Package 'bayesLopod'

January 4, 2018

Type Package

Title Bayesian Inference of Landscape Occupancy from Presence-Only Data

Version 1.0.1

Date 2017-12-28

Description
Natural history museums and herbaria collectively hold hundreds of millions of zoological, botanical, and paleontological specimens. These collections serve as the foundation for understanding the distribution of life on Earth and the basis for addressing loss of biodiversity, emerging diseases, and other pressing global problems as well as important question in ecology and evolution. One of the short comings of these kind of data is that the lack of evidence of the presence of a species in a certain region does not mean the species is truly absent there. Likewise, specimens are often misidentified, and therefore the report of a species in a locality is not always evidence that a viable population occurs there. The goal of this project is to develop a method which could be used to estimate the probability of presence of a species in a certain study region based on certain sampling effort and presence reports.

License GPL-3

Depends R (>= 3.0.2), Rcpp (>= 0.12.12), methods, rstan (>= 2.16.2), rstantools (>= 1.2.0)

Imports raster (>= 2.5-8), slam (>= 0.1-40), sp (>= 1.2-4), inline (>= 0.3.14), rgeos (>= 0.3-26)

LinkingTo StanHeaders (>= 2.16.0.1), rstan (>= 2.16.2), BH (>= 1.65.0.1), Rcpp (>= 0.12.12), RcppEigen (>= 0.3.3.3.0)

RoxygenNote 6.0.1

NeedsCompilation yes

Author Camilo Sanin [aut, cre, cph],
Adam B. Smith [aut, cph],
Missouri Botanical Garden [cph],
Trustees of Columbia University [cph]

Maintainer Camilo Sanin <camilosanin@gmail.com>

Repository CRAN

Date/Publication 2018-01-04 15:29:49 UTC

R topics documented:

`Andropogon_shape` *Number of detection of Andropogon gerardii and it's target group in several counties in the Midwest*

Description

Number of detection of Andropogon gerardii and it's target group in several counties in the Midwest

Usage

```
Andropogon_shape
```

Format

An object of class `SpatialPolygonsDataFrame` with 617 rows and 3 columns.

LopodData-class *An S4 class to contain data to be input into a bayesLopod model.*

Description

An S4 class to contain data to be input into a bayesLopod model.

Slots

`geoDataObject` Spatial object supported by bayesLopod

`geoType` Type of geographical data (Only rasters supported for now)

`geoInfo` Additional spatial information to be passed to modelLopod

lopodDens *Kernel density estimates of global occupancy model parameters.*

Description

Kernel density estimates of global occupancy model parameters.

Usage

```
lopodDens(LopodModel, params = NULL)
```

Arguments

`LopodModel`	A LopodModel object
`params`	Parameters to be plotted. Default is NULL, which plots all global parameters

Value

A ggplot object.

Examples

```
## Not run:
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
ld_Raster_adMatrix = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = TRUE )
mLopodRaster = modelLopod(LopodData = ld_Raster_adMatrix, varP = TRUE, q = NULL,
pmin = 0.1, CAR = FALSE, nChains = 4,warmup = 500,sampling = 100,nCores = 4)
lopodDens(mLopodRaster, c("alpha", "tau"))
```

```
data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = TRUE, keepFields = FALSE)
mLopodShape = modelLopod(LopodData = ld_Shape, varP = TRUE, q = NULL,
pmin = 0, CAR = TRUE, nChains = 4,warmup = 500,sampling = 100,nCores =4)
lopodDens(mLopodShape, c("q", "pmin", "pmax"))

## End(Not run)
```

`LopodModel-class` *An S4 class containing a LopodModel.*

Description

An S4 class containing a LopodModel.

Slots

`LopodData` The LopodData object used to build the LopodModel

`StanFit` stanfit object with the run model

`modelInfo` List of settings used to run the LopodModel

`lopodRaster` *Crate raster object for a parameter estimated in a LopodModel*

Description

Crate raster object for a parameter estimated in a LopodModel

Usage

```
lopodRaster(LopodModel, param, extrapolate = T, metric = NULL,
  quant = 0.5)
```

Arguments

`LopodModel`	A LopodModel object
`param`	Unit-level model parameter to be mapped. Values "psi_Sampled" can be mapped for models without CAR analyses, "psi_i" for LopodModels with CAR analysis and "pp","cellpres_i", "pCorr","sim_y","sim_true_y","sim_false_y" for both.
`extrapolate`	Boolean. If True, parameters are mapped for cells that have not been sampled, this can only be done in LopodModels with CAR analysis. Only plotted for "psi_i", "pp" and, "cellpres_i".

metric	"mean" or "sd". Plots the mean or standard deviation of the posterior distribution. If NULL, the value in quant is used.
quant	Returns the raster for a given quantile of the posterior distribution. Default is 0.5 (the median of the posterior distribution). Not used if metric is other than NULL

Value

A Raster object.

Examples

```
## Not run:
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
ld_Raster_adMatrix = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = TRUE )
mLopodRaster = modelLopod(LopodData = ld_Raster_adMatrix, varP = TRUE, q = NULL,
pmin = 0.1, CAR = FALSE, nChains = 4,warmup = 500,sampling = 100,nCores = 4)

psiRaster = lopodRaster(mLopodRaster, param = "psi_i", extrapolate = TRUE, quant = 0.5)
ppRaster = lopodRaster(mLopodRaster, param = "pp", extrapolate = FALSE, metric = "mean")

Visualize results
sp::spplot(psiRaster)
sp::spplot(ppRaster)

## End(Not run)
```

---

lopodShape *Crate Shape object for a parameter estimated in a LopodModel*

Description

Crate Shape object for a parameter estimated in a LopodModel

Usage

```
lopodShape(LopodModel, param, extrapolate = T, metric = NULL, quant = 0.5)
```

Arguments

LopodModel	A LopodModel object
param	Unit-level model parameter to be mapped. Values "psi_Sampled" can be mapped for models without CAR analyses, "psi_i" for LopodModels with CAR analysis and "pp","cellpres_i", "pCorr","sim_y","sim_true_y","sim_false_y" for both.

extrapolate	Boolean. If True, parameters are mapped for cells that have not been sampled, this can only be done in LopodModels with CAR analysis. Only plotted for "psi_i", "pp" and, "cellpres_i".
metric	"mean" or "sd". Plots the mean or standard deviation of the posterior distribution. If NULL, the value in quant is used.
quant	Returns the raster for a given quantile of the Posterior Distribution. Default is 0.5 (the median of the posterior distribution). Not used if metric is other than NULL

Value

A Raster object.

Examples

```
## Not run:
data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = TRUE, keepFields = FALSE)
mLopodShape = modelLopod(LopodData = ld_Shape, varP = TRUE, q = NULL,
pmin = 0, CAR = TRUE, nChains = 4,warmup = 500,sampling = 100,nCores =4)
psiShape_95 = lopodShape(mLopodShape, "psi_i", extrapolate = FALSE,  quant = 0.95)
psiShape_05 = lopodShape(mLopodShape, "psi_i", extrapolate = TRUE, quant = 0.05)
 #Visualize results
sp::spplot(psiShape_05, zcol = "psi_i")
sp::spplot(psiShape_95, zcol = "psi_i")

## End(Not run)
```

---

lopodSummary *Summary statistics for a LopodModel*

Description

Summary statistics for a LopodModel

Usage

```
lopodSummary(LopodModel, params = NULL, probs = c(0.05, 0.5, 0.95))
```

Arguments

LopodModel	A LopodModel object
params	Parameters to be plotted. Default is NULL, which plots all global parameters
probs	Quantiles to be estimated

Value

Returns a dataframe which contain summaries for all chains merged for the Global Parameters of a LopodModel. Included in the summary are quantiles, means, standard deviations (sd), effective sample sizes (n_eff), Monte Carlo standard errors (se_mean) and Rhats.

Examples

```
 ## Not run:
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
ld_Raster_adMatrix = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = TRUE )
mLopodRaster = modelLopod(LopodData = ld_Raster_adMatrix, varP = TRUE, q = NULL,
pmin = 0.1, CAR = FALSE, nChains = 4,warmup = 500,sampling = 100,nCores = 4)
lopodSummary(mLopodRaster, params = c("psi","p","q"))

data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = TRUE, keepFields = FALSE)
mLopodShape = modelLopod(LopodData = ld_Shape, varP = TRUE, q = NULL,
pmin = 0, CAR = TRUE, nChains = 4,warmup = 500,sampling = 100,nCores =4)
lopodSummary(mLopodRaster)

## End(Not run)
```

---

lopodTrace *Plots the values of model parameters for each chain across iterations*

Description

Plots the values of model parameters for each chain across iterations

Usage

```
lopodTrace(LopodModel, params = "lp__", inc_warmup = FALSE)
```

Arguments

LopodModel	A LopodModel object
params	Parameters to be plotted. Default is "lp__" which plots the log posterior probability
inc_warmup	Boolean. If true, warm-up iterations are plotted. Default is FALSE.

Value

A ggplot object.

Examples

```
## Not run:
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
ld_Raster_adMatrix = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = TRUE )
mLopodRaster = modelLopod(LopodData = ld_Raster_adMatrix, varP = TRUE, q = NULL,
pmin = 0.1, CAR = FALSE, nChains = 4,warmup = 500,sampling = 100,nCores = 4)
lopodTrace(mLopodRaster, inc_warmup = FALSE, params = c("p","q"))

data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = TRUE, keepFields = FALSE)
mLopodShape = modelLopod(LopodData = ld_Shape, varP = TRUE, q = NULL,
pmin = 0, CAR = TRUE, nChains = 4,warmup = 500,sampling = 100,nCores =4)
lopodTrace(mLopodShape, inc_warmup = TRUE)

## End(Not run)
```

`modelLopod` *Run a Bayes inference of Landscape Occupancy from Presence-Only Data on Stan*

Description

Run a Bayes inference of Landscape Occupancy from Presence-Only Data on Stan

Usage

```
modelLopod(LopodData, varP = F, q = NULL, pmin = 0, CAR = F,
  nChains = 4, warmup = 2000, sampling = 1000, nCores = 4)
```

Arguments

`LopodData`	Object with the data to be used in the Model
`varP`	Boolean. If TRUE, detectability will vary across cells. If FALSE a global value for detectability will be estimated.
`q`	Number between 0 and 1 or NULL. Rate of false detections. If NULL the values will be estimated by the model.
`pmin`	Number between 0 and 1. Minimum value for detectability in a unit in which the species occurs.
`CAR`	Boolean. If TRUE, (and if a adjacency matrix is included in the LopodData object) a conditional auto-regression analysis will be performed for occupancy across units.

nChains	Number of Markov chains used by the Stan model.
warmup	Number of iterations for each chain to be discarded as warm-up.
sampling	Number of iterations for each chain to be sampled (after warm-up).
nCores	Number of cores to use when executing the chains in parallel.

Value

A LopodModel object.

Examples

```
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 10, extentExpansion = 0)
ld_Raster = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = FALSE )
mLopodRaster = modelLopod(LopodData = ld_Raster, varP = TRUE, q = NULL,
pmin = 0, CAR = FALSE, nChains = 1,warmup = 10,sampling = 10,nCores = 1)

data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = FALSE, keepFields = FALSE)
mLopodShape = modelLopod(LopodData = ld_Shape, varP = FALSE, q = NULL,
pmin = 0, CAR = FALSE, nChains = 1,warmup = 5,sampling = 5,nCores = 1)

## Not run:
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
ld_Raster_adMatrix = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = TRUE )
mLopodRaster = modelLopod(LopodData = ld_Raster_adMatrix, varP = TRUE, q = NULL,
pmin = 0.1, CAR = FALSE, nChains = 4,warmup = 500,sampling = 100,nCores = 4)

data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = TRUE, keepFields = FALSE)
mLopodShape = modelLopod(LopodData = ld_Shape, varP = TRUE, q = NULL,
pmin = 0, CAR = TRUE, nChains = 4,warmup = 500,sampling = 100,nCores =4)

## End(Not run)
```

---

modelParams *Internal function to extract parameters from a StanModel object*

Description

Internal function to extract parameters from a StanModel object

Usage

```
modelParams(LopodModel)
```

Arguments

LopodModel	A LopodModel object

Value

A list of parameters.

---

rasterDistToRecords *Internal - Create a Stack with distance matrices to sampled and detected cells*

Description

Internal - Create a Stack with distance matrices to sampled and detected cells

Usage

```
rasterDistToRecords(rasterN, rasterY)
```

Arguments

rasterN	Raster object with sampling effort (number of sampling events)in each cell.
rasterY	Raster object with number of detections in each cell.

Value

A Stack object with distance to sampled cells and distance to detections.

---

rasterLopodData *Create a LopodData object from Raster data*

Description

Create a LopodData object from Raster data

Usage

```
rasterLopodData(rasterN, rasterY, Adjacency = T, extSample = 0.025,
  extDetection = 0.15)
```

Arguments

rasterN	Raster object with sampling effort (number of sampling events)in each cell.
rasterY	Raster object with number of detections in each cell.
Adjacency	Boolean. If TRUE, and adjacency matrix is computed.
extSample	Number between 0 and 1. Maximum distance (relative to the diagonal of the raster) from a sampled cell that should be included in the study area. If 0, there is no extrapolation to not sampled cells.
extDetection	Number between 0 and 1. Maximum distance (relative to the diagonal of the raster) from cell in which the species was detected that should be included in the study area. If 0, there is no extrapolation to not sampled cells.

Value

A LopodData object to be used in modelLopod.

Examples

```
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 10, extentExpansion = 0)
ld_Raster = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = FALSE )
## Not run:
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
ld_Raster_adMatrix = rasterLopodData(rasterN = simSpRasters[["samplingEffort"]],
rasterY = simSpRasters[["spDetections"]], Adjacency = TRUE )

## End(Not run)
```

---

rasterStudyArea *Internal - Create a Raster object with Study Area from Raster data*

Description

Internal - Create a Raster object with Study Area from Raster data

Usage

```
rasterStudyArea(DistStak, maxDist, extSample = 0.025, extDetection = 0.15)
```

Arguments

DistStak	Stack object distance to Sampled cells and detections (in that order).
maxDist	Maximum distance in Raster
extSample	Number between 0 and 1. Maximum distance (relative to the diagonal of the raster) from a sampled cell that should be included in the study area. If 0, there is no extrapolation to not-sampled cells.
extDetection	Number between 0 and 1. Maximum distance (relative to the diagonal of the raster) from cell in which the species was detected that should be included in the study area. If 0, there is no extrapolation to no-sampled cells.

Value

A Raster object with Study Area.

---

shapeLopodData *Create a LopodData object from Raster data*

Description

Create a LopodData object from Raster data

Usage

```
shapeLopodData(Shapefile, fieldN = "sampEffort", fieldY = "detections",
  Adjacency = T, keepFields = T)
```

Arguments

Shapefile	SpatialPolygonsDataFrame Object with at least two Fields corresponding to sampling effort and number of detections in each feature
fieldN	Field in Shapefile corresponding to sampling effort (number of sampling events)in each feature.
fieldY	Field in Shapefile corresponding to number of detections in each feature.
Adjacency	Boolean. If TRUE, and adjacency matrix is computed.
keepFields	Boolean. If TRUE, other fields of the Shapefile will be kept and "sampEffort" and "detections" will be added. If FALSE, only the "sampEffort" and "detections" will be kept in the LopodData Object.

Value

A LopodData object to be used in modelLopod.

Examples

```
data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = FALSE, keepFields = FALSE)

## Not run:
data("Andropogon_shape", package = "bayesLopod")
ld_Shape = shapeLopodData(Shapefile = Andropogon_shape, fieldN = "sampEffort",
fieldY = "detections",  Adjacency = TRUE, keepFields = TRUE)

## End(Not run)
```

---

simSpRecords *Coordinates for detections of a simulated species*

Description

Coordinates for detections of a simulated species

Usage

```
simSpRecords
```

Format

An object of class `SpatialPoints` of length 8634.

---

simSpSamplingEffort *Coordinates for sampling events targeting a simulated species*

Description

Coordinates for sampling events targeting a simulated species

Usage

```
simSpSamplingEffort
```

Format

An object of class `SpatialPoints` of length 37860.

xyToRaster *Crate Shape object for a parameter estimated in a LopodModel*

Description

Crate Shape object for a parameter estimated in a LopodModel

Usage

```
xyToRaster(xyRecords, xySamplingEffort, nrows = 50, extentExpansion = 0.1,
  extent = NULL, basemap = getData("worldclim", var = "alt", res = 10))
```

Arguments

xyRecords	Object of class SpatialPoints or SpatialPointsDataFrame with the locality-records of the species.
xySamplingEffort	Object of class SpatialPoints or SpatialPointsDataFrame with the coordinates of all sampling events (including those in which the species was found).
nrows	Number of rows that the final Raster will have. It will be use to determine its resolution.
extentExpansion	Factor by which the final extent should be expended. If 0 and extent is NULL, the final extent will be determined by the most extreme locality records. If extent is given, extentExpansion will be ignored.
extent	Object of class Extent delimiting the region to be included in the final raster
basemap	Object of class Raster in which cell with NA values will not be included. If NULL, all cells in the raster will be included.

Value

A Stack object with two layers: one for the number of records per cell ("samplingEffort"), and another one with the number of sampling events ("spDetections").

Examples

```
data("simSpRecords", package = "bayesLopod")
data("simSpSamplingEffort", package = "bayesLopod")
simSpRasters = xyToRaster(xyRecords = simSpRecords,xySamplingEffort = simSpSamplingEffort,
basemap = NULL, nrows = 50, extentExpansion = 0)
```

Index