
init_coordsdf <- data.frame(x=-84.38751, y= 22.02932)
initial_points <- bamm::occs2sparse(modelsparse = msparse, init_coordsdf)
set_M <- bamm::adj_mat(modelsparse = msparse, ngbs = 1)
ura_sim <- bamm::bam_sim(sp1=ura, sp2=omp, set_M=set_M,
                      initial_points=initial_points,
                      periods_toxic=5,
                      periods_suitable=1,
                      nsteps=40)
ura_omp <- bamm::sim2Raster(ura_sim)
raster::plot(ura_omp[[c(1,5,10,15,20,30,35,40)]]

if(requireNamespace("animation")){
  # Animation example
  anp <- tempfile(pattern = "simulation_results_.",fileext = ".gif")
  new_sim <- bamm::sim2Animation(sdm_simul = ura_sim,
                       which_steps = seq_len(ura_sim@sim_steps),
                       fmt = "GIF",
                       filename = anp)
}
```
bam_ssim

bam_ssim: Simulate dispersal dynamics using the set B of the BAM framework.

Description

bam_ssim: Simulate dispersal dynamics using the set B of the BAM framework.

Usage

bam_ssim(
  sp1,
  sp2,
  set_M,
  initial_points,
  periods_toxic,
  periods_suitable,
  dispersal_prob = 0.85,
  palatable_matrices = FALSE,
  nsteps,
  progress_bar = TRUE
)

Arguments

sp1 Niche model of the focal species (the one that disperses).
sp2 Niche model of the species with whom sp1 interacts (currently no dispersal dynamics for this species).
set_M A setM object containing the adjacency matrix for sp1. See adj_mat
initial_points A sparse vector returned by the function occs2sparse
periods_toxic Time periods that sp2 takes to develop defense mechanisms (i.e. toxic).
periods_suitable This is the time that sp2 takes to become non-toxic
dispersal_prob A numeric value indicating the probability to disperse to neighboring cells. This probability is assumed to be binomially distributed
palatable_matrices Logical. If TRUE palatable matrices for each time will be returned.
nsteps Number of steps to run the simulation
progress_bar Show progress bar

Details

The returned object inherits from setA, setM classes. Details about the dynamic model can be found in Soberon and Osorio-Olvera (2022).
**Value**

An object of class bam. The object contains 12 slots of information (see details) from which simulation results are stored in sdm_sim object, a list of sparse matrices with results of each simulation step. Palatable matrices are returned as a list of sparse matrices with information about palatable pixels for each step of the simulation.

**Author(s)**

Luis Osorio-Olvera & Jorge Soberón

**References**


**Examples**

```r
urap <- system.file("extdata/urania_omph/urania_guanahacabibes.tif",
                    package = "bamm")
ura <- raster::raster(urap)
ompp <- system.file("extdata/urania_omph/omphalea_guanahacabibes.tif",
                    package = "bamm")
omp <- raster::raster(ompp)
msparse <- bamm::model2sparse(ura)
init_coordsdf <- data.frame(x=-84.38751, y= 22.02932)
initial_points <- bamm::occs2sparse(modelsparse = msparse,init_coordsdf)
set_M <- bamm::adj_mat(modelsparse = msparse,ngbs = 1)
ura_ssim <- bamm::bam_ssim(sp1=ura, sp2=omp, set_M=set_M,
                          dispersal_prob = 0.75,
                          initial_points=initial_points,
                          periods_toxic=5,
                          periods_suitable=1,
                          nsteps=40)
ura_omp <- bamm::sim2Raster(ura_ssim)
raster::plot(ura_omp[[c(1,2,5,10,15,20,30,35,40)]]

if(requireNamespace("animation")){
  # Animation example
  anp <- tempfile(pattern = "simulation_results_",fileext = ".gif")
  new_sim <- bamm::sim2Animation(sdm_simul = ura_ssim,
                                which_steps = seq_len(ura_ssim@sim_steps),
                                fmt = "GIF",
                                filename = anp)
}
```
bioindex-class

Class bioindex

Description
Class bioindex

Value
An object of class bioindex

Slots
  alpha  A matrix with the richness of species per site
  omega  A matrix with the range size of every species
  dispersion_field  A matrix with the set of ranges of all species that occur in at each locality

Author(s)
Luis Osorio-Olvera & Jorge Soberón

bioindex_sparse-class

Class bioindex_sparse

Description
Class bioindex_sparse

Value
An object of class bioindex_sparse

Slots
  alpha  A sparse matrix with the richness of species per site
  omega  A sparse matrix with the range size of every species
  dispersion_field  A sparse matrix with the set of ranges of all species that occur in at each locality

Author(s)
Luis Osorio-Olvera & Jorge Soberón
community_sim

Description

Estimate community dynamics using the bamm framework

Usage

```r
community_sim(
  en_models,
  ngbs_vect,
  init_coords,
  nsteps,
  threshold_vec = NULL,
  stochastic_dispersal = FALSE,
  disp_prop2_suitability = TRUE,
  disper_prop = 0.5
)
```

Arguments

- `en_models`: A stack or directory with the ecological niche models for each species in the community.
- `ngbs_vect`: A vector containing the number of neighbors for each adjacency matrix of each species in the community see `adj_mat`.
- `init_coords`: A data.frame with 3 columns: sp_name, x and y; x is the longitude and y is the latitude of initial dispersal points.
- `nsteps`: Number of iteration steps for the simulation.
- `threshold_vec`: A vector of threshold values used to binarize niche models.
- `stochastic_dispersal`: Logical. If dispersal depends on a probability of visiting neighbor cells (Moore neighborhood).
- `disp_prop2_suitability`: Logical. If probability of dispersal is proportional to the suitability of reachable cells. The proportional value must be declared in the parameter ‘disper_prop’.
- `disper_prop`: Probability of dispersal to reachable cells.

Details

Each element in `community_sim` is an object of class. For more details about the simulation see `sdm_sim`, `bam`.

Value

An object of class `community_sim`. The object contains simulation results for each species in the community.
Author(s)
Luis Osorio-Olvera & Jorge Soberon

References

Examples

```r
lagos_path <- system.file("extdata/conejos", 
    package = "bamm")
enm_path <- list.files(lagos_path, 
    pattern = ".tif", 
    full.names = TRUE)[seq(1,10)]
en_models <- raster::stack(enm_path) 
ngbs_vect <- sample(1:2,replace = TRUE, 
    size = raster::nlayers(en_models))
init_coords <- read.csv(file.path(lagos_path, 
    "lagos_initit.csv"))[seq(1,10),]
nsteps <- 12
sdm_comm <- bamm::community_sim(en_models = en_models, 
    ngbs_vect = ngbs_vect, 
    init_coords = init_coords, 
    nsteps = nsteps)
com_pam <- bamm::csim2pam(sdm_comm, which_steps = seq(1,nsteps)) 
rich_pam <- pam2richness(com_pam, which_steps = c(1,5,10)) 
raster::plot(rich_pam)
```

community_sim-class

Class community_sim digram

Description
Class community_sim digram

Value
An object of class community_sim

Slots
community_sim A list of sparse vectors representing the area occupied by the species

Author(s)
Luis Osorio-Olvera & Jorge Soberón
## csd-class

*Class csd*

### Description

Class csd

### Value

An object of class csd

### Slots

- **connections** A data.frame with four columns: x, y, clusterID and cluster_size
- **interactive_map** A leaflet map with markers showing the geographical clusters
- **raster_map** A raster map with cluster IDs as values.

### Author(s)

Luis Osorio-Olvera & Jorge Soberón

---

## csd_estimate

*csd_estimate: Estimate the connectivity suitability and dispersal plot*

### Description

csd_plot gives an estimate of the number of geographic clusters given a set of dispersal hypothesis and a suitability raster

### Usage

`csd_estimate(model, dispersal_steps = c(2, 4, 8, 16, 32, 64))`

### Arguments

- **model** A raster model or a setA object representing the suitability model
- **dispersal_steps** A numeric vector with elements representing the dispersal hypothesis to test.

### Details

For more information about the Connectivity-Suitability-Diagram see [bam_clusters](#).
Value

A list of length three. The first element contains the Connectivity-Suitability-Diagram information estimated for each element in the vector of dispersal steps. The second is tbl_df object with a summary of the number of cluster of each dispersal step and the mean number of connected clusters. The last element is base plot showing the information contained in the tbl_df object.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

```r
model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
model <- model > 0.7
csd_plot <- bamm::csd_estimate(model, dispersal_steps=c(2,4,8))
csd_plot$plot
```

csim2pam

**csim2pam**: Converts community simulation to a Presence Absence Matrix (PAM)

Description

Converts community simulation object into a Presence Absence Matrices (PAM) for a given simulation steps.

Usage

csim2pam(community_sim, which_steps)

Arguments

- community_sim: An object of class `community_bam`.
- which_steps: Steps in the simulation object to be converted into a PAM

Details

For details about the object `community_sim` see `community_sim`
Value

An object of class `pam`; it contains five slots. 1) `pams`: a list of sparse matrices with Presence- Absence information (PAMs). 2) `which_steps`: time steps corresponding to each PAM. 3) `sp_names`: a vector of species names. 4) the grid area used in the simulation. 5) Non NA cell (pixel) IDs.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

```r
lagos_path <- system.file("extdata/conejos",  
    package = "bamm")
enm_path <- list.files(lagos_path,  
    pattern = "\.tif",  
    full.names = TRUE)[seq(1,10)]
en_models <- raster::stack(enm_path)
ngbs_vect <- sample(1:2,replace = TRUE,  
    size = raster::nlayers(en_models))
init_coords <- read.csv(file.path(lagos_path,  
    "lagos_initit.csv"))[seq(1,10),]
nsteps <- 10
sdm_comm <- bamm::community_sim(en_models = en_models,  
    ngbs_vect = ngbs_vect,  
    init_coords = init_coords,  
    nsteps = nsteps,  
    threshold = 0.1)
pamt10 <- bamm::csim2pam(community_sim = sdm_comm ,  
    which_steps = 10)
pams <- bamm::csim2pam(community_sim = sdm_comm ,  
    which_steps = seq_len(10))
rich_pam <- bamm::pam2richness(pams,which_steps = c(1,5))
print(rich_pam)
```
diversity_range_analysis

Value

An object of class diversity_range

Slots

alpha A column vector with species richness per site
omega A column vector with the size of the area of distribution per species.
alpha_raster Species richness in raster format.
dispersion_field A matrix with the set of ranges of all species that occur in at each locality.
dispersion_field_raster Raster object with the observed values of dispersion field.
diversity_range_raster Raster object of diversity range.
diversity_range_colors Colors to plot endemism levels.
null dispersion_field_dist A matrix with dispersion field null distribution.
xy_coordinates A matrix of geographical coordinates
n_iterations Number of iterations used to estimate the dispersion field null distribution.
nsps Number of species in the PAM.
nsites Number of sites in the PAM.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

diversity_range_analysis
range_diversity_analysis: diversity analysis

description

diversity_range_analysis biodiversity indices related to diversity-range plots

Usage

diversity_range_analysis(
  pam,
  xy_mat = NULL,
  lower_interval = 0.05,
  upper_interval = 0.95,
  raster_templete = NULL,
  niter = 100,
  return null dfield = FALSE,
  parallel = TRUE,
  n_cores = 2
)

Arguments

- **pam**: A Presence-Absence-Matrix of matrix class or sparse matrix.
- **xy_mat**: A two dimensional matrix with longitude and latitude data.
- **lower_interval**: Lower interval.
- **upper_interval**: Upper interval.
- **raster_template**: A raster template.
- **niter**: Number of iterations to obtain the distribution.
- **return_null_dfield**: If TRUE the null distribution of dispersal field will be returned.
- **parallel**: If TRUE the computations will be performed in parallel.
- **n_cores**: Number of cores for the parallel computation.

Details

For more information about the biodiversity indices see Soberon and Cavner (2015). For detail about the diversity range analysis see Soberon et al. (2022). To plot diversity range results use `plot` method for objects of class `diversity_range`.

Value

An object of class `diversity_range`. The main result is the diversity range analysis which shows jointly two indices describing the community composition of every cell in the grid: (1) the relative number of species, and (2) the mean dispersion field (see plot method for `plot` (Soberon et al. 2022). The contains 12 slots with different measurements of biodiversity such as alpha diversity (species richness in each site or pixel), omega (size of the area of distribution of each species), dispersion field (the standardized size of the area of distribution of all species occurring in each pixel).

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

```r
set.seed(111)
pam <- matrix(rbinom(10000,1,0.5),nrow = 100,ncol = 1000)
rdivan <- bamm::diversity_range_analysis(pam=pam,
                                        parallel = FALSE)
```
eigen_bam

Description

Calculates the Eigen values and Eigen vectors of bam objects

Usage

eigen_bam(A = NULL, M = NULL, which_eigen = 1, rmap = TRUE)

Arguments

A  A bam object of class setA.
M  A bam object of class setM.
which_eigen  An integer representing the which eigen value and eigen vector will be computed.
rmap  Logical. If TRUE the function will return a map of the eigen vector of the product AM.

eigen_bam

eigen_bam: Compute the Eigen system of two bam objects

lagos_path <- system.file("extdata/conejos",           
  package = "bamm")
enm_path  <- list.files(lagos_path,                   
  pattern = ".tif",                                  
  full.names = TRUE)
en_models <- raster::stack(enm_path) > 0.01
nonas  <- which(!is.na(en_models[[1]]))
xy_mat  <- sp::coordinates(en_models[[1]])[ nonas,]
pam <- bamm::models2pam(en_models, sparse=FALSE)

rdivan <- bamm::diversity_range_analysis(pam=pam,
  xy_mat=xy_mat,
  raster_templete = en_models[[1]],
  parallel=TRUE,
  n_cores=2,
  return_null_dfield=TRUE)

bamm::plot(rdivan,plot_type="diversity_range")
# Lagomorphos

if(requireNamespace("plotly")){
  #bamm::plot(rdivan,plot_type="diversity_range_interactive")
}
Details

The eigenvector associated with the dominant eigenvalue of an adjacency matrix provides information about the number of forms in which a cell can be visited from other cells. Details about the eigen analysis in the context of the area of distribution can be found in Soberón and Osorio-Olvera (2022).

Value

A list with four objects. 1) eigen_values (these are indicated in which_eigen parameter of the function), 2) eigen_vectors (the corresponding eigen vectors of each eigen value), 3) Standardized eigen vectors (0 to 1), 4) A RasterLayer depicting the information of the first eigen vector of the system.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

```r
model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
sparse_mod <- bamm::model2sparse(model = model, 0.75)
plot(sparse_mod@niche_model)
adj_mod <- bamm::adj_mat(sparse_mod, ngbs = 1, eigen_sys = TRUE)
eig_bam <- bamm::eigen_bam(A = sparse_mod, M = adj_mod)
raster::plot(eig_bam$map)
```

---

**g_area-class**  
*S4 classes to organize data and results of bamm objects*

Description

*S4 classes to organize data and results of bamm objects*

Value

An object of class g_area
**jaccard**

Slots

coordinates A two column matrix with coordinates
eigen_vec Eigen vector of adjacency matrix
eigen_val Eigen value of adjacency matrix slot g_model A raster representing the geographic area slot g_sparse A sparse matrix of the geographic area

Author(s)

Luis Osorio-Olvera & Jorge Soberón

---

**jaccard**

`jaccard: Estimates the Jaccard index for comparing two binary maps`

**Description**

Estimates the Jaccard index for comparing two binary maps

**Usage**

```
jaccard(m1, m2)
```

**Arguments**

- `m1` A binary raster A or an object of class setA returned by the function `model2sparse`.
- `m2` A binary raster A or an object of class setA returned by the function `model2sparse`.

**Details**

The Jaccard index is computed as follows

\[
J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.
\]

**Value**

Returns a data.frame with three values: 1) jaccard (Jaccard index), 2) percentage_m1 (the percentage of m1 that the intersection \(|A \cap B|\) represents), and 3) percentage_m2

**Author(s)**

Luis Osorio-Olvera & Jorge Soberón
Examples

```r
m1_path <- system.file("extdata/conejos/Lepus_othus_cont.tif", package = "bamm")
m2_path <- system.file("extdata/conejos/Brachylagus_idahoensis_cont.tif", package = "bamm")
m1 <- raster::raster(m1_path) > 0.01
m2 <- raster::raster(m2_path) > 0.01
jcc <- bamm::jaccard(m1,m2)
print(jcc)
```

---

###leaflet-class

**Class**: leaflet

**Description**

Class leaflet leaflet

**Value**

An object of class leaflet

**Author(s)**

Luis Osorio-Olvera & Jorge Soberón

---

###model2sparse

**model2sparse**: Converts a niche model into a diagonal sparse matrix

**Description**

model2sparse: Converts a niche model into a diagonal sparse matrix

**Usage**

`model2sparse(model, threshold = NULL)`

**Arguments**

- `model`: A raster object representing the geographic projection of a niche model.
- `threshold`: A threshold to convert a continuous model into a binary model.

**Details**

threshold parameter represents the suitability value used to convert continuous model into a binary model.
Value
An object of class `setA`. The niche model is stored as diagonal sparse matrix (slot `sparse_model`).

Author(s)
Luis Osorio-Olvera & Jorge Soberón

Examples
```r
model_path <- system.file("extdata/Lepus_californicus_cont.tif", 
                          package = "bamm")
model <- raster::raster(model_path)

sparse_mod <- bamm::model2sparse(model, threshold=0.75)
print(sparse_mod)

raster::plot(sparse_mod@niche_model)
```

Description
Function to convert binary raster models to a Presence Absences Matrix.

Usage
`models2pam(mods_stack, sparse = TRUE, parallel = FALSE, ncores = 2)`

Arguments
- **mods_stack** A raster stack containing binary models of each species in the community.
- **sparse** Logical. If TRUE the PAM will be returned as a sparse matrix.
- **parallel** Logical. If TRUE computations will be done in parallel
- **ncores** Integer. Number of cores to run the parallel process.

Details
For more information about PAM see Soberon and Cavner (2015).

Value
A presence-absence matrix (PAM).

Author(s)
Luis Osorio-Olvera & Jorge Soberón
null\_dispersion\_field\_distribution

References

Examples

```r
lagos_path <- system.file("extdata/conejos", package = "bamm")
enm_path <- list.files(lagos_path, pattern = ".tif", full.names = TRUE)[1:10]
en_models <- raster::stack(enm_path) > 0.01
pam <- bamm::models2pam(en_models, sparse=FALSE, parallel=FALSE, ncores=2)
head(pam)
```

null\_dispersion\_field\_distribution

---

**null\_dispersion\_field\_distribution**: Null distribution of the dispersion field

**Description**

null\_dispersion\_field\_distribution estimates a random distribution of the dispersion field values.

**Usage**

```r
null\_dispersion\_field\_distribution(
pam,
  n_iter = 10,
  parallel = TRUE,
  n_cores = 2
)
```

**Arguments**

- **pam**: A Presence-Absence-Matrix of matrix class or sparse matrix.
- **n_iter**: Number of iterations to obtain the distribution.
- **parallel**: If TRUE the computations will be performed in parallel.
- **n_cores**: Number of cores for the parallel computation.

**Details**

Estimates a random distribution of the dispersion field values. To obtain random values it uses the function `codepermute_pam` at each step of the iterations. Randomization of the PAM is performed using the Babe Ruth Algorithm see Strona et al. (2014).
Value

A data matrix of size nrow(pam) X n_iter with dispersion field values.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

set.seed(111)
pam <- matrix(rbinom(100,1,0.3),nrow = 10,ncol = 10)
dfield_rand <- bamm::null Dispersion_field_distribution(pam,n_iter=10,
 parallel=FALSE,
 n_cores = 2)

head(dfield_rand)

---------

occs2sparse

occs2sparse: Converts occurrence data into a sparse matrix object

Description

occs2sparse: Converts occurrence data into a sparse matrix object

Usage

occs2sparse(modelSparse, occs)

Arguments

modelSparse A setA object returned by the function model2sparse

occs A matrix or a data.frame containing two columns. The first one is the longitude and the second is the latitude.

Details

Rows of this column vector represent non NA pixels of the niche model.

Value

A sparse vector of zeros (presences) and ones (absences).
pam-class

Author(s)
Luis Osorio-Olvera & Jorge Soberón

Examples

```r
code
model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
sparse_mod <- bamm::model2sparse(model, threshold=0.05)

occs_lep_cal <- data.frame(longitude = c(-115.10417, -104.90417),
                         latitude = c(29.61846, 29.81846))

occs_sparse <- bamm::occs2sparse(modelsparse = sparse_mod,
                                  occs = occs_lep_cal)

head(occs_sparse)
```

pam-class  

Class pam Presence-Absence Matrix

Description
Class pam Presence-Absence Matrix

Value
An object of class pam

Slots
- `pams`: A list of sparse matrices representing Presence-Absence Matrix for each simulation time
- `which_steps`: Simulation steps
- `sp_names`: Names of species in the PAM
- `grid`: Raster grid of the studied area
- `cellIDs`: Cells with ids of the PAM sites

Author(s)
Luis Osorio-Olvera & Jorge Soberón
pam2bioindex

**pam2bioindex: PAM to biodiversity index**

**Description**

pam2bioindex estimates various biodiversity indices for a certain PAM.

**Usage**

pam2bioindex(pam, biodiv_index = "dispersion_field", as_sparse = FALSE)

**Arguments**

- **pam** A Presence-Absence-Matrix of matrix class or sparse matrix.
- **biodiv_index** Possible values are alpha, omega, dispersion_field, all.
- **as_sparse** Return indices as sparse objects

**Details**

The biodiversity indices can be found in Soberón and Cavner (2015).

**Value**

An object of class bioindex with three slots each represents a matrix of diversity indices: alpha, omega, and dispersion field.

**Author(s)**

Luis Osorio-Olvera & Jorge Soberón

**References**


**Examples**

```r
set.seed(111)
pam <- matrix(rbinom(100, 1, 0.3), nrow = 10, ncol = 10)
bioindices <- bamm::pam2bioindex(pam=pam, biodiv_index="all")
# Return results as sparse models
bioindices <- bamm::pam2bioindex(pam=pam, biodiv_index="all", as_sparse=TRUE)
bioindices@alpha
bioindices@omega
bioindices@dispersion_field
```
pam2richness

pam2richness: Converts Presence Absence Matrix (pam object) to richness raster

Description

Converts Presence Absence Matrix (pam object) to richness raster

Usage

pam2richness(pamobj, which_steps)

Arguments

pamobj 
An object of class pam see csim2pam
which_steps 
Time steps in the pam to convert

Value

A RasterStack richness for each simulation step

Author(s)

Luis Osorio-Olvera & Jorge Soberón.

Examples

lagos_path <- system.file("extdata/conejos", 
  package = "bamm")

enm_path <- list.files(lagos_path, 
  pattern = ".tif", 
  full.names = TRUE)[seq(1,10)]

en_models <- raster::stack(enm_path)

ngbs_vect <- sample(2,replace = TRUE, 
  size = raster::nlayers(en_models))

init_coords <- read.csv(file.path(lagos_path, 
  "lagos_initit.csv"))[seq(1,10),]

nsteps <- 10

sdm_comm <- bamm::community_sim(en_models = en_models, 
  ngbs_vect = ngbs_vect, 
  init_coords = init_coords, 
  nsteps = nsteps, 
  threshold = 0.1)

pams <- bamm::csim2pam(community_sim = sdm_comm, 
  which_steps = seq_len(nsteps))

richness_stack <- bamm::pam2richness(pams, which_steps=pams@which_steps)
raster::plot(richness_stack)
permute_pam

permute_pam: Function to permute a Presence-Absence-Matrix.

Description

permute_pam: Function to permute a Presence-Absence-Matrix.

Usage

permute_pam(m, niter = NULL, as_sparse = FALSE)

Arguments

m Presence-Absence-Matrix (PAM) or a binary matrix with columns representing species and rows sites.
niter Number of iterations to permute the PAM.
as_sparse If TRUE the PAM will be returned as a sparse matrix

Details

This function is an implementation of the curve ball algorithm following Strona et al. (2014).

Value

Returns a permuted matrix of the same dimensions of m (same number of rows and columns). Note that the sum of each row and column of this permuted matrix is equal to that of m. species.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

References


Examples

set.seed(111)
pam <- matrix(rbinom(100,1,0.3),nrow = 10,ncol = 10)
ppam <- bamm::permute_pam(m = pam,niter = NULL,as_sparse = FALSE)
# Check if matrices are different
all(pam == ppam)
# Check if row totals are the same
all(Matrix::rowSums(pam) == Matrix::rowSums(ppam))
# Check if column total are the same
all(Matrix::colSums(pam) == Matrix::colSums(ppam))
Plot method for objects of class `diversity_range` \texttt{bamm}.

**Description**

Plot method for objects of class `diversity_range` \texttt{bamm}.

**Usage**

```r
## S4 method for signature 'diversity_range,ANY'
plot(
  x,
  xlab = NULL,
  plot_type = "diversity_range",
  legend = TRUE,
  legend_position = "bottomright",
  ylab = NULL,
  col = NULL,
  pch = NULL,
  pch_legend = 19,
  radius = 0.5,
  ...)
```

**Arguments**

- \texttt{x}\hspace{1cm}An object of class `diversity_range`
- \texttt{xlab}\hspace{1cm}x label
- \texttt{plot_type}\hspace{1cm}Plot type: possible options: "diversity_range" (range-diversity plot), "diversity_range_map" (a raster map with diversity_range categories), "alpha" (a raster map with alpha diversity values), "dispersion_field" (a raster with dispersion field)
- \texttt{legend}\hspace{1cm}Logical. If TRUE the legend of the categorical diversity range values will appear.
- \texttt{legend_position}\hspace{1cm}Legend position.
- \texttt{ylab}\hspace{1cm}y label
- \texttt{col}\hspace{1cm}Plot colors.
- \texttt{pch}\hspace{1cm}Patch type.
- \texttt{pch_legend}\hspace{1cm}Patch type for legends.
- \texttt{radius}\hspace{1cm}Size of the patch for the interactive map.
- \texttt{...}\hspace{1cm}Graphical parameters. Any argument that can be passed to 1) base::plot, such as axes=FALSE, main='title', ylab='latitude' 2) leaflet::leaflet or 3) leaflet::addCircleMarkers.
predict.bam-method

Details
To show interactive diversity_range plots install the 'plotly' R package.

Value
Plot of the results of the diversity_range analysis

Author(s)
Luis Osorio-Olvera & Jorge Soberón

predict, bam-method

Description
predicts species' distribution under suitability changes

Usage
## S4 method for signature 'bam'
predict(
  object,
  niche_layers,
  nbgs_vec = NULL,
  nsteps_vec,
  stochastic_dispersal = FALSE,
  disp_prop2_suitability = TRUE,
  disper_prop = 0.5,
  animate = FALSE,
  period_names = NULL,
  fmt = "GIF",
  filename,
  bg_color = "#F6F2E5",
  suit_color = "#0076BE",
  occupied_color = "#03C33F",
  png_keyword = "sdm_sim",
  ani.width = 1200,
  ani.height = 1200,
  ani.res = 300
)

Arguments
object a of class bam.
niche_layers A raster or RasterStack with the niche models for each time period
predict.bam-method

nbgs_vec A vector with the number of neighbors for the adjacency matrices
nsteps_vec Number of simulation steps for each time period.
stochastic_dispersal Logical. If dispersal depends on a probability of visiting neighbor cells (Moore neighborhood).

disp_prop2_suitability Logical. If probability of dispersal is proportional to the suitability of reachable cells. The proportional value must be declared in the parameter ‘disper_prop’.
disper_prop Probability of dispersal to reachable cells.
animate Logical. If TRUE a dispersal animation on climate change scenarios will be created
period_names Character vector with the names of periods that will be animated. Default NULL.
fmt Animation format. Possible values are GIF and HTML
filename File name.
bg_color Color for unsuitable pixels. Default "#F6F2E5".
suit_color Color for suitable pixels. Default "#0076BE".
occupied_color Color for occupied pixels. Default "#03C33F".
png_keyword A keyword name for the png images generated by the function
ani.width Animation width unit in px
ani.height Animation height unit in px
ani.res Animation resolution unit in px

Value
A RasterStack of predictions of dispersal dynamics as a function of environmental change scenarios.

Author(s)
Luis Osorio-Olvera & Jorge Soberón

Examples

# rm(list = ls())
# Read raster model for Lepus californicus
model_path <- system.file("extdata/Lepus_californicus_cont.tif",
    package = "bamm")
model <- raster::raster(model_path)
# Convert model to sparse
sparse_mod <- bamm::model2sparse(model = model,threshold=0.1)
# Compute adjacency matrix
adj_mod <- bamm::adj_mat(sparse_mod,ngbs=1)

# Initial points to start dispersal process
occs_lep_cal <- data.frame(longitude = c(-115.10417,
sdm_sim: Simulate single species dispersal dynamics using the BAM framework.

Description

sdm_sim: Simulate single species dispersal dynamics using the BAM framework.
Usage

sdm_sim(
  set_A,
  set_M,
  initial_points,
  nsteps,
  stochastic_dispersal = TRUE,
  disp_prop2_suitability = TRUE,
  disper_prop = 0.5,
  progress_bar = TRUE
)

Arguments

set_A: A setA object returned by the function `model2sparse`
set_M: A setM object containing the adjacency matrix of the study area. See `adj_mat`
initial_points: A sparse vector returned by the function `occs2sparse`
nsteps: Number of steps to run the simulation
stochastic_dispersal: Logical. If dispersal depends on a probability of visiting neighbor cells (Moore neighborhood).
disp_prop2_suitability: Logical. If probability of dispersal is proportional to the suitability of reachable cells. The proportional value must be declared in the parameter ‘disper_prop’.
disper_prop: Probability of dispersal to reachable cells.
progress_bar: Show progress bar

Details

The model is cellular automata where the occupied area of a species at time $t + 1$ is estimated by the multiplication of two binary matrices: one matrix represents movements (M), another abiotic niche-tolerances (A) (Soberon and Osorio-Olvera, 2022).

$$G_j(t + 1) = A_j(t)M_jG_j(t)$$

The equation describes a very simple process: To find the occupied patches in $t + 1$ start with those occupied at time $t$ denoted by $G_j(t)$, allow the individuals to disperse among adjacent patches, as defined by $M_j$, then remove individuals from patches that are unsuitable, as defined by $A_j(t)$.

Value

An object of class `bam` with simulation results. The simulation are stored in the `sdm_sim` slot (a list of sparse matrices).

Author(s)

Luis Osorio-Olvera & Jorge Soberón
setA-class

References

Examples

model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
sparse_mod <- bamm::model2sparse(model, threshold=0.05)
adj_mod <- bamm::adj_mat(sparse_mod, ngbs=1)
occs_lep_cal <- data.frame(longitude = c(-110.08880, -98.89638),
latitude = c(30.43455, 25.19919))
occs_sparse <- bamm::occs2sparse(modelsparse = sparse_mod, occs = occs_lep_cal)

setA-class

Class for the A set of the BAM diagram

Description
A class for the A set of the BAM diagram. It contains raster models and IDs of pixels with values different than NA.

Value
An object of class setA showClass("setA")
setM-class

Slots

niche_model A niche model in raster format. It can be a binary model or continuous. If the model is in a continuous format.
suit_threshold Suitability value used to binarize continuous model
cellIDs A numeric vector with the IDs of the cells with prediction values
suit_values A numeric vector with suitability value of the continuous map
sparse_model A niche model in sparse matrix format

Author(s)

Luis Osorio-Olvera & Jorge Soberón

Description

Class for the M set of the bamm diagram

Value

An object of class setM

Slots

adj_matrix An adjacency matrix
adj_list An adjacency list
initial_points A presence-absence vector with species' occurrences
n_initial_points Number of initial points used to start the dispersal process
ngbs Number of neighbors

Author(s)

Luis Osorio-Olvera & Jorge Soberón

Examples

showClass("setM")
Description

shape2Grid creates a raster grid given a spatial polygon and a grid resolution.

Usage

shape2Grid(shpolygon, resolution, ones = TRUE)

Arguments

- shpolygon: A SpatialPolygon, SpatialPolygonDataFrame representing the desired shape of the grid.
- resolution: Numeric. Spatial resolution of the grid.
- ones: Logical. Fill with ones the values of the raster. If not the values will be written as cellID values.

Value

Returns a raster object with the shape of `shpolygon` of a given resolution.

Author(s)

Luis Osorio-Olvera & Jorge Soberón

Examples

```r
x_coord <- c(-106.5699, -111.3737, -113.9332, -110.8913, -106.4262, -106.5699)
xy <- cbind(x_coord, y_coord)
p <- sp::Polygon(xy)
ps <- sp::Polygons(list(p), 1)
sps <- sp::SpatialPolygons(list(ps))
r1 <- bamm::shape2Grid(sps, resolution = 0.1, ones = FALSE)
plot(r1)
sp::plot(sps, add = TRUE)
```
Description

Show information in setA class `bamm`.
Show information in csd class `bamm`.
Show information in pam class `bamm`.
Show information in pam class `bamm`.
Show information in setA class `bamm`.
Show information in diversity_range class `bamm`.

Usage

```r
## S4 method for signature 'setA'
show(object)

## S4 method for signature 'csd'
show(object)

## S4 method for signature 'pam'
show(object)

## S4 method for signature 'bioindex_sparse'
show(object)

## S4 method for signature 'setM'
show(object)

## S4 method for signature 'diversity_range'
show(object)
```

Arguments

- `object`: An object of class `diversity_range`

Value

- Display information about the setA object
- Display information about the csd object
- Display information about the pam object
- Display information about the bioindex_sparse object
- Display information about the setM object
- Display information about the diversity_range object
**Author(s)**
Luis Osorio-Olvera & Jorge Soberón

---

**sim2Animation**

*sim2Animation: Animate BAM simulation object.*

---

**Description**
Animates BAM simulation object.

**Usage**

```r
sim2Animation(
  sdm_simul,
  which_steps,
  fmt = "GIF",
  filename,
  png_keyword = "sdm_sim",
  extra_legend = NULL,
  bg_color = "#F6F2E5",
  suit_color = "#0076BE",
  occupied_color = "#03C33F",
  gif_vel = 0.8,
  ani.width = 1200,
  ani.height = 1200,
  ani.res = 300
)
```

**Arguments**

- **sdm_simul**: A bam object. See `sdm_sim`
- **which_steps**: A numeric vector indicating the simulation steps that are going to be converted into raster layers.
- **fmt**: Animation format. Possible values are GIF and HTML
- **filename**: File name.
- **png_keyword**: A keyword name for the png images generated by the function
- **extra_legend**: A legend to add to the animation.
- **bg_color**: Color for unsuitable pixels. Default "#F6F2E5".
- **suit_color**: Color for suitable pixels. Default "#0076BE".
- **occupied_color**: Color for occupied pixels. Default "#03C33F".
- **gif_vel**: A value that regulates the velocity of frame transitions. The bigger it is the transition will be slower default 0.8
- **ani.width**: Animation width unit in px
- **ani.height**: Animation height unit in px
- **ani.res**: Animation resolution unit in px
Details

The animation can be saved in a GIF or HTML format. Note that the generation of the GIF can be time consuming for large simulation (simulations with more than 60 time steps).

Value

A RasterStack of species’ distribution at each simulation step

Author(s)

Luis Osorio-Olvera & Jorge Soberón

Examples

```r
model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
sparse_mod <- bamm::model2sparse(model, 0.1)
adj_mod <- bamm::adj_mat(sparse_mod, ngbs = 2)
occs_lep_cal <- data.frame(longitude = c(-115.10417, -104.90417),
                          latitude = c(29.61846, 29.81846))
occs_sparse <- bamm::occs2sparse(modelsparse = sparse_mod,
occs = occs_lep_cal)
sdm_lep_cal <- bamm::sdm_sim(set_A = sparse_mod,
                          set_M = adj_mod,
                          initial_points = occs_sparse,
                          nsteps = 50)

if(requireNamespace("animation")){
  ani_name <- tempfile(pattern = "simulation_", fileext = ".html")
sdm_lep_cal_st <- bamm::sim2Animation(sdm_simul = sdm_lep_cal,
                                          which_steps = seq(1, 50, by = 1),
                                          fmt = "HTML", ani.width = 1200,
                                          ani.height = 1200,
                                          filename = ani_name)
}
```

---

**sim2Raster**  
*sim2Raster: Convert a BAM simulation object to RasterStack*

Description

Convert a BAM simulation object to RasterStack.

Usage

```r
sim2Raster(sdm_simul, which_steps = NULL)
```
Arguments

sdm_simul A bam object. See sdm_sim
which_steps A numeric vector indicating the simulation steps that are going to be converted into raster layers.

Value

A RasterStack of species’ distribution at each simulation step

Author(s)

Luis Osorio-Olvera & Jorge Soberón

Examples

model_path <- system.file("extdata/Lepus_californicus_cont.tif", package = "bamm")
model <- raster::raster(model_path)
sparse_mod <- bamm::model2sparse(model, threshold=0.1)
adj_mod <- bamm::adj_mat(sparse_mod, ngbs = 1)
occs_lep_cal <- data.frame(longitude = c(-115.10417, -104.90417), latitude = c(29.61846, 29.81846))
occs_sparse <- bamm::occs2sparse(model_sparse = sparse_mod, occs = occcs_lep_cal)

sdm_lep_cal <- bamm::sdm_sim(set_A = sparse_mod, set_M = adj_mod, initial_points = occcs_sparse, nsteps = 10)

sdm_lep_cal_st <- bamm::sim2Raster(sdm_simul = sdm_lep_cal, which_steps = seq(1,10,by=1))

raster::plot(sdm_lep_cal_st)
Index

.A-class (setA-class), 35
.adj_mat, 4, 7, 9, 12, 34
.bam, 12, 34
.bam (bam-class), 5
.bam-class, 5
.bam-method (predict, bam-method), 31
.bam_clusters, 6, 14
.bam_sim, 7
.bam_ssim, 9
.biodiversity_index (bioindex-class), 11
.biodiversity_index_sparse
(bioindex_sparse-class), 11
.bioindex, 27
.bioindex (bioindex-class), 11
.bioindex-class, 11
.bioindex_sparse
(bioindex_sparse-class), 11
.bioindex_sparse-class, 11
.community-class (community_sim-class), 13
.community_bam, 15
.community_bam (community_sim-class), 13
.community_sim, 12, 15
.community_sim-class, 13
.coordinates-class (g_area-class), 20
.csd, 6
.csd (csd-class), 14
.csd-class, 14
.csd_estimate, 14
.csim2pam, 15, 28
.diversity_range, 18
.diversity_range
(diversity_range-class), 16
.diversity_range-class, 16
.diversity_range_analysis, 17
.diversityrange (diversity_range-class), 16
.eigen_bam, 19
.g_area (g_area-class), 20
.g_area-class, 20
.jaccard, 21
.leaflet (leaflet-class), 22
.leaflet-class, 22
.M-class (setM-class), 36
.model2sparse, 4, 6, 21, 22, 25, 34
.models2pam, 23
.null_dispersion_field_distribution, 24
.occs2sparse, 7, 9, 25, 34
.PAM (pam-class), 26
.pam, 16
.pam (pam-class), 26
.pam-class, 26
.pam2bioindex, 27
.pam2richness, 28
.permute_pam, 24, 29
.plot, 18
.plot, diversity_range, ANY-method, 30
.predict (predict, bam-method), 31
.predict, bam-method, 31
.sdm_sim, 12, 33, 39, 41
.setA, 6, 7, 9, 23
.setA (setA-class), 35
.setA-class, 35
.setM, 4, 7, 9
.setM (setM-class), 36
.setM-class, 36
.shape2Grid, 37
show, bioindex_sparse-method (show, setA-method), 38
show, csd-method (show, setA-method), 38
show, diversity_range-method (show, setA-method), 38
show, pam-method (show, setA-method), 38
show, setA-method, 38
show, setM-method (show, setA-method), 38
sim2Animation, 39
sim2Raster, 40