

# Package ‘trocSDM’

October 10, 2022

**Title** Including known interactions in SDM

**Version** 0.0.0.9000

**Author** Giovanni Poggiato and J  r  my Andr  oletti

**Maintainer** Giovanni Poggiato <giov.poggiato@gmail.com>

**Description** Package to fit a trophic Species Distribution Model, analyse it and predict. See Poggiato et al. In prep.

**License** GPL-3

**Encoding** UTF-8

**Imports** igraph, parallel, rstanarm, dplyr, rstan, arm, glmnet, formula.tools, abind, brms, piecewiseSEM, stringr, broom, jtools, ggstance, ggplot2, gridExtra, dismo, stats, grDevices, Matrix

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.2

## R topics documented:

buildFormula . . . . .	2
coef.SDMfit . . . . .	2
coef.trophicSDMfit . . . . .	3
computeVariableImportance . . . . .	4
compute_TL_laplacian . . . . .	5
evaluateModelFit . . . . .	5
G . . . . .	6
loo.trophicSDMfit . . . . .	6
plot.SDMfit . . . . .	7
plot.trophicSDMfit . . . . .	7
plotG . . . . .	8
plotG_inferred . . . . .	8
predict.SDMfit . . . . .	9
predict.trophicSDMfit . . . . .	10
predictFundamental . . . . .	11
print.SDMfit . . . . .	12
print.trophicSDMfit . . . . .	12
SDMfit . . . . .	13
summary.SDMfit . . . . .	15
summary.trophicSDMfit . . . . .	15
trocSDM . . . . .	16
trophicSDM . . . . .	16

trophicSDM_CV	19
X	20
Y	20

<b>Index</b>	<b>22</b>
--------------	-----------

---

buildFormula	<i>Builds SDM formulae</i>
--------------	----------------------------

---

## Description

Builds the formula of both the abiotic and biotic terms to fit a single species SDM based on the input parameters. The function is called inside the SDMfit function

## Usage

```
buildFormula(
  form.init,
  species,
  sp.formula = NULL,
  sp.partition = NULL,
  useBRMS
)
```

## Arguments

form.init	The abiotic part of the formula
species	The preys (or predators) of the focal species
sp.formula	optional parameter for composite variables. See ?trophicSDM
sp.partition	optional parameter to specify groups of species for composite variables. See ?trophicSDM
useBRMS	whether brms is used (TRUE if penal = "coeff.signs" and method = "stan_glm").

## Author(s)

Giovanni Poggiato and J  r  my Andr  oletti

---

coef.SDMfit	<i>Gets regression coefficients from a local model, i.e. a SDMfit object.</i>
-------------	---

---

## Description

Gets regression coefficients (eventually standardised) of a local model, i.e. a SDMfit object. p-values or credible intervals are returned when available.

## Usage

```
## S3 method for class 'SDMfit'
coef(object, standardise = F, level = 0.95, ...)
```

**Arguments**

object	A SDMfit object, typically obtained with trophicSDM() and available in the field \$model of a trophicSDMfit object
standardise	Whether to standardise regression coefficients. Default to FALSE. If TRUE, coefficients are standardised using the latent variable standardisation (see Grace et al. 2018) for more details.
level	The confidence level of credible intervals, only available for stan_glm method. Default to 0.95.
...	additional arguments

**Value**

A table containing the inferred coefficients (with credible intervals or p-values when available).

**Author(s)**

Giovanni Poggiato

**References**

Grace, J. B., Johnson, D. J., Lefcheck, J. S., and Byrnes, J. E. K.. 2018. Quantifying relative importance: computing standardized effects in models with binary outcomes. *Ecosphere* 9(6):e02283.

---

coef.trophicSDMfit	<i>Gets regression coefficients from a fitted trophicSDM model.</i>
--------------------	---

---

**Description**

Gets regression coefficients (eventually standardised) of a fitted trophicSDM. p-values or credible intervals are returned when available.

**Usage**

```
## S3 method for class 'trophicSDMfit'
coef(object, standardise = F, level = 0.95, ...)
```

**Arguments**

object	A trophicSDMfit object obtained with trophicSDM()
standardise	Whether to standardise regression coefficients. Default to FALSE. If TRUE, coefficients are standardised using the latent variable standardisation (see Grace et al. 2018) for more details.
level	The confidence level of credible intervals, only available for stan_glm method. Default to 0.95.
...	additional arguments

**Value**

A list containing, for each species, the inferred coefficients (with credible intervals or p-values when available).

**Author(s)**

Giovanni Poggiato

**References**

Grace, J. B., Johnson, D. J., Lefcheck, J. S., and Byrnes, J. E. K.. 2018. Quantifying relative importance: computing standardized effects in models with binary outcomes. *Ecosphere* 9(6):e02283.

---

computeVariableImportance

*Computes variable importance of (groups of) variables of fitted a trophicSDM model.*

---

**Description**

Computes variable importance of (groups of) variables of fitted a trophicSDM model, for each species. Variable importance are computed as the standardised regression coefficients (summed across species of the same group). Standardisation is done using latent variable standardisation described in Grace et al. 2018.

**Usage**

```
computeVariableImportance(tSDM, groups = NULL)
```

**Arguments**

tSDM	A trophicSDMfit object obtained with trophicSDM()
groups	A list where each element is group. Each group is specified as a vector containing species or environmental covariates names of a given group. Each element of the list (i.e. each group) has to be named.

**Value**

A groups x species matrix containing variable importance for each groups of variables and each species.

**Author(s)**

Giovanni Poggiato #' @references Grace, J. B., Johnson, D. J., Lefcheck, J. S., and Byrnes, J. E. K.. 2018. Quantifying relative importance: computing standardized effects in models with binary outcomes. *Ecosphere* 9(6):e02283.

---

compute_TL_laplacian	<i>Compute topological ordering</i>
----------------------	-------------------------------------

---

**Description**

Method to compute trophic levels from an igraph object G with the method described in MacKay et al 2020.

**Usage**

```
compute_TL_laplacian(G)
```

**Arguments**

G	The metaweb, it has to be an igraph object
---	--

**Author(s)**

Giovanni Poggiato

**References**

MacKay, R. S., Johnson, S., & Sansom, B. (2020). How directed is a directed network?. Royal Society open science, 7(9), 201138.

---

evaluateModelFit	<i>Evaluates prediction goodness of fit</i>
------------------	---

---

**Description**

Evaluate goodness of fit by comparing a true versus a predicted dataset of species distribution. Ypredicted is typically predicted using a prediction method of trophicSDM (in cross-validation if trophicSDM\_CV() is used).

**Usage**

```
evaluateModelFit(tSDM, Ynew = NULL, Ypredicted = NULL)
```

**Arguments**

tSDM	A trophicSDMfit object obtained with trophicSDM().
Ynew	A sites x species matrix containing the true species occurrences state.
Ypredicted	A sites x species matrix containing the predicted species occurrences state.

**Value**

A table specifying the goodness of fit metrics for each species. For presence-absence data, the model computes TSS and AUC. For Gaussian data, the R2.

**Author(s)**

Giovanni Poggiato

**References**

Grace, J. B., Johnson, D. J., Lefcheck, J. S., and Byrnes, J. E. K.. 2018. Quantifying relative importance: computing standardized effects in models with binary outcomes. *Ecosphere* 9(6):e02283.

G

*Simulated environemntal covariates G***Description**

Simulated environemntal covariates G

**Usage**

data(G)

**Format**

A simulated graph of trophic interactions G

**Author(s)**

Giovanni Poggiato

**Examples**

data(G)

loo.trophicSDMfit

*Computes an approximation of loo for the whole model***Description**

The global loo is computed by summing the loo of all the local models (since the likelihood factorises, the log-likelihood can be summed)

**Usage**

```
## S3 method for class 'trophicSDMfit'
loo(tSDM)
```

**Arguments**

tSDM                      A trophicSDMfit object obtained with trophicSDM()

**Author(s)**

Giovanni Poggiato

---

plot.SDMfit	<i>Plots the regression coefficients of a local model</i>
-------------	---

---

**Description**

Plots the regression coefficients of a local SDMfit model

**Usage**

```
## S3 method for class 'SDMfit'  
plot(x, level = 0.95, ...)
```

**Arguments**

x	A SDMfit object, typically obtained with trophicSDM() and available in the field \$model of a trophicSDMfit object
level	the confidence level of the confidence intervals
...	additional arguments

**Author(s)**

Giovanni Poggiato

---

plot.trophicSDMfit	<i>Plots the regression coefficients of a fitted trophicSDM model</i>
--------------------	---

---

**Description**

Plots the regression coefficients of a fitted trophicSDM model. A subset of species to be plotted can be specified in the parameterspecies.

**Usage**

```
## S3 method for class 'trophicSDMfit'  
plot(x, species = NULL, ...)
```

**Arguments**

x	A trophicSDMfit object obtained with trophicSDM()
species	A vector of species names to be plot. If NULL (default), all species are plotted.
...	additional arguments

**Author(s)**

Giovanni Poggiato

---

plotG	<i>Plots the metaweb G</i>
-------	----------------------------

---

**Description**

Plots the metaweb G used to fit the trophicSDM model

**Usage**

```
plotG(tSDM)
```

**Arguments**

tSDM                      A trophicSDMfit object obtained with trophicSDM()

**Value**

A ggnet object

**Author(s)**

Giovanni Poggiato

---

plotG_inferred	<i>Plot the metaweb G according to the inferred coefficients</i>
----------------	--

---

**Description**

Plot the metaweb G with links colored accordingly to the inferred prey-predator regression coefficients of a fitted trophicSDM model. Plots the metaweb G, where each predator-prey link is colored according to whether the related regression coefficient is inferred as positive (in red), negative (in blue) or non-significant (dashed grey line) according to the confidence level specified in "level". Estimates of the significant standardised regression coefficients are pasted on the links. Only works if species are modelled as a function of their preys or predators without composite variables (i.e., if tSDM is fitted with sp.formula = NULL and sp.partition = NULL)

**Usage**

```
plotG_inferred(tSDM, level = 0.9)
```

**Arguments**

tSDM                      A trophicSDMfit object obtained with trophicSDM()  
 level                    The confidence level used to decide whether regression coefficients are non-significant or not. Default to 0.9.

**Value**

A ggnet object.



**Author(s)**

Giovanni Poggiato

---

predict.SDMfit	<i>Predicts with a local model</i>
----------------	------------------------------------

---

**Description**

Computes predicted values for a local model, i.e., a fitted SDMfit object This is sequentially called, for each species, by the function trophicSDM.predict

**Usage**

```
## S3 method for class 'SDMfit'
predict(object, newdata, pred_samples, prob.cov, ...)
```

**Arguments**

object	A SDMfit object, typically obtained with trophicSDM() and available in the field \$model of a trophicSDMfit object
newdata	A matrix containing both environmental covariates and the biotic variables that the local model uses to predict the species distribution.
pred_samples	Number of samples to draw from species posterior predictive distribution when method = "stan_glm". If NULL, set by the default to the number of iterations/10.
prob.cov	If set to FALSE, it gives back also predicted presence-absences (which is then used by trophicSDM.predict to predict the predators).
...	additional arguments

**Value**

A list containing for each species the predicted value at each sites. If method = "stan\_glm", then each element of the list is a sites x pred\_samples matrix containing the posterior predictive distribution of the species at each sites.

**Author(s)**

Giovanni Poggiato and J  r  my Andr  oletti

---

predict.trophicSDMfit *Computes predicted values from the fitted trophicSDMfit model*

---

## Description

Computes predicted values from the fitted trophicSDMfit model at environmental conditions specified by Xnew. Once predictions have been obtained, their quality can eventually be evaluated with evaluateModelFit().

## Usage

```
## S3 method for class 'trophicSDMfit'
predict(
  object,
  Xnew = NULL,
  prob.cov = F,
  pred_samples = NULL,
  run.parallel = T,
  verbose = F,
  fullPost = T,
  filter.table = NULL,
  ...
)
```

## Arguments

object	A trophicSDMfit object obtained with trophicSDM()
Xnew	a matrix specifying the environmental covariates for the predictions to be made. If NULL (default), predictions are done on the training dataset (e.g. by setting Xnew = tSDM\$data\$X).
prob.cov	Parameter to predict with trophicSDM with presence-absence data. Whether to use predicted probability of presence (prob.cov = T) or the transformed presence-absences (default, prob.cov = F) to predict species distribution.
pred_samples	Number of samples to draw from species posterior predictive distribution when method = "stan_glm". If NULL, set by the default to the number of iterations/10.
run.parallel	Whether to use parallelise code when possible. Can speed up computation time.
verbose	Whether to print advances of the algorithm
fullPost	Optional parameter for stan_glm only. Whether to give back the full posterior predictive distribution (default, fullPost = TRUE) or just the posterior mean, and 2.5% and 97.5% quantiles,
filter.table	Optional, default to NULL, should be provided only if the users wants to filter some species predictions. A sites x species matrix of zeros and ones.
...	additional arguments

## Value

A list containing for each species the predicted value at each sites. If method = "stan\_glm", then each element of the list is a sites x pred\_samples matrix containing the posterior predictive distribution of the species at each sites.



---

print.SDMfit	<i>Prints a SDMfit object</i>
--------------	-------------------------------

---

**Description**

Prints a SDMfit object

**Usage**

```
## S3 method for class 'SDMfit'
print(x, ...)
```

**Arguments**

x	A SDMfit object, typically obtained with trophicSDM() and available in the field \$model of a trophicSDMfit object
...	additional arguments

**Author(s)**

Giovanni Poggiato

---

print.trophicSDMfit	<i>Prints a fitted trophicSDM model</i>
---------------------	---

---

**Description**

Prints a fitted trophicSDM model

**Usage**

```
## S3 method for class 'trophicSDMfit'
print(x, ...)
```

**Arguments**

x	A trophicSDMfit object obtained with trophicSDM()
...	additional arguments

**Author(s)**

Giovanni Poggiato

SDMfit

*Fitting a single-species SDM***Description**

SDMfit is used to fit a single species SDM, what we call a 'local model' of trophicSDM. It returns an object of class 'SDMfit'. Requires basically the same inputs of trophicSDM, with the requirement to specify with the parameter 'focal' the species that is modelled by the SDMfit.

**Usage**

```
SDMfit(
  focal,
  Y,
  X,
  G,
  formula.foc,
  sp.formula,
  sp.partition,
  mode = "prey",
  method = "stan_glm",
  family = NULL,
  penal = NULL,
  iter = 1000,
  chains = 2,
  verbose = T
)
```

**Arguments**

focal	the name of the species to be modeled
Y	The sites x species matrix containing observed species distribution (e.g. presence-absence).
X	The design matrix, i.e. sites x predictor matrix containing the value of each explanatory variable (e.g. the environmental conditions) at each site.
G	The species interaction network (aka metaweb). Needs to be an igraph object. Links must go from predator to preys. It needs to be a directed acyclic graph.
formula.foc	The formula for the abiotic part of the species distribution model.
sp.formula	(optional) It allows to specify a particular definition of the biotic part of the model, e.g., using composite variables (e.g., richness), or an interaction of the biotic and abiotic component. More details in 'Details'.
sp.partition	(optional) a list to specify groups of species that are used to compute composite variables, e.g., a species can be modelled as a function of the richness of each group of preys. It has to be a list, each element is a vector containing the names of species in the group.
mode	"prey" if bottom-up control (default), "predators" otherwise. Notice that G needs to be such that links point from predators to prey.

method	which SDM method to use. For now the available choices are: "glm" (frequentist) or "stan_glm" (full bayesian MCMC, default). Notice that using "glm" does not allow error propagation when predicting.
family	the family parameter of the glm function (see glm). family=gaussian for gaussian data or family=binomial(link = "logit") or binomial(link = "probit") for presence-absence data.
penal	(optional, default to NULL) Penalisation method to shrink regression coefficients. If NULL (default), the model does not penalise the regression coefficient. For now, available penalisation method are "horshoe" for stan_glm, "elasticnet" for glm and "coeff.signs" (prey coefficients are set to positive and predator coefficients to negative) for glm and stan_glm.
iter	(for method="stan_glm" only) Number of iterations for each MCMC chain if stan_glm is used
chains	(for method="stan_glm" only) Number of MCMC chains (default to 2)
verbose	Whether to print algorithm progresses

### Details

"sp.formula" and "sp.partition" can be combined to define any kind of composite variables for the biotic part of the formula. "sp.formula" can be :

- A string defining a formula as function of "richness". E.g., sp.formula="richness+I(richness)^2" (species are modelled as a function of a quadratic polyome of their prey richness), "I(richness>0)" (species are modelled as a function of a dummy variable that is equal to 1 when at least one species is present). Importantly, when group of preys (or predators) are specified by "sp.partition", species are modeled as a function of the composite variable specified by "sp.formula" for each of their prey groups.
- A more flexible option is to specify sp.formula as a list (whose names are species' names) that contains for each species the definition biotic part of the model. Notice that, in this case, the function does not check that the model is a DAG. This allow to define any kind of composite variable, or to model interactions between environmental covariates and preys (or predators).

### Value

A "SDMfit" object, containing:

model	The output of the function used to fit the SDM. E.g., an object of class "glm" is method = "glm", an object of class "stanreg" if method = "stan_glm".
Y	A numeric vector of standard errors on parameters
form.all	The formula used to fit the SDM (both abiotic and biotic terms)
method, family, penal, iter, chains	The input parameters used to fit the SDM.
sp.name	The name of the species modeled
data	The model.frame data.frame used to fit the model
coef	The inferred coefficients (with credible intervals or p-values when available)
AIC	The AIC of the local model
log.lik	The log.likelihood of the local model

**Author(s)**

Giovanni Poggiato and Jérémy Andréletti

---

summary.SDMfit	<i>Summary of a fitted SDMfit model</i>
----------------	---

---

**Description**

Summary of a fitted SDMfit model

**Usage**

```
## S3 method for class 'SDMfit'  
summary(object, ...)
```

**Arguments**

object	A SDMfit object, typically obtained with trophicSDM() and available in the field \$model of a trophicSDMfit object
...	additional arguments

**Author(s)**

Giovanni Poggiato

---

summary.trophicSDMfit	<i>Summary of a fitted trophicSDM model</i>
-----------------------	---

---

**Description**

Summary of a fitted trophicSDM model

**Usage**

```
## S3 method for class 'trophicSDMfit'  
summary(object, ...)
```

**Arguments**

object	A trophicSDMfit object obtained with trophicSDM()
...	additional arguments

**Author(s)**

Giovanni Poggiato

trocsDM

*trocsDM.***Description**

Package to fit a trophic Species Distribution Model, analyse it and predict. See Poggiato et al. In prep.

**Author(s)**

Giovanni Poggiato <giov.poggiato@gmail.com> and J  r  my Andr  oletti

trophicSDM

*Fitting a trophic Species distribution model***Description**

trophicSDM is used to fit a trophic species distribution model. Requires the species distribution data  $Y$  (the sites  $\times$  species matrix), explanatory variables  $X$  and a directed acyclic graph  $G$  containing species interactions (i.e., the metaweb, with links going from predators to prey). The function fits the distribution of each species as a function of their preys (with `mode = "prey"`, by default) or predators (if set `mode = "predator"`).

**Usage**

```
trophicSDM(
  Y,
  X,
  G,
  env.formula = NULL,
  sp.formula = NULL,
  sp.partition = NULL,
  penal = NULL,
  mode = "prey",
  method = "stan_glm",
  family,
  iter = 1000,
  chains = 2,
  run.parallel = F,
  verbose = F
)
```

**Arguments**

$Y$	The sites $\times$ species matrix containing observed species distribution (e.g. presence-absence).
$X$	The design matrix, i.e. sites $\times$ predictor matrix containing the value of each explanatory variable (e.g. the environmental conditions) at each site.



G	The species interaction network (aka metaweb). Needs to be an igraph object. Links must go from predator to preys. It needs to be a directed acyclic graph.
env.formula	The definition of the abiotic part of the model. It can be : <ul style="list-style-type: none"> <li>• a string specifying the formula (e.g. "<math>\sim X_1 + X_2</math>"). In this case, the same environmental variables are used for every species.</li> <li>• A list that contains for each species the formula that describes the abiotic part of the model. In this case, different species can be modeled as a function of different environmental covariates. The names of the list must coincide with the names of the species.</li> </ul>
sp.formula	(optional) It allows to specify a particular definition of the biotic part of the model, e.g., using composite variables (e.g., richness), or an interaction of the biotic and abiotic component. More details in 'Details'.
sp.partition	(optional) a list to specify groups of species that are used to compute composite variables, e.g., a species can be modelled as a function of the richness of each group of preys. It has to be a list, each element is a vector containing the names of species in the group. More details in 'Details'.
penal	Penalisation method to shrink regression coefficients. If NULL (default), the model does not penalise the regression coefficient. For now, available penalization method are "horshoe" for method stan_glm, "elasticnet" for method glm. It is also possible to constrain the sign of biotic coefficients (prey coefficients are set to positive and predator coefficients to negative) by setting "coeff.signs" for methods glm and stan_glm.
mode	"prey" if bottom-up control (default), "predators" otherwise. Notice that G needs to be such that links point from predators to prey.
method	which SDM method to use. For now the available choices are: "glm" (frequentist) or "stan_glm" (full bayesian MCMC, default). Notice that using "glm" does not allow error propagation when predicting.
family	the family parameter of the glm function (see glm). gaussian for gaussian data. binomial(link = "logit") or binomial(link = "probit") for presence-absence data.
iter	(for "stan_glm" only) Number of iterations for each MCMC chain if stan_glm is used
chains	(for "stan_glm" only) Number of MCMC chains (default to 2)
run.parallel	Whether species models are fitted in parallel (can speed computational up time). Default to FALSE.
verbose	Whether to print algorithm progresses

## Details

"sp.formula" and "sp.partition" can be combined to define any kind of composite variables for the biotic part of the formula. "sp.formula" can be :

- A string defining a formula as function of "richness". E.g., " $\text{richness} + \text{I}(\text{richness})^2$ " (species are modelled as a function of a quadratic polyome of their prey richness), " $\text{I}(\text{richness} > 0)$ " (species are modelled as a function of a dummy variable that is equal to 1 when at least one species is present). Importantly, when group of preys (or predators) are specified by "sp.partition", species are modeled as a function of the composite variable specified by "sp.formula" for each of their prey (or predator) groups.

- A more flexible option is to specify `sp.formula` as a list (whose names are species' names) that contains for each species the definition of the biotic part of the model. Notice that, in this case, the function does not check that the model is a DAG. This allow to define any kind of composite variable, or to model interactions between environmental covariates and preys (or predators).

## Value

A "trophicSDMfit" object, containing:

<code>model</code>	A list containing the local models (i.e. a SDM for each species). Each local model is an object of class "SDMfit". See <code>?SDMfit</code> for more informations.
<code>Y</code>	A numeric vector of standard errors on parameters
<code>form.all</code>	A list describing each species formula (both biotic and abiotic terms)
<code>data</code>	A list containing all the data used to fit the model
<code>model.call</code>	A list containing the modeling choices of the fitted model (e.g. method, penalisation...)
<code>coef</code>	A list containing, for each species, the inferred coefficients (with credible intervals or p-values when available)
<code>MCMC.diag</code>	MCMC convergence metrics, only available for MCMC methods
<code>AIC</code>	Model's AIC
<code>log.lik</code>	Model's log.likelihood

## Author(s)

Giovanni Poggiato and J  r  my Andr  letti

## Examples

```
data(Y)
data(X)
data(G)
# define abiotic part of the model
env.formula = "~ X_1 + X_2"
# Run the model with bottom-up control using stan_glm as fitting method and no penalisation
m = trophicSDM(Y,X,G, env.formula,
               family = binomial(link = "logit"), penal = NULL,
               mode = "prey", method = "stan_glm")
print(m)

# Access local models (e.g. species "Y5")
m$model$Y5
coef(m$model$Y5)
plot(m$model$Y5)
```

---

trophicSDM_CV	<i>Compute K-fold cross-validation predicted values from a fitted trophicSDM model</i>
---------------	--

---

### Description

Once the CV predicted values are obtained, their quality can be evaluated with `evaluateModelFit()`.

### Usage

```
trophicSDM_CV(
  tSDM,
  K,
  partition = NULL,
  prob.cov = F,
  pred_samples = NULL,
  iter = NULL,
  chains = NULL,
  run.parallel = T,
  verbose = F
)
```

### Arguments

tSDM	A trophicSDMfit object obtained with trophicSDM()
K	The number of folds for the K-fold cross validation
partition	Optional parameter. A partition vector to specify a partition in K fold for cross validation
prob.cov	Parameter to predict with trophicSDM with presence-absence data. Whether to use predicted probability of presence (prob.cov = T) or the transformed presence-absences (default, prob.cov = F) to predict species distribution.
pred_samples	Number of samples to draw from species posterior predictive distribution when method = "stan_glm". If NULL, set by the default to the number of iterations/10.
iter	For method = "stan_glm": number of iterations of each MCMC chains to fit the trophicSDM model. Default to the number of iterations used to fit the provided trophicSDMfit object
chains	For method = "stan_glm": number of MCMC chains to fit the trophicSDM model. Default to the number of iterations used to fit the provided trophicSDMfit object
run.parallel	Whether to use parallelise code when possible. Default to TRUE. Can speed up computation time
verbose	Whether to print advances of the algorithm

### Value

A list containing:

meanPred	a sites x species matrix of predicted occurrences of species for each site (e.g. probability of presence). With stan_glm the posterior predictive mean is return
----------	--

Pred975, Pred025

Only for method = "stan\_glm", the 97.5% and 2.5% quantiles of the predictive posterior distribution

partition the partition vector used to compute the K fold cross-validation

### Author(s)

Giovanni Poggiato

---

X	<i>Simulated environemntal covariates X</i>
---	---

---

### Description

Simulated environemntal covariates X

### Usage

data(X)

### Format

A site x covariates matrix X

### Author(s)

Giovanni Poggiato

### Examples

data(X)

---

Y	<i>Simulated species distribution Y</i>
---	---

---

### Description

Simulated species distribution Y

### Usage

data(Y)

### Format

A site x species matrix Y, a site x covariates matrix X and a trophic interaction network G (object igraph)

### Author(s)

Giovanni Poggiato

**Examples**`data(Y)`

# Index

## \* datasets

G, [6](#)

X, [20](#)

Y, [20](#)

buildFormula, [2](#)

coef.SDMfit, [2](#)

coef.trophicSDMfit, [3](#)

compute\_TL\_laplacian, [5](#)

computeVariableImportance, [4](#)

evaluateModelFit, [5](#)

G, [6](#)

loo.trophicSDMfit, [6](#)

plot.SDMfit, [7](#)

plot.trophicSDMfit, [7](#)

plotG, [8](#)

plotG\_inferred, [8](#)

predict.SDMfit, [9](#)

predict.trophicSDMfit, [10](#)

predictFundamental, [11](#)

print.SDMfit, [12](#)

print.trophicSDMfit, [12](#)

SDMfit, [13](#)

summary.SDMfit, [15](#)

summary.trophicSDMfit, [15](#)

trocSDM, [16](#)

trophicSDM, [16](#)

trophicSDM\_CV, [19](#)

X, [20](#)

Y, [20](#)