

# Package ‘intSDM’

October 13, 2022

**Title** Reproducible Integrated Species Distribution Models Across Norway using 'INLA'

**Description** Integration of disparate datasets is needed in order to make efficient use of all available data and thereby address the issues currently threatening biodiversity. Data integration is a powerful modeling framework which allows us to combine these datasets together into a single model, yet retain the strengths of each individual dataset. We therefore introduce the package, 'intSDM': an R package designed to help ecologists develop a reproducible workflow of integrated species distribution models, using data both provided from the user as well as data obtained freely online. An introduction to data integration methods is discussed in Issac, Jarzyna, Keil, Dambly, Boersch-Supan, Browning, Freeman, Golding, Guillera-Arroita, Henrys, Jarvis, Lahoz-Monfort, Pagel, Pescott, Schmucki, Simmonds and O'Hara (2020) <[doi:10.1016/j.tree.2019.08.006](https://doi.org/10.1016/j.tree.2019.08.006)>.

**Version** 1.0.0

**Depends** R (>= 3.5), inlabru (>= 2.3.1), ggplot2, sp (>= 1.4-5), stats, PointedSDMs, methods

**Imports** dplyr, spocc, raster, maps, maptools

**Suggests** fields, RANN, INLA (>= 21.08.31), rgeos, viridis, ggpolypath, ggmap, spatstat, RColorBrewer, knitr, rmarkdown, testthat (>= 3.0.0), sf, covr

**Additional\_repositories** <https://inla.r-inla-download.org/R/testing/>

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.0

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Philip Mostert [aut, cre],  
Angeline Bruls [aut],  
Ragnhild {Bjørkås} [aut],  
Wouter Koch [aut],  
Ellen Martin [aut]

**Maintainer** Philip Mostert <philip.s.mostert@ntnu.no>

**Repository** CRAN

**Date/Publication** 2022-08-30 12:40:02 UTC

## R topics documented:

listorNULL-class . . . . .	2
PA_redlist . . . . .	2
show,structuredData-method . . . . .	3
species_model . . . . .	3
structuredData-class . . . . .	6
structured_data . . . . .	6
<b>Index</b>	<b>8</b>

---

listorNULL-class	<i>Outputs for structuredData class</i>
------------------	---

---

### Description

Outputs for structuredData class

---

PA_redlist	<i>A dataset containing the presences and absences of three species: "Fraxinus excelsior", "Ulmus glabra" and "Arnica montana"</i>
------------	--

---

### Description

A dataset containing the presences and absences of three species: "Fraxinus excelsior", "Ulmus glabra" and "Arnica montana"

### References

<https://www.gbif.org>

---

 show,structuredData-method

*Print methods for structuredData object*


---

### Description

Print methods for structuredData object

### Usage

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

### Arguments

object            A "structuredData" object.

### Value

A print of the names of the datasets, as well as their lengths.

---

 species\_model

*species\_model: function to construct an integrated species distribution model, as well as other useful outputs from the model.*


---

### Description

This function is used to construct integrated species distribution models using data specified with [structured\\_data](#) as well as present only data obtained from the Global Biodiversity Information Facility (GBIF).

### Usage

```
species_model(
  speciesNames,
  date = NULL,
  structuredData = NULL,
  spatialCovariates = NULL,
  worldclimCovariates = NULL,
  res = 0.5,
  scale = FALSE,
  location = "Norway",
  boundary = NULL,
  return = "predictions map",
  mesh = NULL,
```

```

    meshParameters = NULL,
    spdeModel = NULL,
    projection = CRS("+proj=utm +zone=32 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"),
    limit = 10000,
    options = list(),
    ...
)

```

## Arguments

**speciesNames** A vector of species' names to collect from GBIF.

**date** Vector of length two denoting the date range to select species from. Defaults to NULL.

**structuredData** Additional datasets to integrate with the presence only GBIF data. See the `structured_data` function. Defaults to NULL.

**spatialCovariates** Spatial covariates to include in the model. May be a Raster or Spatial object. Cannot be non-NULL if `worldclimCovariates` is non-NULL.

**worldclimCovariates** Names of the covariates to extract from Worldclim. Defaults to NULL; cannot be non-NULL if `spatialCovariates` is non-NULL.

**res** Resolution for the world clim covariates. Valid values are: 0.5, 2.5, 5, 10. Defaults to 0.5.

**scale** Should the spatial covariates be scaled. Defaults to FALSE.

**location** Which area of Norway to model. Defaults to 'Norway' which suggests a model for the entire county.

**boundary** SpatialPolygons object of the study area. If NULL an object may be formed with `location`.

**return** Object to return. Has to be one of `c('boundary', 'species', 'species plot', 'mesh', 'mesh plot', 'model', 'predictions', 'predictions map')`.

**mesh** An `inla.mesh` object to include in the model. Defaults to NULL.

**meshParameters** A list of `inla.mesh` arguments to create a mesh if `mesh = NULL`.

**spdeModel** `inla.spde` model used in the model. May be a named list where the name of the `spde` object is the name of the associated dataset. Default NULL uses `inla.spde2.matern`.

**projection** CRS projection to use. Defaults to `CRS('+proj=utm +zone=32 +ellps=WGS84 +datum=WGS84 +units=m +no_defs')`.

**limit** Set the number of species downloaded. Defaults to 10000.

**options** A list of INLA and `inlabru` options. Defaults to NULL.

**...** Additional arguments used in `PointedSDMs`'s `intModel` function.

**Value**

The return of the function is determined by the argument `return`. For the different values of `return`:

<code>boundary</code>	A <code>SpatialPolygon</code> of the boundary used in the model,
<code>species</code>	A <code>data.frame</code> object of the species' coordinates used in the model,
<code>species plot</code>	a <code>ggplot</code> plot of the species across a map,
<code>mesh</code>	an <code>inla.mesh</code> object,
<code>mesh plot</code>	a plot of the <code>inla.mesh</code> object,
<code>model</code>	the integrated model,
<code>predictions</code>	predictions from the integrated model (on the linear scale),
<code>predictions map</code>	A <code>ggplot</code> plot of the predictions across the boundary.

**Examples**

```
{

## Not run:
if(requireNamespace('INLA')) {

#Objects required for example
data("PA_redlist")
speciesNames <- c('Fraxinus excelsior', 'Ulmus glabra')

#Set up structured dataset
dataObj <- structured_data(PA_redlist, datasetType = c('PA'), responsePA = 'individualCount',
  speciesName = 'species',
  coordinateNames = c("longitude", "latitude" ))

#Get species map
predictions <- species_model(return = 'species plot',
  boundary = boundary, speciesNames = species,
  limit = 10, structuredData = dataObj,
  meshParameters = list(cutoff=0.08, max.edge=c(1, 3), offset=c(1,1)),
  worldclimCovariates = 'Annual Mean Temperature')
}

## End(Not run)

}
```

---

structuredData-class    *bru\_sdm\_data class*

---

### Description

bru\_sdm\_data class

### Value

An S4 object with three slots, which contain lists of the data for the three observation models allowed in the modelling framework.

### Slots

dataPO A list of the present only datasets.

dataPA A list of the present absence datasets.

dataCount A list of the count datasets.

---

structured\_data            *structured\_data: function to organize structured datasets.*

---

### Description

The function is used to convert a collection of structured datasets into an object which can be used in `speciesModel`. The three types of structured data allowed by this function are present absence (PA), present only (PO) and counts/abundance datasets, which are controlled using the `datasetType` argument. The other arguments of this function are used to specify the appropriate variable (such as response name, trial name, species name and coordinate name) names in these datasets.

### Usage

```
structured_data(
  ...,
  datasetType = NULL,
  responsePA = NULL,
  trialsPA = NULL,
  responseCount = NULL,
  speciesName = "species",
  coordinateNames = c("longitude", "latitude")
)
```

**Arguments**

...	The datasets used in the model. May be either datasets or a SpatialPoints-DataFrames.
datasetType	A vector which gives the type of dataset. Must be either 'count', 'PO' or 'PA'. Defaults to NULL.
responsePA	Name of the response variable in the PA datasets. Defaults to NULL.
trialsPA	Name of the trial name variable in the PA datasets. Defaults to NULL.
responseCount	Name of the response variable in the count datasets. Defaults to NULL.
speciesName	Name of the species variable name. Defaults to 'species'.
coordinateNames	Name of the coordinates used in the model. Defaults to c('longitude', 'latitude').

**Value**

An S4 object of class `structuredData`, which contains three slots for the data of each observation model allowed in the framework.

**Examples**

```
{  
  
  #Objects required for example  
  data("PA_redlist")  
  speciesNames <- c('Fraxinus excelsior', 'Ulmus glabra')  
  
  #Set up structured dataset  
  dataObj <- structured_data(PA_redlist, datasetType = c('PA'), responsePA = 'individualCount',  
                             speciesName = 'species',  
                             coordinateNames = c("longitude", "latitude" ))  
  
}
```

# Index

## \* **data**

PA\_redlist, [2](#)

listorNULL-class, [2](#)

PA\_redlist, [2](#)

show, structuredData-method, [3](#)

species\_model, [3](#)

structured\_data, [3](#), [6](#)

structuredData-class, [6](#)