Package 'siland'

May 9, 2026

Type Package

Title Spatial Influence of Landscape

Version 3.0.2

Date 2023-03-23

Author Carpentier F. and Martin O.

Maintainer Martin Olivier <olivier.martin@inrae.fr>

Description Method to estimate the spatial influence scales of landscape variables on a response variable. The method is based on Chandler and Hepinstall-Cymerman (2016) Estimating the spatial scales of landscape effects on abundance, Landscape ecology, 31: 1383-1394, <doi:10.1007/s10980-016-0380-z>.

License GPL (>= 2.0) | file LICENSE

Depends base, graphics, stats, sf, R (>= 3.5.0)

Imports lme4, sp, raster, ggplot2, ggforce, fasterize, reshape2, fields, methods

NeedsCompilation no

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.1.2

Repository CRAN

Date/Publication 2023-03-30 14:20:12 UTC

Contents

`siland-package` *Spatial Influence of Landscape*

Description

Method to estimate the spatial influence scales of landscape variables on a response variable. The method is based on Chandler and Hepinstall-Cymerman (2016) Estimating the spatial scales of landscape effects on abundance, Landscape ecology, 31: 1383-1394, <doi:10.1007/s10980-016-0380-z>. This package allows for analyzing the effect of landscape features on georeferenced point observations (described in a Geographic Information System shapefile format). It simultaneously estimates the spatial scales and intensities of landscape variable effects. It does not require any information about the scale of effect. It allows for effects tests, effects maps and models comparison. \

Details

The DESCRIPTION file:

Package:	siland
Type:	Package
Title:	Spatial Influence of Landscape
Version:	3.0.2
Date:	2023-03-23
Author:	Carpentier F. and Martin O.
Maintainer:	Martin Olivier <olivier.martin@inrae.fr>
Description:	Method to estimate the spatial influence scales of landscape variables on a response variable. The metho
License:	GPL (>=2.0) | file LICENSE
Depends:	base, graphics, stats, sf, R (>= 3.5.0)
Imports:	lme4, sp, raster, ggplot2, ggforce, fasterize, reshape2, fields, methods
NeedsCompilation:	no
Packaged:	2019-11-13 12:46:55 UTC; omartin
Suggests:	knitr, rmarkdown
VignetteBuilder:	knitr
RoxygenNote:	7.1.2

Index of help topics:

Author(s)

Carpentier F. and Martin O. Maintainer: Martin Olivier <olivier.martin@inrae.fr>

`AIC.siland` *Akaike's An Information Criterion for siland package*

Description

Function calculating thue usual Akaike's 'An Information Criterion' for an object of class siland, according to the formula -2*loglikelihood+k*npar, with npar the number of parameters of the model and k = 2.

Usage

```
## S3 method for class 'siland'
AIC(object,...,k=2)
```

Arguments

`object`	an object of class siland
`...`	Dots are not take into account in this version
`k`	not used

Value

AIC value

Author(s)

O. Martin

References

Sakamoto, Y., Ishiguro, M., and Kitagawa G. (1986). Akaike Information Criterion Statistics. D. Reidel Publishing Company.

`BIC.siland` *Bayesian Information criterion*

Description

Function computing the Bayesian Information Criterion for an object of class siland, according to the formula -2*loglikelihood+k*npar, with npar the number of parameters of the model and k = log(n) (n being the number of observations).

Usage

```
## S3 method for class 'siland'
BIC(object,...)
```

Arguments

`object`	an object of class siland
`...`	Dots are not take into account in this version

Value

BIC value

Author(s)

O. Martin

`bufferforsiland` *Percentage of landscape variable in buffers.*

Description

Function computing the percentages of landscape variables in buffers of given sizes (i.e. radii).

Usage

```
bufferforsiland(d, sfGIS, loc.sf, landnames, border = F)
```

Arguments

`d`	a vector of the distances of the buffer sizes. The length of `d` has to be equal to that of `landnames`.
`sfGIS`	an object of class sf containing the landscape variables (land uses). Landscape variables are equal to 1 (presence) or 0 (absence).
`loc.sf`	an object of class sf or a dataframe containing locations of buffers centers. For dataframe, locations have to be given in columns "X" and "Y".
`landnames`	a vector of characters of the names of the landscape variables
`border`	a logical (`FALSE` by default). If `border=FALSE` buffers are computed from the locations points given in `loc.sf` If `border=TRUE` buffers are computed from the border of the polygon that contain the points in `loc.sf`.

Value

a dataframe containing the percentages of the landscape variables named `landnames` (in columns) in buffers of sizes described in `d` and centered on locations described in `loc.sf` (in rows).

Author(s)

Martin O.

Examples

```
data(dataSiland)
data(landSiland)
dataSiland[1:10,c("X","Y")]
landSiland$L1
landSiland$L2

res=bufferforsiland(c(100,200),sfGIS=landSiland, loc.sf=dataSiland, landnames=c("L1","L2"))
res
```

dataSiland *Simulated data*

Description

Simulated data of response variable and local variables.

Usage

```
data("dataSiland")
```

Format

A data frame with 100 observations on 5 variables.

[,1]	X	numeric	location on X-axis
[,2]	Y	numeric	location on Y-axis
[,3]	x1	numeric	a continous local variable
[,4]	Id	numeric	identifier
[,5]	obs	numeric	response variable

Examples

```
data(dataSiland)
plot(dataSiland[,c("X","Y")])
```

fitted.siland *Extract siland model Fitted values*

Description

Function extracting fitted values of an object of class siland.

Usage

```
## S3 method for class 'siland'
fitted(object,...)
```

Arguments

object	an object of class siland
...	Dots are not take into accounbt in this version

Value

Fitted values extracted from the object `object`.

Author(s)

O. Martin

`landSiland` *Simulated landscape.*

Description

Simulated landscape associated to dataSiland, containing two landscape variables, named L1 and L2.

Usage

```
data("landSiland")
```

Format

A `sf` object with 4884 observations on 3 variables.

[,1]	L1	numeric	presence/absence of L1 (0/1)
[,2]	L2	numeric	presence/absence of L2 (0/1)
[,3]	geometry	sfc_MULTIPOLYGON	polygon location

Examples

```
data(landSiland)
plot(landSiland$geometry)
plot(landSiland[landSiland$L1==1,]$geometry,col=2,add=TRUE)
plot(landSiland[landSiland$L2==1,]$geometry,col=3,add=TRUE)
```

`landtoraster` *Transformation of an sf object into raster*

Description

Function transforming an object of class sf into raster (points grid) with `wd`, the pixel size of the raster. `landname` gives the variable names for which raster points have to be computed.

Usage

```
landtoraster(landgis, landname, wd)
```

Arguments

landgis	an object of class sf
landname	vector of names of landscape variable for which rasters are computed.
wd	numeric, pixels size of raster

Value

a list of dataframes. The components of the list correspond to variables specifed in argument landname. Each dataframe gives the X and Y locations of pixels in raster.

Author(s)

O. Martin

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

data(dataSiland)
data(landSiland)
head(dataSiland)
head(landSiland)
u=landtoraster(landSiland,c("L1","L2"),wd=20)
names(u)
du1=as.data.frame(raster::rasterToPoints(u[[1]]))
du2=as.data.frame(raster::rasterToPoints(u[[2]]))
plot(du1[,1:2],type="n")
sel1=du1[,3]==1
sel2=du2[,3]==1
points(du1[sel1,c(1,2)],pch=".",col=1)
points(du2[sel2,c(1,2)],pch=".",col=2)
```

---

plot.siland *Plot an object of class siland*

Description

This function calls the function plot.lm() and plot the object result stored in x

Usage

```
## S3 method for class 'siland'
plot(x, ...)
```

Arguments

x	an object of class siland
...	the same options as for the function plot.lm()

`plotsiland` *Plot results from siland function*

Description

Plot locations of observed data and locations of landscape variables. The black points correspond to the locations of observed data, and the size is propotional to numeric values. The locations of different landscape variables are displayed with small points of different colours. Estimated mean distance of spatial influence for each landscape variable is indicated beside the graphic with a continuous line. Dashed line indicates the radius that gives 95 percent of the total influence of each landscape variable.

Usage

```
plotsiland(res, land, data)
```

Arguments

res	an object obtained from the function siland
land	an object of class sf that gives the landscape variables
data	a dataframe with the response variable and the local variables

Value

an object of class ggplot

Examples

```
data(dataSiland)
data(landSiland)
res=siland(obs~x1+L1+L2,land=landSiland,data=dataSiland,sif="exponential")
plotsiland(res,landSiland, dataSiland)
```

`plotsiland.land` *Map of siland result*

Description

Function creating a map of siland result, i.e. estimatied spatial influence of each landscape variable or for the cumulative effect of all variables.

Usage

```
plotsiland.land(x, land, data, var = 0, lw = 100, xlim=NULL, ylim=NULL,plot=T)
```

Arguments

`x`	an object resulting from `siland`
`land`	the object of class sf describing the landscape variable, i.e. the `land` argument in the call to the `siland` function that computed `x`
`data`	the dataframe describing response variable and the local variables, i.e. the `data` argument in the call to the siland function that computed `x`.
`var`	a vector containing the numbers (numeric) of the mapped landscape variables. If `var=0` (by default), the cumulative effect of all landscape variables is represented.
`lw`	the number of pixels on x-axis and y-axis for the graphical representation. A too high value can lead to memory allocation problems. By default lw=100.
`xlim`	a length-2 numeric vector, lower and upper limits of x-axis.
`ylim`	a length-2 numeric vector, lower and upper limits of y-axis.
`plot`	a logical (TRUE by default). If TRUE, the result is a graphic. If FALSE, the result is a dataframe of the cumulative spatial influence of the landscape variable(s) var.

Value

an object of class ggplot

See Also

siland

Examples

```
data(dataSiland)
data(landSiland)
res=siland(obs~x1+L1+L2,land=landSiland,data=dataSiland,sif="exponential")
plotsiland.land(x=res,land=landSiland,data=dataSiland)
#plotsiland.land(x=res,land=landSiland,data=dataSiland,var=1,lw=20)
#plotsiland.land(x=res,land=landSiland,data=dataSiland,var=2,lw=20)
```

plotsiland.sif *Plot the estimated spatial influence functions.*

Description

Graphic representations for the different influence functions estimated with the function siland().

Usage

```
plotsiland.sif(x)
```

Arguments

x	an object obtained with function siland().

Value

no value is returned

See Also

siland

Examples

```
data(dataSiland)
data(landSiland)
res=siland(obs~x1+L1+L2,land=landSiland,data=dataSiland,sif="exponential")
plotsiland.sif(x=res)
```

print.siland *siland print*

Description

function printing an object of class siland. It returns the model used and the estimated parameters.

Usage

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

Arguments

x	an object of class siland
...	not take into account in this version

residuals.siland *Extract siland Residuals*

Description

Function extracting residuals for an estimated siland model.

Usage

```
## S3 method for class 'siland'
residuals(object,...)
```

Arguments

object	an object of class siland
...	Dots are not take into account in this version

siland *Estimation of landscape influence using siland method (Spatial Influence Function)*

Description

Function estimating landscape effect (and eventual local effects) using siland method, i.e. estimating simultaneously : the SIF parameter (i.e. mean distance of SIF) and the effect of each landscape variable (and the effect of each local variable).

Usage

```
siland(formula,land,data,family ="gaussian",sif="exponential", init = 100,
 border=F,wd=50,maxD=3000)
```

Arguments

formula	a symbolic description (see lm() or glm()) of the model to be fitted. The details of model specification are given under 'Details'.
land	an object of class sf containing the landscape variables.
data	a dataframe containing the response variable and the local variables.
family	the distribution of response variable. family can be "gaussian", "poisson" or "binomial" and the associated link function are identity, log and logit respectively.
sif	the family of the Spatial Influence Function. sif can be "exponential", "gaussian" or "uniform".

`init`	a vector indicating the starting values for SIF parameters in the estimation procedure. By default, starting values of SIF parameters of all landscape variable are equal to 100.
`border`	a logical indicating wether the spatial influence of the polygon where observations are located is taken into account. If `border=FALSE`, all pixels of a landscape variable have a spatial influence whether or not they belong to the polygon where the observation is located. If `border=TRUE`, only pixels outside the polygon of the observation, i.e. from the border of the polygon have an influence on this observation.
`wd`	a numeric, the size of pixels.
`maxD`	a numeric, give the maximum distance used to evaluate the influence of pixel on each observation. It is recommanded that maxD is greater than three times the greatest SIF value.

Details

The siland method is based on the model:

$$E(Y_i) = \mu + \sum_l \alpha_l x_{l,i} + \sum_k \beta_k (\sum_r f_{\delta_k}(d_{r,i}) z_r^k)$$

with

- Y_i, response variable,
- x_l,i, local variable l,
- r, the discritzation of study site in pixels r
- z^k_r, the value of the landscape variable k at the pixel r. If the landscape variable k is a presence/absence variable, z^k_r is equal to one or zero.
- μ, the intercept
- α_l and β_k, the effects of local and landscape variables, respectively.
- f_{δ_k},the SIF associated with the landscape variable k with parameter δ_k (the mean distance of the SIF, see detail further)
- d_r,i the distance between the center of pixel r and the observation at site i.

All parameters, μ, $\alpha_1, \ldots, \alpha_L$, $\beta_1, \ldots, \beta_K$ but also $\delta_1, \ldots, \delta_K$ SIF parameters of the landscape variables are simultaneously estimated by likelihood maximization.

Models for *siland* have the form `response ~ terms` where `response` is the columns name of the response variable in `data` and `terms` is a series of terms which specifies the names of landscape and local explanatory variables. The explanatory variables described in `data` are considered as local in the model, those described in the sf object `land` are considered as landscape variables.

For local variables, random effects can be considered using the syntax `(1|x)` similar to package lme4 (see lmer() function in package lme4). Interaction terms are not yet implemented in `siland` contrary to `Bsiland`.

The explanatory variables are added using the symbol `"+"`. Interaction terms can be considered using the usual symbols `"*"` or `":"`. Notice that only interactions between local x local and local x landscape variables are considered.

Two families of SIF are currently implemented in the siland package, exponential and Gaussian families defined as $f_\delta(d) = 2/(\pi\delta^2)exp(-2d/\delta)$ and $f\delta(d) = 1/(2\delta\sqrt{\pi})exp(-d\pi/(2\delta))^2$, respectively. By definition, δ represents the mean distance of these SIF.

The computation of the spatial influence of landscape variables is based on an approximation through the description of landscape variables distribution along a regular grid (modelisation of spatial influence of each pixel of landscape variable). The choice of wd, the size of pixels can have an impact on the estimation. The smaller the pixels, the more accurate the approximation, but also the longer the computation time. In our experience, we recommend using a pixel size at most three times smaller than the smallest estimated mean distance among the landscape variables SIF.

Value

siland returns an object of type siland, a list :

coefficients	a vector of the estimated coefficients
paramSIF	a vector of the estimated parameters of SIF
formula	an object of class formula that indicates the model
landcontri	a dataframe of estimated cumulative influence of each landscape variable (in columns) on each observation (in rows).
loglik	the estimated maximum loglikelihood
loglik0	the estimated maximum loglikelihood of the local model (no landscape variable)
result	a lm/glm/lmer object that corresponds to the model estimate knowing the estimated SIF parameters of landscape variables.
fitted	the fitted values
sif	the family of the SIF
resoptim	an object of class optim or optimize giving informations about the optimization procedure see optim() or optimize() for further details.
AIC	the akaike information criterion of the model
AIC0	the akaike information criterion of the local model (no landscape variable)
nparam	the number of parameters
pval0	the p.value of the test of the landscape effect (the global effect of all spatial variables). Issued from the likelihood ratio test of the (complete) model vs the local model (complete model without landscape variables).
family	the family distribution for the model
sd.error	the estimated standard error for gaussian family, NA in other case
model.Type	the type of local model: GLM for generalised model, LMM for linear mixed model or GLMM for generalised linear mixed model
rand.StdDev	the estimated standard deviation of random effects for LMM or GLMM
err	the estimated residuals
border	a logical indicating the value used for estimation
wd	a number indicating the size of pixels

Author(s)

Carpentier, F. and Martin, O.

References

Carpentier, F., & Martin, O. (2019). SILand: an R package for estimating the spatial influence of landscape. bioRxiv, 692566.

Examples

```
data(dataSiland)
data(landSiland)
res=siland(obs~x1+L1+L2,land=landSiland,data=dataSiland,sif="exponential")
res
res$AIC
```

`siland.lik` *-(Loglikelihood) as a function of SIF parameters.*

Description

Function investigating eventual optimization problems during numerical optimization of function siland by plotting the -(loglikelihhod) against parameters of Spatial Influence Functions (SIF) of each landscape variable. The parameter of a SIF is defined as the mean distance of SIF.

Usage

```
siland.lik(res, land,data, varnames = NULL, seqd = seq(2, 2000, length = 10))
```

Arguments

`res`	an object resulting from siland
`land`	the object of class sf describing the landscape variable, i.e. the `land`" argument in the call to the siland function that computed `res`.
`data`	the dataframe describing response variable and the local variables, i.e. the `data` argument in the call to the siland function that computed `res`.
`varnames`	a vector containing the names (characters) of studied landscape variables. If varnames is `NULL` (by default), all landscape variables of the `res` model are considered.
`seqd`	a vector of SIF parameters (positive values) for which the -loglikelihood is computed.

Details

As with all numerical maximisation procedures, optimization problems may arise. The function `siland.lik` allows to point out possible problems of optimization.

`siland.lik` plots the -Loglikelihood against SIF parameters. During the siland procedure, the estimation of SIF parameters is made by maximazing the likelihood i.e. by minimizing the -Loglikelihood. The orange horizontal line indicates the minimal value of -Loglikelihood obtained during the estimation (`res` result). The other continous curves represent the -loglikehood as a function of SIF parameters for each studied landscape variable (considering the other parameters of the model at their estimated value in `res`). The dotted lines indicate the values of SIF parameter estimated (`res` result).

When the minization correctly occurs, the minimal values of the -loglikelihoods for each landscape variable are equal to the value of the estimated minimal -Loglikelihood. This means that the value of each continuous curves is minimal at its intersection with the orange horizontal line. This also means that dotted lines intersect their associated continuous lines at their minimum.

If it is not the case, the minimizing procedure has failed and it is necessary to proceed again with an estimation with different initialisation values. This is possible with the argument `init` in function `siland`.

Value

a ggplot of the -(loglikelihhod) against SIF parameters of each landscape variables.

Author(s)

O. Martin and F. Carpentier

Examples

```
data(dataSiland)
data(landSiland)
res=siland(obs~x1+L1+L2,data=dataSiland,land=landSiland)
siland.lik(res,dataSiland,land=landSiland,varnames=c("L1","L2"),seqd=seq(5,500,length=20))

# Illustration of failed estimation caused by wrong initial
# values (chosen initial values that are deliberately and obviously far too high)
# In a such case, the estimation should be done again ,
# by  calling siland with init=c(250,250) for instance.

res2=siland(obs~x1+L1+L2,data=dataSiland,land=landSiland,init=c(20000,20000))
siland.lik(res2,dataSiland,land=landSiland,varnames=c("L1","L2"),seqd=seq(1e9,1e10,length=20))
```

`siland.quantile` *Quantile of spatial influence function (SIF)*

Description

function computing quantiles of SIF, i.e. the radius of disk on which the integrated influence of the SIF is equal to given proportions, `p`. Calculations are performed for the estimated SIF of each landscape variable in `x`.The **area of medium influence and significant influence of a landscape variable**, that we defined as the disc containing 50% and 95% of the influence of the SIF (neglecting 50% and 5% of its broader effect) respectively can be computed using argument `p = 0.5` and `p = 0.95`, respectively.

Usage

```
siland.quantile(x, p = 0.95)
```

Arguments

`x`	an object of class siland resulting from function `siland`
`p`	a vector of probabilities

Value

A matrix of the radii of disks on wich the integrated influences of the SIF estimated for each landscape variable of `x` (in columns) are equal to each probability of the vector `p` (in rows).

Author(s)

O. Martin and F. Carpentier

Examples

```
data(dataSiland)
data(landSiland)
res=siland(obs~x1+L1+L2,land=landSiland,data=dataSiland,sif="exponential")
#siland.quantile(x=res,p=c(0.5,0.95))
```

summary.siland *The function gives a summary of an object of class siland*

Description

The function gives a detailed summary of an object of class siland. It gives results as the summary for an object of type glm or lm.

Usage

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

Arguments

object	an object of type siland
...	Dots are not take into account in this version

Index