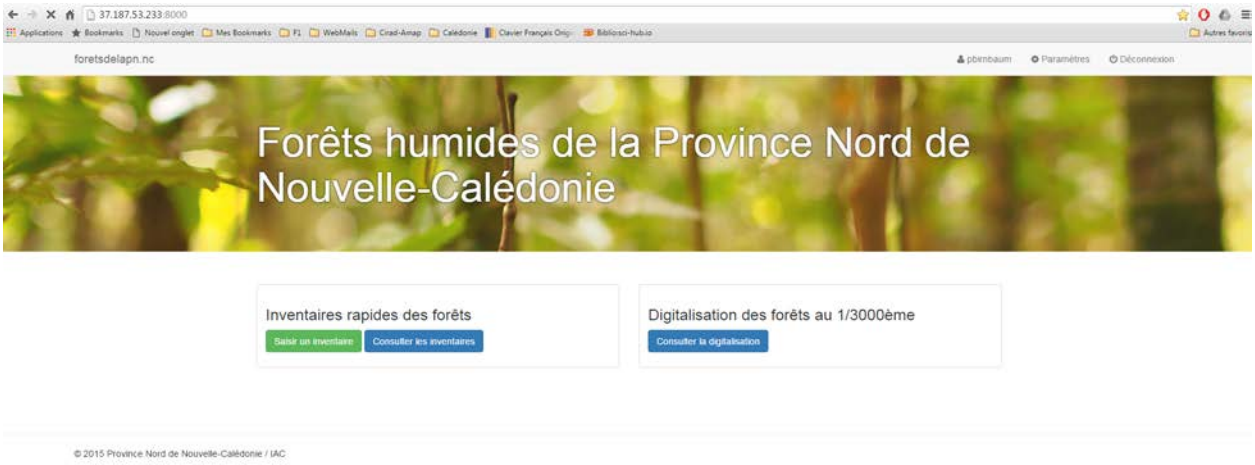


Spatialisation de la richesse floristique et développement d'un portail permettant la saisie et la consultation des formulaires d'évaluation rapide des forêts

Rapport final
(2015-2016)

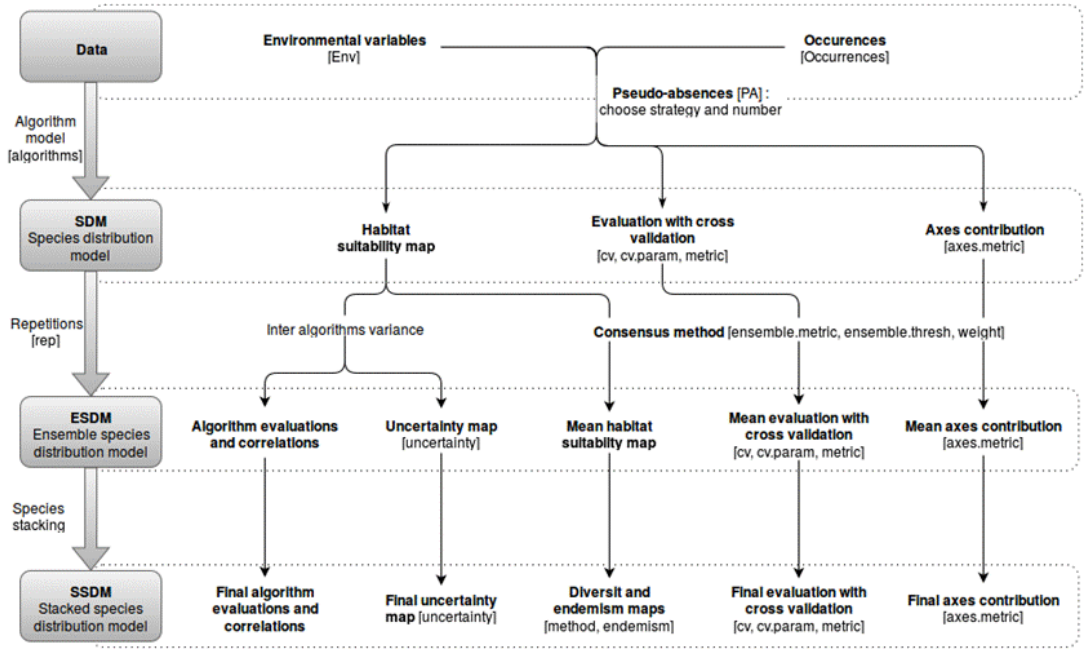
Philippe Birnbaum (IAC/CIRAD - UMR AMAP 51)

Rapport relatif à la convention n°15C534(181) entre la Province Nord et l'IAC

Rappel : cette convention engageait l'IAC à mettre en place : (1) un outil permettant la spatialisation de la richesse des espèces d'arbres par l'utilisation et l'évaluation de différents modèles prédictifs ; et (2) un outil permettant la consultation et la saisie en ligne des formulaires d'évaluation rapide de l'état des forêts.

1/ Le *package* 'SSDM'

Le premier point a été abordé par le développement d'un *package* sous le language de programmation et l'environnement statistique R qui a été baptisé 'SSDM'. Bien que la cartographie de biodiversité basée sur les SSDM (« *Stacked Species Distribution Models* ») fasse l'objet d'un intérêt croissant de la part des biologistes de la conservation, il n'existait jusqu'alors aucune interface conçue spécifiquement pour fournir les outils de base nécessaires à la construction de tels modèles. Ce *package* offre toute une gamme de méthodes et de possibilités de paramétrage à chaque étape du *workflow* du modèle : sélection de pseudo-absences, évaluation des modèles et de la contribution des variables environnementales, construction de modèles de consensus fusionnant plusieurs algorithmes, assemblage des espèces, cartographie de l'endémisme, etc.



Son architecture orientée objet permet aux utilisateurs de modifier les méthodes existantes ou d'en implanter de nouvelles. Son interface simple d'utilisation et sa licence libre rendent en outre cet outil accessible aux biologistes de la conservation les moins familiers avec la programmation et/ou disposant de ressources financières limitées.

Le *package* et le manuel d'utilisation associé peuvent être téléchargés sur le portail *The Comprehensive R Archive Network* via le lien : https://cran.r-project.org/web/packages/SSDM/index.html

Il fait également l'objet d'un article en cours de préparation qui sera prochainement soumis à la revue internationale à comité de lecture *Ecological Informatics* :

Schmitt, S., Birnbaum, P., Justeau, D., de Boissieu, F., Pouteau, R. (in prep.) SSDM: An R package to map species richness from natural history records.

2/ Le portail 'Inventaires rapides des forêts'

Pour le second point, nous avons construit un portail mis en ligne sur le lien : http://37.187.53.233:8000/rapid_inventories/

Title

SSDM: an R package to map species richness from natural history records

Authors

Sylvain Schmitt[1,2], Philippe Birnbaum[1,3], Dimitri Justeau[1], Florian de Boissieu[4] and Robin Pouteau[1]

Affiliations

[1] Institut Agronomique néo-Calédonien (IAC), Noumea, New Caledonia

[2] AgroParisTech, Nancy, France

[3] Centre de Coopération Internationale en Recherche Agronomique pour le Développement (CIRAD), Montpellier, France

[4] Institut de Recherche pour le Développement (IRD), Noumea, New Caledonia

Author for correspondence

Sylvain Schmitt, Diversité biologique et fonctionnelle des écosystemes terrestres, Laboratoire de botanique et d'écologie végétale appliquées, Institut Agronomique néo-Calédonien (IAC), BP A5, 98 848 Noumea, New Caledonia

E-mail: sylvain.schmitt@agroparistech.fr

Abstract

Although biodiversity mapping based on stacked species distribution models (SSDMs) is gaining growing interest among conservationists, no user-friendly interface specifically designed to provide the basic tools needed to fit such models was available until now. The package 'SSDM' is a computer platform implemented in R providing a range of methodological approaches and parametrization at each step of the model building: e.g., pseudo-absence selection, model and environmental variable evaluation, inter-model consensus forecasting, species assembly and endemism mapping. The object-oriented design of the package is such that conservation scientists can modify existing methods, extend the framework by implementing new methods, and share them to be reproduced by others. The package includes a global user interface to broaden the use of SSDMs to a larger public.

Introduction

Understanding patterns of local species richness (α-diversity) is a critical prerequisite to implement effective conservation strategies. Richness maps can provide the basis for reserve selection (Murray-Smith et al., 2009; Raes et al., 2009; Cañadas et al., 2014; Moraes et al., 2014), prevention of biological invasions (Bellard et al., 2013; Kelly et al., 2014; Gallardo et al., 2015; Pouteau et al., 2015a), and mitigation of future impacts of climate change (Midgley et al., 2003; Siqueira and Peterson, 2003; Fitzpatrick et al., 2008; Colombo and Joly, 2010; Ogawa-Onishi et al., 2010; Bellard et al., 2013; Brown et al., 2015).

As it is not realistic to capture the full variation of species richness over large areas using comprehensive species inventories, a range of more pragmatic methods to extrapolate scattered local observations have been developed. They include:

(1) point-to-grid maps, which assemble natural history records (e.g., herbarium or museum specimens) within grid cells and count the number of species (Droissart et al., 2012; Wulff et al., 2013; Cañadas et al., 2014; Tovaranonte et al., 2015). Unfortunately, natural history records are seldom evenly sampled so as the accuracy of this method tends to decrease as cell resolution increases so it reaches its maximum reliability at a scale too coarse for local decision-makers (Graham and Hijmans, 2006);

(2) macro-ecological models, which relate species richness observed over a network of inventories (e.g., plots, transects, quadrats) with spatially explicit environmental variables (Bhattarai and Vetaas, 2003; Sánchez-González and López-Mata, 2005; Tomasetto et al., 2013). This method however has the disadvantage of needing a large number of inventories to be accurately calibrated and appears unable to extrapolate beyond known communities (Ferrier and Guisan, 2006);

(3) stacked species distribution models (SSDMs), which combine multiple individual species distribution models (SDMs) to produce a community-level model (Ferrier and Guisan, 2006).

An SDM (also referred as to ‘niche model’ or ‘habitat suitability model’) refers to the process of using a computer algorithm to predict the distribution of a species in geographical space on the basis of a mathematical representation of its known distribution in environmental space (Guisan and Thuiller, 2005). With the increasing availability of distributional data in biodiversity databases, SDMs have gained much attention for a wide variety of conservation applications like managing biological invasions, identifying and protecting critical habitats, selecting nature reserve, and translocating rare and endangered species (Guisan et al., 2013). Diversity mapping based on multiple SDMs promises to have great potential for conservationists and the method is attracting growing interest with regard to the literature (Midgley et al., 2003; Siqueira and Peterson, 2003; Fitzpatrick et al., 2008; Murray-Smith et al., 2009; Raes et al., 2009; Colombo and Joly, 2010; Ogawa-Onishi et al., 2010; Pérez and Font, 2012; Schmidt-Lebuhn et al., 2012; Mateo et al., 2013; Moraes et al., 2014; Brown et al., 2015; Pouteau et al., 2015b). However, the main limitation to the use of SSDM is that the method requires computationally complex routines that only conservationists with advanced computer skills can implement. Indeed, no user-friendly interface specifically designed to provide the basic tools needed to build an SSDM was available until now (Table 1).

The package ‘`SSDM`’ is a free and open source object-oriented platform for stacked species distribution modelling implemented in R, perhaps the most commonly used software for ecological analysis in which state-of-the-art methods can easily be incorporated. It provides a standardized and unified structure for visualizing and handling species distributions data and models. The package proposes a range of cutting-edge methods including nine model algorithms and allows building ensembles of forecasts to account for inter-model variability. The easy-to-use graphical user interface is likely to broaden the use of SSDMs to a large number of conservation scientists.

Model flow

The workflow of the package ‘`SSDM`’ is based on three levels: (1) an individual SDM is fitted

by linking occurrences of a single species to environmental predictor variables based on the response curve of a single computer algorithm; (2) for each species, an ensemble SDM (ESDM) can be created from several algorithm outputs to create a model that captures components of each; and (3) species assembly is predicted by stacking several SDM or ESDM outputs (Fig. 1).

Data inputs

Natural history records

Most model algorithms included in the package 'SSDM' (introduced below) require presence/absence occurrence datasets. When a sampling scheme did not accounted for species absences, the package can select pseudo-absences (randomly selected sites where a species is assumed to be absent). Three modalities can be chosen to select pseudo-absences: (1) the selection strategy: either within the extent of the environmental rasters or within a disk of a user-specified radius around each presence (Barbet-Massin et al., 2012); (2) the number of selected pseudo-absences: either a user-specified number or a number equal to the number of presences available for each species; and (3) the number of times the selection is repeated: repetition reduces potential errors due to randomization in pseudo-absence selection. When pseudo-absences are selected repeatedly, the package will merge results of all runs by averaging habitat suitability probabilities and the associated accuracy metrics. Default parameters have been set to recommendations from Barbet-Massin et al. (2012) adapted to each model algorithm. In order to deal with natural history records derived from opportunistic sampling schemes prone to spatial autocorrelation, the R package for spatial thinning of species occurrences 'spThin' has been integrated (Aiello-Lammens et al., 2015).

Environmental variables

Nine image formats can be uploaded into the package 'SSDM' to describe the environment

species occupy, which facilitates data management and exchange with conventional GIS packages. The package supports both continuous (e.g., climate maps, digital elevation models, bathymetric maps) and categorical environmental variables (e.g., land cover maps, soil type maps) as inputs. The package also allows normalizing environmental variables, which may be useful to improve the fit of certain algorithms (like artificial neural networks).

Rasters of environmental data need to have the same projection while spatial extent and resolution of the environmental layers do not need to be the same. During processing, the package will deal with between-variables discrepancies in spatial extent and resolution by rescaling all environmental rasters to the smallest common spatial extent then upscaling them to the coarsest resolution.

Model algorithms

Individual species distribution models (SDMs)

The package '`SSDM`' includes a comprehensive set of algorithms to model species distributions including four regression algorithms: general additive models (GAM), generalized linear models (GLM), multivariate adaptive regression splines (MARS) and maximum entropy (Maxent); two classification algorithms: classification tree analysis (CTA) and generalized boosted models (GBM); and three machine learning algorithms: artificial neural networks (ANN), random forests (RF), and support vector machines (SVM). Default parameters of the original R package of each algorithm were conserved but most of them remain settable (Table 2).

A major assumption behind the concept of SDM is that species would be in equilibrium with their environment so as species dispersal limitation is ignored by the most classical SDM implementations (Guisan and Thuiller, 2005). Hence, a SDM may overestimate the geographical area that a species occupy if its distribution is shaped by dispersal barriers. In order to account for this potential over-prediction, the package contains an option to perform

a user-specified range restriction on habitat suitability maps around each presence (Crisp et al., 2001).

For each species, the package can store two results in raster format: (1) a continuous raster giving the habitat suitability index for presence-only data, and the probability of presence (ranging from 0 to 1) for presence/absence data; and (2) a binary presence/absence raster based on the threshold specified by the user.

Ensemble species distribution models (ESDMs)

Two consensus methods are implemented in the package 'SSDM': (1) a simple averaging of model outputs; and (2) a weighted averaging based on a user-specified metric or group of metrics (presented below) (Marmion et al., 2009). The package also provides an uncertainty map representing the between-algorithms variance. The between-algorithms pairwise degree of agreement can be assessed through a correlation matrix giving the Pearson's coefficients of habitat suitability maps yield by each algorithm.

Stacked species distribution models (SSDMs)

The final map of local species richness can be built by summing either habitat suitability maps or threshold-based presence/absence maps. In the latter case, thresholding can be based either on one of the available accuracy metrics or by drawing repeatedly from a Bernouilli distribution (Calabrese et al., 2014; Dubuis et al., 2011). Resulting maps can be exported then imported into other GIS software packages for further data analysis and visualization.

Additional outputs

Model accuracy assessment

A range of metrics to evaluate models have been implemented in the package 'SSDM'. They include the area under the receiving operating characteristic (ROC) curve (AUC), the

Cohen's Kappa coefficient, the omission rate, the sensitivity (true positive rate) and the specificity (true negative rate) (Fielding and Bell, 1997). These metrics are all based on the confusion matrix (also called 'error matrix', it represents the instances in a predicted class *versus* the instances in an actual class) and, consequently, need an *a priori* conversion of habitat suitability probabilities into binary maps. The optimal threshold to split presences and absences on the basis of habitat suitability probabilities can be set to the probability that maximizes: the Cohen's Kappa coefficient, the correct classification rate (CCR), the true skill statistic (TSS), the sensitivity/specificity equality (SES), the lowest prediction occurrence probability or the shortest distance between the ROC curve and the upper left corner of the ROC plot. Recommendations from Liu et al. (2005, 2013) for thresholding were set to default in the package. To ensure independence between training and evaluation sets, three methods to split the initial dataset are available: (1) 'holdout', in which the initial dataset is partitioned using a user-defined fraction, (2) '*k*-folds', in which the initial dataset is partitioned into *k* folds being *k*-1 times the training set and once the evaluation set, and (3) 'leave-one-out', in which each point is successively used for evaluation.

Importance analysis of environmental variables

The package 'SSDM' can provide two measures of the relative contribution of environmental variables, which quantifies the relevance of an environmental variable to determine the species distribution. The first one is based on a jackknife approach that evaluates the change in accuracy between a full model and one with each environmental variable omitted in turns (Phillips et al., 2006). All metrics available in the package can serve to assess the change in accuracy. The second method is based on the Pearson's correlation coefficient between a full model and one with each environmental variable omitted in turns (Thuiller et al., 2009).

Endemism mapping

In addition to species richness, endemism is an important feature for conservation decision-making (Crisp et al., 2001; Raes et al., 2009; Moraes et al., 2014). The package 'SSDM' offers

the opportunity to map local species endemism using two metrics: (1) the weighted endemism index (WEI); and (2) the corrected weighted endemism index (CWEI) (Crisp et al., 2001). WEI seeks to avoid the problem that an arbitrary region or range-size threshold is used to define what constitutes an endemic species. WEI avoids using a threshold for endemism by applying a simple continuous weighting function, assigning high weights to species with small ranges, and progressively smaller weights to species with larger ranges. WEI is calculated by counting all species in a given area, then weighting by the inverse of its range size. CWEI is an alternative measure to reduce the correlation between richness and endemism. CWEI is calculated as the weighted endemism score for each cell divided by the richness score and represents the average degree of endemism of the species recorded in an area.

Examples

Vulnerability to invasive species at global scale

Occurrences for 100 of the world's worst invasive alien species (as defined by the Invasive Species Specialist Group of the International Union for Conservation of Nature; http://www.issg.org/) were gathered from the Global Biodiversity Information Facility (GBIF) (http://www.gbif.org/). Occurrences containing invalid coordinates and country or taxon issues were removed. The set of 19 WorldClim climate variables (all continuous) at a 2.5 arcmin x 2.5 arcmin resolution were used as environmental variables (Hijmans et al., 2005). Variable multicollinearity was addressed by examining cross-correlations. For variables with correlations of $r^2 > 0.8$, only the variable that decreased model accuracy the most when omitted from the full model was retained. Then, an SSDM with all model settings set to default was fitted. The output provides a picture of how richness in 100 of the world's worst invasive alien species might be distributed without any constraints of spread or competitive interactions (Fig. 2).

Endemism of the genus Psychotria *in New Caledonia*

Psychotria (Rubiaceae) is the second most speciose genus on the megadiverse archipelago of New Caledonia (Barrabé et al., 2014). Occurrences of all described species belonging to this genus were extracted from databases of the Noumea (NOU) and Paris herbaria (P), respectively VIROT and SONNERAT. Six environmental variables (five continuous and one categorical) at a 100 m x 100 m resolution were used to fit an SSDM: elevation, potential insolation, slope steepness, substrate type, windwarness, and a topographical wetness index (Pouteau et al., 2015b). Continuous variable were correlated with an $r^2 < 0.80$. A WEI map was built with all model settings set to default. The output provides a picture of how the level of endemism of this focal genus is spatially organised in New Caledonia (Fig. 3).

Installation

The package '`SSDM`' is free and open source (GPL v3 licence). It is available from the CRAN repository < https://cran.r-project.org/web/packages/SSDM/index.html >, and can be installed either from CRAN or within the R environment using the command *install.packages('SSDM')*. The project is hosted on Github (url: < https://github.com/sylvainschmitt/SSDM >), which allows future users to openly contribute to the project.

Acknowledgement

We are grateful to Thomas Ibanez (IAC) and François Muñoz (University of Montpellier, France) for their useful comments on an earlier draft, and to Laure Barrabé (IAC) and Frédéric Rigault (IRD) for gathering and pre-processing occurrences of *Psychotria* used in the second example. We also would like to thank the package '`BIOMOD2`' for inspiration. The implementation of the package '`SSDM`' has been funded by the Direction for Economic and Environmental Development (DDEE) of the North Province of New Caledonia.

References

1. Aiello-Lammens, M.E., Boria, R.A., Radosavljevic, A., Vilela, B., Anderson, R.P., 2015. “spThin: an R package for spatial thinning of species occurrence records for use in ecological niche models. Ecography 38, 541–545.

2. Barbet-Massin, M., Jiguet, F., Albert, C.H., Thuiller, W., 2012. Selecting pseudo-absences for species distribution models: how, where and how many? Methods in Ecology and Evolution 3, 327–338.

3. Barrabé, L., Maggia, L., Pillon, Y., Rigault, F., Mouly, A., Davis, A.P., Buerki, S., 2014. New Caledonian lineages *Psychotria* (Rubiaceae) reveal different evolutionary histories and the largest documented plant radiation for the archipelago. Molecular Phylogenetics and Evolution 71, 15–35.

4. Bhattarai, K.R., Vetaas, O.R., 2003. Variation in plant species richness of different life forms along a subtropical elevation gradient in the Himalayas, east Nepal. Global Ecology and Biogeography 12, 327–340.

5. Bellard, C., Thuiller, W., Leroy, B., Genovesi, P., Bakkenes, M., Courchamp, F., 2013. Will climate change promote future invasions? Global Change Biology 19, 3740–3748.

6. Brown, K.A., Parks, K.E., Bethell, C.A., Johnson, S.E., Mulligan, M. (2015) Predicting plant diversity patterns in Madagascar: understanding the effects of climate and land cover in a biodiversity hotspot. PLoS ONE 10, e0122721.

7. Cañadas, E.M., Fenu, G., Peñas, J., Lorite, J., Mattana, E. & Bacchetta, G., 2014. Hotspots within hotspots: endemic plant richness, environmental drivers, and implications for conservation. Biological Conservation 170, 282–291.

8. Calabrese, J.M., Certain, G., Kraan, C., Dormann, C.F., 2014. Stacking species

distribution models and adjusting bias by linking them to macroecological models: stacking species distribution models. Global Ecology and Biogeography 23, 99–112.

9. Colombo, A.F., Joly, C.A., 2010. Brazilian Atlantic Forest lato sensu: the most ancient Brazilian forest, and a biodiversity hotspot, is highly threatened by climate change. Brazilian Journal of Biology 70, 697–708.

10. Crisp, M.D., Laffan, S., Linder, H.P., Monro, A., 2001. Endemism in the Australian flora. Journal of Biogeography 28, 183–98.

11. Diniz-Filho, J.A.F., Bini, L.M., Rangel, T.F., Loyola, R.D., Hof, C., Nogués-Bravo, D., Araújo, M.B., 2009. Partitioning and mapping uncertainties in ensembles of forecasts of species turnover under climate change. Ecography 32, 897–906.

12. Droissart, V., Hardy, O.J., Sonké, B., Dahdouh-Guebas, F., Stévart, T., 2012. Subsampling herbarium collections to assess geographic diversity gradients: a case study with endemic Orchidaceae and Rubiaceae in Cameroon. Biotropica 44, 44–52.

13. Dubuis, A., Pottier, J., Rion, V., Pellissier, L., Theurillat, J.-P., Guisan, A., 2011. Predicting spatial patterns of plant species richness: a comparison of direct macroecological and species stacking modelling approaches. Diversity and Distributions 17, 1122–1131.

14. Ferrier, S., Guisan, A., 2006. Spatial modelling of biodiversity at the community level. Journal of Applied Ecology 43, 393–404.

15. Fielding, A.H., Bell, J.F., 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. Environmental Conservation 24, 38–49.

16. Fitzpatrick, M.C., Gove, A.D., Sanders, N.J., Dunn, R.R., 2008. Climate change, plant migration, and range collapse in a global biodiversity hotspot: the Banksia

(Proteaceae) of Western Australia. Global Change Biology 14, 1337–1352.

17. Gallardo, B., Zieritz, A., Aldridge, D.C., 2015. The importance of the human footprint in shaping the global distribution of terrestrial, freshwater and marine invaders. PLoS ONE 10, e0125801.

18. Graham, C.H., Hijmans, R.J., 2006. A comparison of methods for mapping species ranges and species richness. Global Ecology and Biogeography 15, 578–587.

19. Guisan, A., Thuiller, W., 2005. Predicting species distribution: offering more than simple habitat models. Ecology Letters 8, 993–1009.

20. Guisan, A. et al., 2013. Predicting species distributions for conservation decisions. Ecology Letters 16, 1424–1435.

21. Guo, Q., Liu, Y., 2010. ModEco: an integrated software package for ecological niche modeling. Ecography 33, 637–642.

22. Kelly, R., Leach, K., Cameron, A., Maggs, C.A., Reid, N., 2014. Combining global climate and regional landscape models to improve prediction of invasion risk. Diversity and Distributions 20, 884–894.

23. Liu, C., Berry, P.M., Dawson, T.P., Pearson, R.G., 2005. Selecting thresholds of occurrence in the prediction of species distributions. Ecography 28, 385–393.

24. Liu, C., White, M., Newell, G., 2013. Selecting thresholds for the prediction of species occurrence with presence-only data. Journal of Biogeography 40, 778–789.

25. Marmion, M., Parviainen, M., Luoto, M., Heikkinen, R.K., Thuiller, W., 2009. Evaluation of consensus methods in predictive species distribution modelling. Diversity and Distributions 15, 59–69.

26. Midgley, G.F., Hannah, L., Millar, D., Thuiller, W., Booth, A., 2003. Developing regional

and species-level assessments of climate change impacts on biodiversity in the Cape Floristic Region. Biological Conservation 112, 87–97.

27. Moraes, M.M., Ríos-Uzeda, B., Moreno, L.R., Huanca-Huarachi, G., Larrea-Alcazar, D., 2014. Using potential distribution models for patterns of species richness, endemism, and phytogeography of palm species in Bolivia. Tropical Conservation Science 7, 45–60.

28. Murray-Smith, C., Brummitt, N.A., Oliviera-Filho, A.T., Bachman, S., Moat, J., Lughadha, E.M.N., Lucas, E.J., 2009. Plant diversity hotspots in the Atlantic coastal forests of Brazil. Conservation Biology 23, 151–163.

29. Naimi, B., Araújo, M.B., 2016. sdm: a reproducible and extensible R platform for species distribution modelling 39, 368–375.

30. Ogawa-Onishi, Y., Berry, P.M., Tanaka, N., 2010. Assessing the potential impacts of climate change and their conservation implications in Japan: a case study of conifers. Biological Conservation 143, 1728–1736.

31. Phillips, S.J., Anderson, R.P., Schapire, R.E., 2006. Maximum entropy modeling of species geographic distributions. Ecological Modelling 190, 231–259.

32. Pouteau, R., Hulme, P.E., Duncan, R.P., 2015a. Widespread native and alien plant species occupy different habitats. Ecography 68, 462–471.

33. Pouteau, R., Bayle, E., Blanchard, E., Birnbaum, P., Cassan, J.-J., Hequet, V., Ibanez, T., Vandrot, H., 2015b. Accounting for the indirect area effect in stacked species distribution models to map species richness in a montane biodiversity hotspot. Diversity and Distributions 21, 1329–1338.

34. Raes, N., Roos, M.C., Slik, J.W.F., Van Loon, E.E., ter Steege, H., 2009. Botanical richness and endemicity patterns of Borneo derived from species distribution models.

Ecography 32, 180–192.

35. Sánchez-González, A., López-Mata, L., 2005. Plant species richness and diversity along an altitudinal gradient in the Sierra Nevada, Mexico. Diversity and Distributions 11, 567–575.

36. Siqueira, M.F., Peterson, A.T., 2003. Consequences of global change for geographic distributions of cerrado tree species. Biota Neotropica 3.

37. de Souza Muñoz, M.E., De Giovanni, R., de Siquiera, M.F., Sutton, T., Brewer, P., Pereira, R.S., Canhos, D.A.L., Canhos, V.P., 2009. openModeller: a generic approach to species' potential distribution modelling. GeoInformatica 15, 111–135.

38. Thuiller, W., Lafourcade, B., Engler, R., Araújo, M.B., 2009. BIOMOD – a platform for ensemble forecasting of species distributions. Ecography 32, 369–373.

39. Tomasetto, F., Duncan, R.P., Hulme, P.E., 2013 Environmental gradients shift the direction of the relationship between native and alien plant species richness. Diversity and Distributions 19, 49–59.

40. Tovaranonte, J., Blach-Overgaard, A., Pongsattayapipat, R., Svenning, J.-C., Barfod, A.S., 2015. Distribution and diversity of palms in a tropical biodiversity hotspot (Thailand) assessed by species distribution modeling. Nordic Journal of Botany 33, 214–224.

41. Wulff, A.S., Hollingsworth, P.M., Ahrends, A., Jaffré, T., Veillon, J.-M., L'Huillier, L., Fogliani, B., 2013. Conservation priorities in a biodiversity hotspot: analysis of narrow endemic plant species in New Caledonia. PLoSONE 8, e73371.

Table caption

Table 1. A non-exhaustive list of software packages designed to perform species distribution modelling with their main advantages and limits in relation to species richness mapping.

Table 2. A list of implemented model algorithms in the first release of the package ‘SSDM’ and their dependent packages

Table 1.

Software	Graphical user interface	Developed in R	Designed to fit SSDMs	Available as of the time of this writing	Reference
BIOENSEMBLES	X			X	Diniz-Filho et al. (2009)
BIOMOD2		X		X	Thuiller et al. (2009)
ModEco	X			X	Guo and Liu (2010)
Openmodeller	X			X	de Souza Muñoz et al. (2009)
sdm	Not for all functions	X	X		Naimi and Araújo (2016)
SSDM	X	X	X	X	This article

Table 2.

Type of algorithm	Model algorithm	Dependent package
Regression	GAM	mgcv
	GLM	stats
	MARS	earth
	MAXENT	dismo
Classification	CTA	rpart
	GBM	gbm
Machine learning	ANN	nnet
	RF	randomForest
	SVM	e1071

Figure captions

Figure 1. Flow chart of the package ‘SSDM’

Figure 2. World map of vulnerability to 100 of the world’s worst invasive species generated with the package ‘SSDM’

Figure 3. Weighted endemism map of the genus *Psychotria* in New Caledonia generated with the package ‘SSDM’

Figure 1.

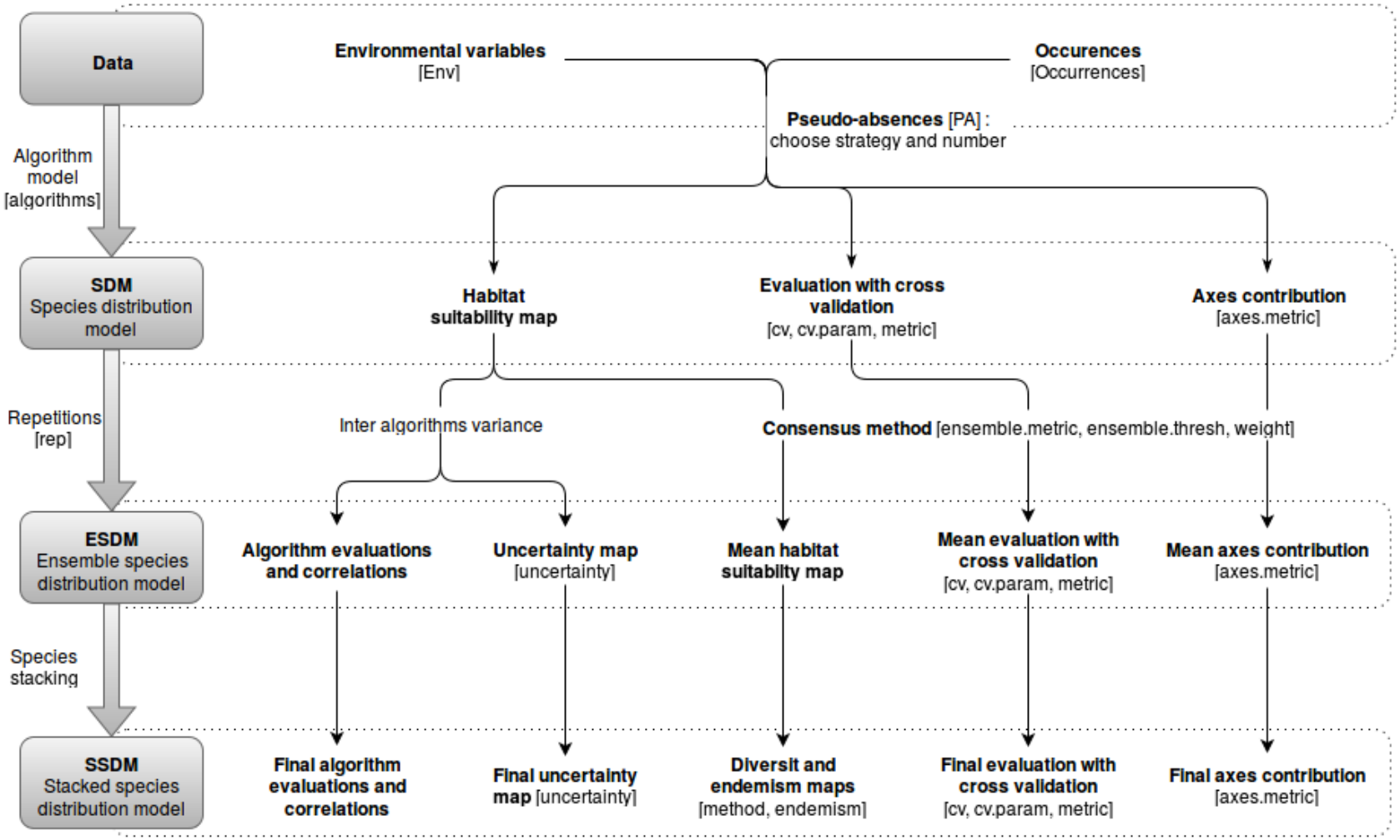

Data
Environmental variables [Env]
Occurences [Occurrences]
Pseudo-absences [PA] : choose strategy and number
Algorithm model [algorithms]
SDM Species distribution model
Habitat suitability map
Evaluation with cross validation [cv, cv.param, metric]
Axes contribution [axes.metric]
Repetitions [rep]
Inter algorithms variance
Consensus method [ensemble.metric, ensemble.thresh, weight]
ESDM Ensemble species distribution model
Algorithm evaluations and correlations
Uncertainty map [uncertainty]
Mean habitat suitability map
Mean evaluation with cross validation [cv, cv.param, metric]
Mean axes contribution [axes.metric]
Species stacking
SSDM Stacked species distribution model
Final algorithm evaluations and correlations
Final uncertainty map [uncertainty]
Diversit and endemism maps [method, endemism]
Final evaluation with cross validation [cv, cv.param, metric]
Final axes contribution [axes.metric]

Figure 2.

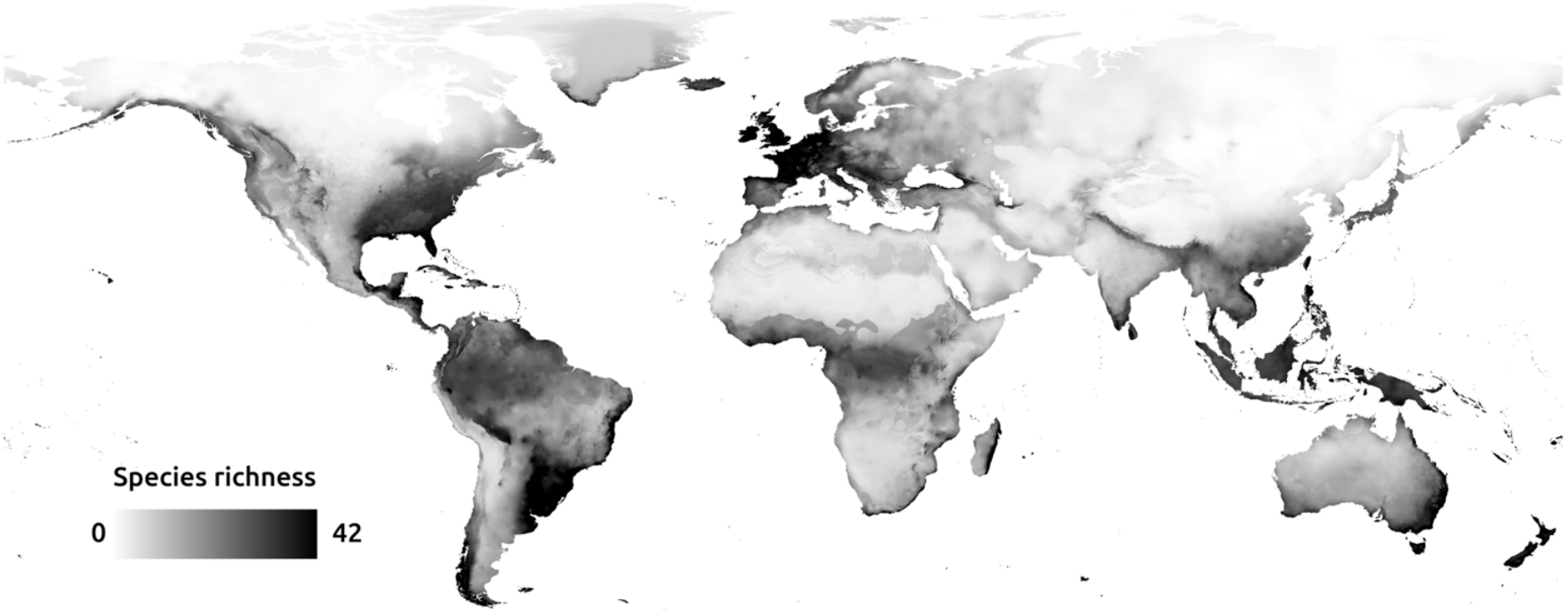
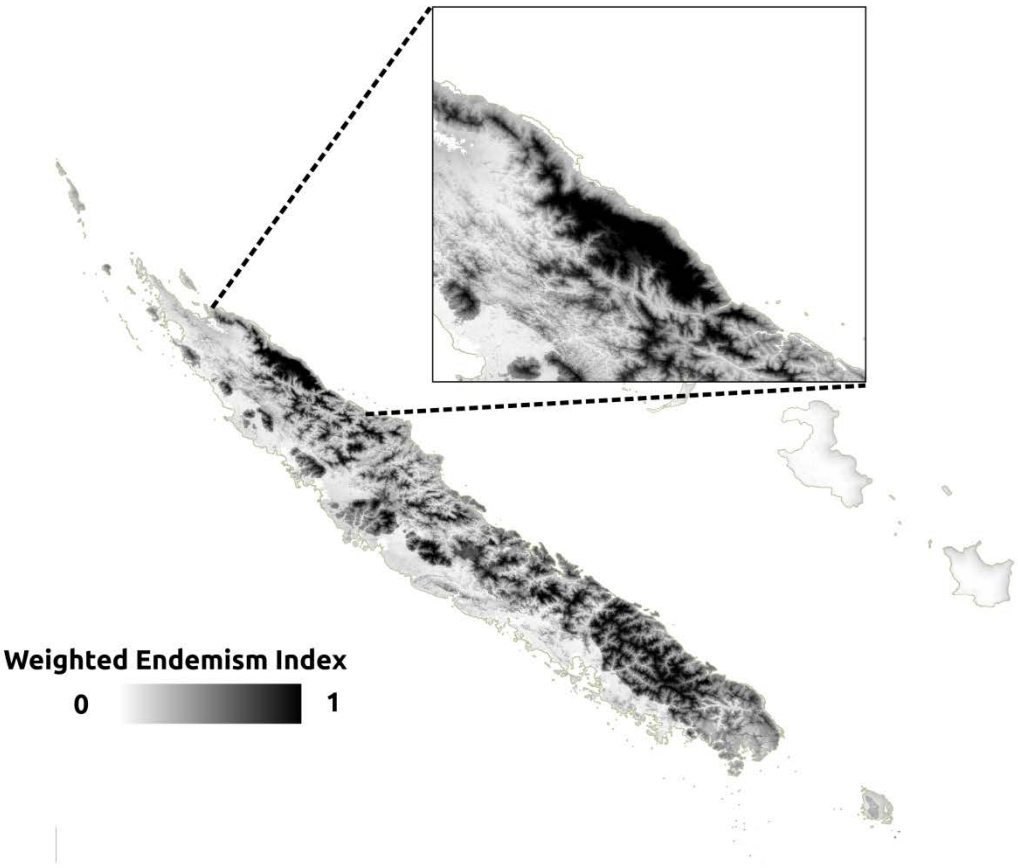


Figure 3.



Package 'SSDM'

February 17, 2016

Type Package

Title Stacked Species Distribution Modelling

Version 0.1.1

Date 2016-02-17

Author Sylvain Schmitt, Robin Pouteau, Dimitri Justeau, Philippe Birnbaum

Maintainer Sylvain Schmitt <sylvain.schmitt@agroparistech.fr>

Description Allows to map species richness and endemism based on stacked species distribution models (SSDM). Individuals SDMs can be created using a single or multiple algorithms (ensemble SDMs). For each species, an SDM can yield a habitat suitability map, a binary map, a between-algorithm variance map, and can assess variable importance, algorithm accuracy, and between-algorithm correlation. Methods to stack individual SDMs include summing individual probabilities and thresholding then summing. Thresholding can be based on a specific evaluation metric or by drawing repeatedly from a Bernoulli distribution. The SSDM package also provides a user-friendly interface.

License GPL (>= 3) | file LICENSE

LazyData TRUE

Imports sp (>= 1.2.0), raster (>= 2.4.20), methods (>= 3.2.2), SDMTools (>= 1.1.221), mgcv (>= 1.8.7), earth (>= 4.4.3), rpart (>= 4.1.10), gbm (>= 2.1.1), randomForest (>= 4.6.10), dismo (>= 1.0.12), nnet (>= 7.3.10), e1071 (>= 1.6.7), shiny (>= 0.12.2), shinydashboard (>= 0.5.1), gplots (>= 0.1.0), spThin (>= 0.1.0)

Suggests shinyFiles (>= 0.6.0)

Depends R (>= 3.2.2)

Collate 'Algorithm.SDM.R' 'Ensemble.SDM.R' 'Env.R' 'Occurrences.R' 'SSDM.R' 'Stacked.SDM.R' 'checkargs.R' 'ensemble.R' 'modelling.R' 'ensemble_modelling.R' 'gui.R' 'load_model.R' 'load_occ.R' 'load_var.R' 'plot.model.R' 'save.model.R' 'stack_modelling.R' 'stacking.R' 'update.stack.R' 'zzz.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-17 09:00:40

R topics documented:

`Algorithm.SDM-class` *An S4 class to represent an SDM based on a single algorithm*

Description

This is an S4 class to represent an SDM based on a single algorithm (including generalized linear model, general additive model, multivariate adpative splines, generalized boosted regression model, classification tree analysis, random forest, maximum entropy, artificial neural network, and support vector machines). This S4 class is obtained with `modelling`.

Slots

`name` character. Name of the SDM (by default Species.SDM).

`projection` raster. Habitat suitability map produced by the SDM.

`evaluation` data frame. Evaluation of the SDM (available metrics include AUC, Kappa, sensitivity, specificity and proportion of correctly predicted occurrences) and identification of the optimal threshold to convert the habitat suitability map into a binary presence/absence map.

`variable.importance` data frame. Relative importance of each variable in the SDM.

`data` data frame. Data used to build the SDM.

`parameters` data frame. Parameters used to build the SDM.

See Also

Ensemble.SDM an S4 class for ensemble SDMs, and Stacked.SDM an S4 class for SSDMs.

ensemble *Methods to assemble multiple algorithms in an ensemble SDM*

Description

This is a method to assemble several algorithms in an ensemble SDM. The function takes as inputs several S4 Algorithm.SDM class objects obtained with the `modelling` function. The function returns an S4 Ensemble.SDM class object containing the habitat suitability map, the binary map, and the uncertainty map (based on the between-algorithm variance) and the associated evaluation tables (model evaluation, algorithm evaluation, algorithm correlation matrix and variable importance).

Usage

```
ensemble(x, ..., name = NULL, ensemble.metric = c("AUC"),
  ensemble.thresh = c(0.75), weight = T, thresh = 1001, uncertainty = T,
  verbose = T, GUI = F)

## S4 method for signature 'Algorithm.SDM'
ensemble(x, ..., name = NULL,
  ensemble.metric = c("AUC"), ensemble.thresh = c(0.75), weight = T,
  thresh = 1001, uncertainty = T, verbose = T, GUI = F)

## S4 method for signature 'Algorithm.SDM'
sum(x, ..., name = NULL,
  ensemble.metric = c("AUC"), ensemble.thresh = c(0.75), weight = T,
  thresh = 1001, format = T, verbose = T, na.rm = F)
```

Arguments

`x,...`	SDMs. SDMs to be assembled.
`name`	character. Optional name given to the final Ensemble.SDM produced (by default 'Ensemble.SDM').
`ensemble.metric`	character. Metric(s) used to select the best SDMs that will be included in the ensemble SDM (see details below).
`ensemble.thresh`	numeric. Threshold(s) associated with the metric(s) used to compute the selection.
`weight`	logical. Choose whether or not you want the SDMs to be weighted using the selection metric or, alternatively, the mean of the selection metrics.
`thresh`	numeric. A single integer value representing the number of equal interval threshold values between 0 and 1 (see `optim.thresh`).
`uncertainty`	logical. If set to true, generates an uncertainty map and an algorithm correlation matrix.
`verbose`	logical. If set to true, allows the function to print text in the console.

`GUI,format,na.rm`
: logical. Don't take those arguments into account (parameters for the user interface and sum function).

Details

ensemble.metric (metric(s) used to select the best SDMs that will be included in the ensemble SDM) can be chosen from among:

AUC Area under the receiving operating characteristic (ROC) curve

Kappa Kappa from the confusion matrix

sensitivity Sensitivity from the confusion matrix

specificity Specificity from the confusion matrix

prop.correct Proportion of correctly predicted occurrences from the confusion matrix

Value

an S4 Ensemble.SDM class object viewable with the `plot.model` function.

See Also

`ensemble_modelling` to build an ensemble SDM from multiple algorithms.

Examples

```
## Not run:
# Loading data
data(Env)
data(Occurrences)
Occurrences = subset(Occurrences, Occurrences$SPECIES == 'elliptica')

# ensemble SDM building
CTA = modelling('CTA', Occurrences, Env, Xcol = 'LONGITUDE', Ycol = 'LATITUDE')
SVM = modelling('SVM', Occurrences, Env, Xcol = 'LONGITUDE', Ycol = 'LATITUDE')
ESDM = ensemble(CTA, SVM, ensemble.thresh = c(0.6))

# Results plotting
plot(ESDM)

## End(Not run)
```

Ensemble.SDM-class *An S4 class to represent an ensemble SDM*

Description

This is an S4 class to represent an ensemble SDM from multiple algorithms (including generalized linear model, general additive model, multivariate adaptive splines, generalized boosted regression model, classification tree analysis, random forest, maximum entropy, artificial neural network, and support vector machines). This S4 class is obtained with `ensemble_modelling` or `ensemble`.

Slots

`uncertainty` raster. Between-algorithm variance map.

`algorithm.correlation` data frame. Between-algorithm correlation matrix.

`algorithm.evaluation` data frame. Evaluation of the ensemble SDM (available metrics include AUC, Kappa, sensitivity, specificity and proportion of correctly predicted occurrences) and identification of the optimal threshold to convert the habitat suitability map into a binary presence/absence map.

See Also

Algorithm.SDM an S4 class to represent an SDM based on a single algorithm, and Stacked.SDM an S4 class for SSDMs.

ensemble_modelling *Build an ensemble SDM that assembles multiple algorithms*

Description

This is a function to build an ensemble SDM that assembles multiple algorithms for a single species. The function takes as inputs an occurrence data frame made of presence/absence or presence-only records and a raster object for data extraction and projection. The function returns an S4 Ensemble.SDM class object containing the habitat suitability map, the binary map, the between-algorithm variance map and the associated evaluation tables (model evaluation, algorithm evaluation, algorithm correlation matrix and variable importance).

Usage

```
ensemble_modelling(algorithms, Occurrences, Env, Xcol = "Longitude",
  Ycol = "Latitude", Pcol = NULL, rep = 10, name = NULL, save = F,
  path = getwd(), PA = NULL, cv = "holdout", cv.param = c(0.7, 1),
  thresh = 1001, metric = "SES", axes.metric = "Pearson",
  uncertainty = T, tmp = F, ensemble.metric = c("AUC"),
  ensemble.thresh = c(0.75), weight = T, verbose = T, GUI = F, ...)
```

Arguments

algorithms	character. Choice of the algorithm(s) to be run (see details below).
Occurrences	data frame. Occurrence table (can be processed first by load_occ).
Env	raster object. Stacked raster object of environmental variables (can be processed first by load_var).
Xcol	character. Name of the column in the occurrence table containing Latitude or X coordinates.
Ycol	character. Name of the column in the occurrence table containing Longitude or Y coordinates.
Pcol	character. Name of the column in the occurrence table specifying whether a line is a presence or an absence, by setting presence to 1 and absence to 0. If NULL presence-only dataset is assumed.
rep	integer. Number of repetitions for each algorithm.
name	character. Optional name given to the final Ensemble.SDM produced (by default 'Ensemble.SDM').
save	logical. If set to true, the ensemble SDM is automatically saved.
path	character. If save is true, the path to the directory in which the ensemble SDM will be saved.
PA	list(nb, strat) defining the pseudo-absence selection strategy used in case of presence-only dataset. If PA is NULL, recommended PA selection strategy is used depending on the algorithm (see details below).
cv	character. Method of cross-validation used to evaluate the ensemble SDM (see details below).
cv.param	numeric. Parameters associated to the method of cross-validation used to evaluate the ensemble SDM (see details below).
thresh	numeric. A single integer value representing the number of equal interval threshold values between 0 and 1 (see optim.thresh).
metric	character. Metric used to compute the binary map threshold (see details below.)
axes.metric	Metric used to evaluate variable relative importance (see details below).
uncertainty	logical. If set to true, generates an uncertainty map and an algorithm correlation matrix.
tmp	logical. If set to true, the habitat suitability map of each algorithm is saved in a temporary file to release memory. But beware: if you close R, temporary files will be destroyed. To avoid any loss you can save your ensemble SDM with save.model. Depending on number, resolution and extent of models, temporary files can take a lot of disk space. Temporary files are written in R environment temporary folder.
ensemble.metric	character. Metric(s) used to select the best SDMs that will be included in the ensemble SDM (see details below).
ensemble.thresh	numeric. Threshold(s) associated with the metric(s) used to compute the selection.

`weight`	logical. Choose whether or not you want the SDMs to be weighted using the selection metric or, alternatively, the mean of the selection metrics.
`verbose`	logical. If set to true, allows the function to print text in the console.
`GUI`	logical. Don't take that argument into account (parameter for the user interface).
`...`	additional parameters for the algorithm modelling function (see details below).

Details

algorithms 'all' allows you to call directly all available algorithms. Currently, available algorithms include Generalized linear model (**GLM**), Generalized additive model (**GAM**), Multivariate adaptive regression splines (**MARS**), Generalized boosted regressions model (**GBM**), Classification tree analysis (**CTA**), Random forest (**RF**), Maximum entropy (**MAXENT**), Artificial neural network (**ANN**), and Support vector machines (**SVM**). Each algorithm has its own parameters settable with the **...** (see each algorithm section below to set their parameters).

"PA" list with two values: **nb** number of pseudo-absences selected, and **strat** strategy used to select pseudo-absences: either random selection or disk selection. We set default recommendation from Barbet-Massin et al. (2012) (see reference).

cv **Cross-validation** method used to split the occurrence dataset used for evaluation: **holdout** data are partitioned into a training set and an evaluation set using a fraction (*cv.param[1]*) and the operation can be repeated (*cv.param[2]*) times, **k-fold** data are partitioned into k (*cv.param[1]*) folds being k-1 times in the training set and once the evaluation set and the operation can be repeated (*cv.param[2]*) times, **LOO** (Leave One Out) each point is successively taken as evaluation data.

metric Choice of the metric used to compute the binary map threshold and the confusion matrix (by default SES as recommended by Liu et al. (2005), see reference below): **Kappa** maximizes the Kappa, **CCR** maximizes the proportion of correctly predicted observations, **TSS** (True Skill Statistic) maximizes the sum of sensitivity and specificity, **SES** uses the sensitivity-specificity equality, **LW** uses the lowest occurrence prediction probability, **ROC** minimizes the distance between the ROC plot (receiving operating characteristic curve) and the upper left corner (1,1).

axes.metric Choice of the metric used to evaluate the variable relative importance (difference between a full model and one with each variable successively omitted): **Pearson** (computes a simple Pearson's correlation *r* between predictions of the full model and the one without a variable, and returns the score *1-r*: the highest the value, the more influence the variable has on the model), **AUC**, **Kappa**, **sensitivity**, **specificity**, and **prop.correct** (proportion of correctly predicted occurrences).

ensemble.metric Ensemble metric(s) sed to select SDMs: **AUC**, **Kappa**, **sensitivity**, **specificity**, and **prop.correct** (proportion of correctly predicted occurrences).

"..." See algorithm in detail section

Value

an S4 Ensemble.SDM class object viewable with the `plot.model` function.

Generalized linear model (GLM)

Uses the `glm` function from the package 'stats', you can set the following parameters (see `glm` for more details):

test character. Test used to evaluate the SDM, default 'AIC'.

epsilon numeric. Positive convergence tolerance eps ; the iterations converge when |*dev - dev_old*|/(|*dev*| + *0.1)* < *eps*. By default, set to 10e-08.

maxit numeric. Integer giving the maximal number of IWLS (Iterative Weighted Last Squares) iterations, default 500.

Generalized additive model (GAM)

Uses the `gam` function from the package 'mgcv', you can set the following parameters (see `gam` for more details):

test character. Test used to evaluate the model, default 'AIC'.

epsilon numeric. This is used for judging conversion of the GLM IRLS (Iteratively Reweighted Least Squares) loop, default 10e-08.

maxit numeric. Maximum number of IRLS iterations to perform, default 500.

Multivariate adaptive regression splines (MARS)

Uses the `earth` function from the package 'earth', you can set the following parameters (see `earth` for more details):

degree integer. Maximum degree of interaction (Friedman's mi) ; 1 meaning build an additive model (i.e., no interaction terms). By default, set to 2.

Generalized boosted regressions model (GBM)

Uses the `gbm` function from the package 'gbm,' you can set the following parameters (see `gbm` for more details):

trees integer. The total number of trees to fit. This is equivalent to the number of iterations and the number of basis functions in the additive expansion. By default, set to 2500.

final.leave integer. minimum number of observations in the trees terminal nodes. Note that this is the actual number of observations not the total weight. By default, set to 1.

algocv integer. Number of cross-validations, default 3.

thresh.shrink integer. Number of cross-validation folds to perform. If cv.folds>1 then gbm, in addition to the usual fit, will perform a cross-validation. By default, set to 1e-03.

Classification tree analysis (CTA)

Uses the `rpart` function from the package 'rpart', you can set the following parameters (see `rpart` for more details):

final.leave integer. The minimum number of observations in any terminal node, default 1.

algocv integer. Number of cross-validations, default 3.

Random Forest (RF)

Uses the `randomForest` function from the package 'randomForest', you can set the following parameters (see `randomForest` for more details):

trees integer. Number of trees to grow. This should not be set to a too small number, to ensure that every input row gets predicted at least a few times. By default, set to 2500.

final.leave integer. Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). By default, set to 1.

Maximum Entropy (MAXENT)

Uses the `maxent` function from the package 'dismo'. Make sure that you have correctly installed the maxent.jar file in the folder ~\R\library\version\dismo\java available at https://www.cs.princeton.edu/~schapire/maxent/ (see `maxent` for more details).

Artificial Neural Network (ANN)

Uses the `nnet` function from the package 'nnet', you can set the following parameters (see `nnet` for more details):

maxit integer. Maximum number of iterations, default 500.

Support vector machines (SVM)

Uses the `svm` function from the package 'e1071', you can set the following parameters (see `svm` for more details):

epsilon float. Epsilon parameter in the insensitive loss function, default 1e-08.

algocv integer. If an integer value k>0 is specified, a k-fold cross-validation on the training data is performed to assess the quality of the model: the accuracy rate for classification and the Mean Squared Error for regression. By default, set to 3.

Warning

Depending on the raster object resolution the process can be more or less time- and memory-consuming.

References

M. Barbet-Massin, F. Jiguet, C. H. Albert, & W. Thuiller (2012) "Selecting pseudo-absences for species distribution models: how, where and how many?" *Methods Ecology and Evolution* 3:327-338 http://onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2011.00172.x/full

C. Liu, P. M. Berry, T. P. Dawson, R. & G. Pearson (2005) "Selecting thresholds of occurrence in the prediction of species distributions." *Ecography* 28:85-393 http://www.researchgate.net/publication/230246974_Selecting_Thresholds_of_Occurrence_in_the_Prediction_of_Species_Distributions

See Also

`modelling` to build SDMs with a single algorithm, `stack_modelling` to build SSDMs.

Examples

```
## Not run:
# Loading data
data(Env)
data(Occurrences)
Occurrences = subset(Occurrences, Occurrences$SPECIES == 'elliptica')

# ensemble SDM building
ESDM = ensemble_modelling(c('CTA', 'MARS'), Occurrences, Env, rep = 1,
                          Xcol = 'LONGITUDE', Ycol = 'LATITUDE',
                          ensemble.thresh = c(0.6))

# Results plotting
plot(ESDM)

## End(Not run)
```

Env *A stack of three environmental variables*

Description

A stack of three 30 arcsec-resolution rasters covering the north part of the main island of New Caledonia 'Grande Terre'. Climatic variables (RAINFALL and TEMPERATURE) are from the WorldClim database, and the SUBSTRATE map is from the IRD Atlas of New Caledonia (2012) (see reference below).

Usage

```
Env
```

Format

A stack of three rasters:

RAINFALL Annual mean rainfall (mm)

TEMPERATURE Annual mean temperature (x10 degree Celsius)

SUBSTRATE Substrate type (categorical variable)

References

R.J. Hijmans, C.H. & Graham (2006) "The ability of climate envelope models to predict the effect of climate change on species distributions." *Global Change Biology* 12:2272-2281 http://se-server.ethz.ch/staff/af/Fi159/H/Hi082_S.pdf

E. Fritsch (2012) "Les sols. Atlas de la Nouvelle-Caledonie (ed. by J. Bonvallot, J.-C. Gay and E. Habert)" *IRD-Congres de la Nouvelle-Caledonie, Marseille.* 73-76

gui *SSDM package Global User Interface*

Description

User interface of the SSDM package.

Usage

```
gui()
```

Details

If your environmental variables have an important size, you should gave enough memory to the interface with the (`maxmem` parameter).

Value

Open a window with a shiny app to use the SSDM package with an user-friendly interface.

Examples

```
## Not run:
gui()

## End(Not run)
```

load.model *Function to load ensemble SDMs and SSDMs*

Description

Allows to load S4 Ensemble.SDM and Stacked.SDM objects saved with their respective save function.

Usage

```
load_enm(name, path = getwd())

load_stack(name = "Stack", path = getwd(), GUI = F)
```

Arguments

`name`	character. Name of the folder that contains the model to be loaded.
`path`	character. Path to the directory containing the model to be loaded, by default the path to the current directory.
`GUI`	logical. Don't take that argument into account (parameter for the user interface).

Value

The corresponding SDM object.

See Also

save.model

---

load_occ *Load occurrence data*

Description

Function to load occurrence data from a table to perform modelling, ensemble_modelling or stack_modelling.

Usage

```
load_occ(path = getwd(), Env, file = NULL, ..., Xcol = "Longitude",
  Ycol = "Latitude", Spcol = NULL, GeoRes = T,
  reso = max(res(Env@layers[[1]])), verbose = T, GUI = F)
```

Arguments

path	character. Path to the directory that contains the occurrence table.
Env	raster stack. Environmental variables in the form of a raster stack used to perform spatial thinning (can be the result of the load_var function).
file	character. File containing the occurrence table, if NULL (default) the .csv file located in the path will be loaded.
...	additional parameters given to read.csv.
Xcol	character. Name of the column in the occurrence table containing Latitude or X coordinates.
Ycol	character. Name of the column in the occurrence table containing Longitude or Y coordinates.
Spcol	character. Name of the column containing species names or IDs.
GeoRes	logical. Geographical thinning will be perform on occurrences to limit geographical biases in the SDMs.
reso	numeric. Resolution used to perform the geographical thinning, by default the resolution of the raster stack (Env).
verbose	logical. If set to true, allows the function to print text in the console.
GUI	logical. Don't take that argument into account (parameter for the user interface).

Value

A data frame containing the occurrence dataset (spatially thinned or not).

See Also

load_var to load environmental variables.

Examples

```
## Not run:
load.occ(path)

## End(Not run)
```

---

load_var *Load environmental variables*

Description

Function to load environmental variables in the form of rasters to perform modelling, ensemble_modelling or stack_modelling.

Usage

```
load_var(path = getwd(), files = NULL, format = c(".grd", ".tif", ".asc",
  ".sdat", ".rst", ".nc", ".envi", ".bil", ".img"), categorical = NULL,
  Norm = T, tmp = T, verbose = T, GUI = F)
```

Arguments

path	character. Path to the directory that contains the environmental variables files.
files	character. Files containing the environmental variables If NULL (default) all files present in the path in the selected format will be loaded.
format	character. Format of environmental variables files (including .grd, .tif, .asc, .sdat, .rst, .nc, .tif, .envi, .bil, .img).
categorical	character. Specify whether an environmental variable is a categorical variable.
Norm	logical. If set to true, normalizes environmental variables between 0 and 1.
tmp	logical. If set to true, rasters are read in temporary file avoiding to overload the random access memory. But beware: if you close R, temporary files will be destroyed.
verbose	logical. If set to true, allows the function to print text in the console.
GUI	logical. Don't take that argument into account (parameter for the user interface).

Value

A stack containing the environmental rasters (normalized or not).

See Also

load_occ to load occurrences.

Examples

```
## Not run:
load.var(path)

## End(Not run)
```

`modelling` *Build an SDM using a single algorithm*

Description

This is a function to build an SDM with one algorithm for a single species. The function takes as inputs an occurrence data frame made of presence/absence or presence-only records and a raster object for data extraction and projection. The function returns an S4 Algorithm.SDM class object containing the habitat suitability map, the binary map and the evaluation table.

Usage

```
modelling(algorithm, Occurrences, Env, Xcol = "Longitude",
  Ycol = "Latitude", Pcol = NULL, name = NULL, PA = NULL,
  cv = "holdout", cv.param = c(0.7, 2), thresh = 1001, metric = "SES",
  axes.metric = "Pearson", select = F, select.metric = c("AUC"),
  select.thresh = c(0.75), verbose = T, GUI = F, ...)
```

Arguments

`algorithm`	character. Choice of the algorithm to be run (see details below).
`Occurrences`	data frame. Occurrence table (can be processed first by `load_occ`).
`Env`	raster object. Raster object of environmental variable (can be processed first by `load_var`).
`Xcol`	character. Name of the column in the occurrence table containing Latitude or X coordinates.
`Ycol`	character. Name of the column in the occurrence table containing Longitude or Y coordinates.
`Pcol`	character. Name of the column in the occurrence table specifying whether a line is a presence or an absence, by setting presence to 1 and absence to 0. If NULL presence-only dataset is assumed.
`name`	character. Optional name given to the final SDM produced (by default 'Algorithm.SDM').
`PA`	list(nb, strat) defining the pseudo-absence selection strategy used in case of presence-only dataset. If PA is NULL, recommended PA selection strategy is used depending on the algorithms (see details below).
`cv`	character. Method of cross-validation used to evaluate the SDM (see details below).

cv.param	numeric. Parameters associated to the method of cross-validation used to evaluate the SDM (see details below).
thresh	numeric. A single integer value representing the number of equal interval threshold values between 0 and 1 (see optim.thresh).
metric	character. Metric used to compute the binary map threshold (see details below.)
axes.metric	Metric used to evaluate variable relative importance (see details below).
select	logical. If set to true, models are evaluated before being projected, and not kept if they don't meet selection criteria (see details below).
select.metric	character. Metric(s) used to pre-select SDMs that reach a sufficient quality (see details below).
select.thresh	numeric. Threshold(s) associated with the metric(s) used to compute the selection.
verbose	logical. If set to true, allows the function to print text in the console.
GUI	logical. Don't take that argument into account (parameter for the user interface).
...	additional parameters for the algorithm modelling function (see details below).

Details

algorithm 'all' allows you to call directly all available algorithms. Currently, available algorithms include Generalized linear model (**GLM**), Generalized additive model (**GAM**), Multivariate adaptive regression splines (**MARS**), Generalized boosted regressions model (**GBM**), Classification tree analysis (**CTA**), Random forest (**RF**), Maximum entropy (**MAXENT**), Artificial neural network (**ANN**), and Support vector machines (**SVM**). Each algorithm has its own parameters settable with the **...** (see each algorithm section below to set their parameters).

"PA" list with two values: **nb** number of pseudo-absences selected, and **strat** strategy used to select pseudo-absences: either random selection or disk selection. We set default recommendation from Barbet-Massin et al. (2012) (see reference).

cv **Cross-validation** method used to split the occurrence dataset used for evaluation: **holdout** data are partitioned into a training set and an evaluation set using a fraction (*cv.param[1]*) and the operation can be repeated (*cv.param[2]*) times, **k-fold** data are partitioned into k (*cv.param[1]*) folds being k-1 times in the training set and once the evaluation set and the operation can be repeated (*cv.param[2]*) times, **LOO** (Leave One Out) each point is successively taken as evaluation data.

metric Choice of the metric used to compute the binary map threshold and the confusion matrix (by default SES as recommended by Liu et al. (2005), see reference below): **Kappa** maximizes the Kappa, **CCR** maximizes the proportion of correctly predicted observations, **TSS** (True Skill Statistic) maximizes the sum of sensitivity and specificity, **SES** uses the sensitivity-specificity equality, **LW** uses the lowest occurrence prediction probability, **ROC** minimizes the distance between the ROC plot (receiving operating curve) and the upper left corner (1,1).

axes.metric Choice of the metric used to evaluate the variable relative importance (difference between a full model and one with each variable successively omitted): **Pearson** (computes a simple Pearson's correlation *r* between predictions of the full model and the one without a variable, and returns the score *1-r*: the highest the value, the more influence the variable has on the model), **AUC**, **Kappa**, **sensitivity**, **specificity**, and **prop.correct** (proportion of correctly predicted occurrences).

select.metric Selection metric(s) used to select SDMs: **AUC**, **Kappa**, **sensitivity**, **specificity**, and **prop.correct** (proportion of correctly predicted occurrences).

"..." See algorithm in detail section

Value

an S4 Algorithm.SDM Class object viewable with the `plot.model` method

Generalized linear model (GLM)

Uses the `glm` function from the package 'stats', you can set the following parameters (see `glm` for more details):

test character. Test used to evaluate the SDM, default 'AIC'.

epsilon numeric. Positive convergence tolerance eps ; the iterations converge when $|dev - dev\_old|/(|dev| + 0.1) < eps$. By default, set to 10e-08.

maxit numeric. Integer giving the maximal number of IWLS (Iterative Weighted Last Squares) iterations, default 500.

Generalized additive model (GAM)

Uses the `gam` function from the package 'mgcv', you can set the following parameters (see `gam` for more details):

test character. Test used to evaluate the model, default 'AIC'.

epsilon numeric. This is used for judging conversion of the GLM IRLS (Iteratively Reweighted Least Squares) loop, default 10e-08.

maxit numeric. Maximum number of IRLS iterations to perform, default 500.

Multivariate adaptive regression splines (MARS)

Uses the `earth` function from the package 'earth', you can set the following parameters (see `earth` for more details):

degree integer. Maximum degree of interaction (Friedman's mi) ; 1 meaning build an additive model (i.e., no interaction terms). By default, set to 2.

Generalized boosted regressions model (GBM)

Uses the `gbm` function from the package 'gbm,' you can set the following parameters (see `gbm` for more details):

trees integer. The total number of trees to fit. This is equivalent to the number of iterations and the number of basis functions in the additive expansion. By default, set to 2500.

final.leave integer. minimum number of observations in the trees terminal nodes. Note that this is the actual number of observations not the total weight. By default, set to 1.

algocv integer. Number of cross-validations, default 3.

thresh.shrink integer. Number of cross-validation folds to perform. If cv.folds>1 then gbm, in addition to the usual fit, will perform a cross-validation. By default, set to 1e-03.

Classification tree analysis (CTA)

Uses the `rpart` function from the package 'rpart', you can set the following parameters (see `rpart` for more details):

final.leave integer. The minimum number of observations in any terminal node, default 1.

algocv integer. Number of cross-validations, default 3.

Random Forest (RF)

Uses the `randomForest` function from the package 'randomForest', you can set the following parameters (see `randomForest` for more details):

trees integer. Number of trees to grow. This should not be set to a too small number, to ensure that every input row gets predicted at least a few times. By default, set to 2500.

final.leave integer. Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). By default, set to 1.

Maximum Entropy (MAXENT)

Uses the `maxent` function from the package 'dismo'. Make sure that you have correctly installed the maxent.jar file in the folder ~\R\library\version\dismo\java available at `https://www.cs.princeton.edu/~schapire/maxent/` (see `maxent` for more details).

Artificial Neural Network (ANN)

Uses the `nnet` function from the package 'nnet', you can set the following parameters (see `nnet` for more details):

maxit integer. Maximum number of iterations, default 500.

Support vector machines (SVM)

Uses the `svm` function from the package 'e1071', you can set the following parameters (see `svm` for more details):

epsilon float. Epsilon parameter in the insensitive loss function, default 1e-08.

algocv integer. If an integer value k>0 is specified, a k-fold cross-validation on the training data is performed to assess the quality of the model: the accuracy rate for classification and the Mean Squared Error for regression. By default, set to 3.

Warning

Depending on the raster object resolution the process can be more or less time- and memory-consuming.

References

M. Barbet-Massin, F. Jiguet, C. H. Albert, & W. Thuiller (2012) "Selecting pseudo-absences for species distribution models: how, where and how many?" *Methods Ecology and Evolution* 3:327-338 http://onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2011.00172.x/full

C. Liu, P. M. Berry, T. P. Dawson, R. & G. Pearson (2005) "Selecting thresholds of occurrence in the prediction of species distributions." *Ecography* 28:85-393 http://www.researchgate.net/publication/230246974_Selecting_Thresholds_of_Occurrence_in_the_Prediction_of_Species_Distributions

See Also

ensemble_modelling to build ensemble SDMs, stack_modelling to build SSDMs.

Examples

```
# Loading data
data(Env)
data(Occurrences)
Occurrences = subset(Occurrences, Occurrences$SPECIES == 'elliptica')

# SDM building
SDM = modelling('GLM', Occurrences, Env, Xcol = 'LONGITUDE', Ycol = 'LATITUDE')

# Results plotting
## Not run:
plot(SDM)

## End(Not run)
```

Occurrences *Plant occurrence data frame*

Description

A dataset containing a list of plant occurrences of five Cryptocarya species native to New Caledonia. Occurrence data come from the Noumea Herbarium (NOU) and NC-PIPPN network (see Ibanez et al (2014) in reference below).

Usage

```
Occurrences
```

Format

A data frame with 57 rows and 3 variables:

SPECIES Species of the occurrence

LONGITUDE Longitude of the occurrence

LATITUDE Latitude of the occurrence

References

T. Ibanez, J. Munzinger, G. Dagostini, V. Hequet, F. Rigault, T. Jaffre, & P. Birnbaum (2014) "Structural and floristic characteristics of mixed rainforest in New Caledonia: new data from the New Caledonian Plant Inventory and Permanent Plot Network (NC-PIPPN)." *Applied Vegetation Science* 17:386-397

http://www.researchgate.net/profile/Jerome_Munzinger/publication/258499017_Structural_and_floristic_diversity_of_mixed_tropical_rain_forest_in_New_Caledonia_new_data_from_the_New_Caledonian_Plant_Inventory_and_Permanent_Plot_Network_%28NC-PIPPN%29/links/0deec52b8b1996488e000000.pdf

---

`plot.model` *Plot SDMs, ensemble SDMs, and SSDMs*

Description

Allows to plot S4 Algorithm.SDM, Ensemble.SDM and Stacked.SDM class objects.

Usage

```
## S4 method for signature 'Stacked.SDM,ANY'
plot(x, y, ...)

## S4 method for signature 'SDM,ANY'
plot(x, y, ...)
```

Arguments

`x`	Object to be plotted (S4 Algorithm.SDM, Ensemble.SDM or Stacked.SDM object).
`y,...`	Plot-based parameter not used.

Value

Open a window with a shiny app rendering all the results (habitat suitability map, binary map, evaluation table, variable importance and/or between-algorithm variance map, and/or algorithm evaluation, and/or algorithm correlation matrix and/or local species richness map) in a user-friendly interface.

save.model *Save ensemble SDMs and SSDMs*

Description

Allows to save S4 Ensemble.SDM and Stacked.SDM class objects.

Usage

```
save.enm(enm, name = strsplit(enm@name, ".", fixed = T)[[1]][1],
  path = getwd(), verbose = T, GUI = F)

## S4 method for signature 'Ensemble.SDM'
save.enm(enm, name = strsplit(enm@name, ".", fixed =
  T)[[1]][1], path = getwd(), verbose = T, GUI = F)

save.stack(stack, name = "Stack", path = getwd(), verbose = T, GUI = F)

## S4 method for signature 'Stacked.SDM'
save.stack(stack, name = "Stack", path = getwd(),
  verbose = T, GUI = F)
```

Arguments

enm	Ensemble.SDM. Ensemble SDM to be saved.
name	character. Folder name of the model to save.
path	character. Path to the directory chosen to save the SDM, by default the path to the current directory.
verbose	logical. If set to true, allows the function to print text in the console.
GUI	logical. Don't take that argument into account (parameter for the user interface).
stack	Stacked.SDM. SSDM to be saved.

Value

Nothing in R environment. Creates folders, tables and rasters associated to the SDM. Tables are in .csv and rasters in .grd/.gri.

See Also

load.model

SSDM *SSDM: Stacked species distribution modelling*

Description

SSDM is a package to map species richness and endemism based on stacked species distribution models (SSDM). Individual SDMs can be created using a single or multiple algorithms (ensemble SDMs). For each species, an SDM can yield a habitat suitability map, a binary map, a between-algorithm variance map, and can assess variable importance, algorithm accuracy, and between-algorithm correlation. Methods to stack individual SDMs include summing individual probabilities and thresholding then summing. Thresholding can be based on a specific evaluation metric or by drawing repeatedly from a Bernouilli distribution. The SSDM package also provides a user-friendly interface (`gui`).

Details

SSDM provides five categories of functions (that you can find in details below): Data preparation, Modelling main functions, Model main methods, Model classes, and Miscellaneous.

Data preparation

`load_occ` Load occurrence data

`load_var` Load environmental variables

Modelling main functions

`modelling` Build an SDM using a single algorithm

`ensemble_modelling` Build an SDM that assembles multiple algorithms

`stack_modelling` Build an SSDMs that assembles multiple algorithms and species

Model main methods

`ensemble,Algorithm.SDM-method` Build an ensemble SDM

`stacking,Ensemble.SDM-method` Build an SSDM

`update,Stacked.SDM-method` Update a previous SSDM with new occurrence data

Model classes

`Algorithm.SDM` S4 class to represent SDMs

`Ensemble.SDM` S4 class to represent ensemble SDMs

`Stacked.SDM` S4 class to represent SSDMs

Miscellaneous

`gui` User-friendly interface for SSDM package

`plot.model` Plot SDMs

`save.model` Save SDMs

`load.model` Load SDMs

`Stacked.SDM-class` *An S4 class to represent SSDMs*

Description

This is an S4 class to represent SSDMs that assembles multiple algorithms (including generalized linear model, general additive model, multivariate adaptive splines, generalized boosted regression model, classification tree analysis, random forest, maximum entropy, artificial neural network, and support vector machines) built for multiple species. It is obtained with `stack_modelling` or `stacking`.

Slots

`name` character. Name of the SSDM (by default 'Species.SSDM').

`diversity.map` raster. Local species richness map produced by the SSDM.

`endemism.map` raster. Endemism map produced by the SSDM (see Crisp et al (2011) in references).

`uncertainty` raster. Between-algorithm variance map.

`evaluation` data frame. Evaluation of the SSDM (AUC, Kappa, omission rate, sensitivity, specificity, proportion of correctly predicted occurrences).

`variable.importance` data frame. Relative importance of each variable in the SSDM.

`algorithm.correlation` data frame. Between-algorithm correlation matrix.

`enms` list. List of ensemble SDMs used in the SSDM.

`parameters` data frame. Parameters used to build the SSDM.

`algorithm.evaluation` data frame. Evaluation of the algorithm averaging the metrics of all SDMs (AUC, Kappa, omission rate, sensitivity, specificity, proportion of correctly predicted occurrences).

References

M. D. Crisp, S. Laffan, H. P. Linder & A. Monro (2001) "Endemism in the Australian flora" *Journal of Biogeography* 28:183-198 http://biology-assets.anu.edu.au/hosted_sites/Crisp/pdfs/Crisp2001_endemism.pdf

See Also

Ensemble.SDM an S4 class to represent ensemble SDMs, and Algorithm.SDM an S4 class to represent SDMs.

stacking *Stack different ensemble SDMs in an SSDM*

Description

This is a function to stack several ensemble SDMs in an SSDM. The function takes as inputs several S4 Ensemble.SDM class objects produced with `ensemble_modelling` or `ensemble` functions. The function returns an S4 Stacked.SDM class object containing the local species richness map, the between-algorithm variance map, and all evaluation tables coming with (model evaluation, algorithm evaluation, algorithm correlation matrix and variable importance), and a list of ensemble SDMs for each species (see `ensemble_modelling`).

Usage

```
stacking(enm, ..., name = NULL, method = "P", rep.B = 1000,
  range = NULL, endemism = c("WEI", "Binary"), verbose = T, GUI = F)

## S4 method for signature 'Ensemble.SDM'
stacking(enm, ..., name = NULL, method = "P",
  rep.B = 1000, range = NULL, endemism = c("WEI", "Binary"),
  verbose = T, GUI = F)
```

Arguments

`enm,...`	character. Ensemble SDMs to be stacked.
`name`	character. Optional name given to the final SSDM produced (by default 'Species.SDM').
`method`	character. Define the method used to create the local species richness map (see details below).
`rep.B`	integer. If the method used to create the local species richness is the random bernoulli (**B**), rep.B parameter defines the number of repetitions used to create binary maps for each species.
`range`	integer. Set a value of range restriction (in pixels) around presences occurrences on habitat suitability maps (all further points will have a null probability, see Crisp et al (2011) in references). If NULL, no range restriction will be applied.
`endemism`	character. Define the method used to create an endemism map (see details below).
`verbose`	logical. If set to true, allows the function to print text in the console.
`GUI`	logical. Don't take that argument into account (parameter for the user interface).

Value

an S4 Stacked.SDM class object viewable with the `plot.model` function.

Methods: Choice of the method used to compute the local species richness map (see Calabrez et al. (2014) for more informations, see reference below):

P (Probablity) sum probabilities of habitat suitability maps

B (Random bernoulli) draw repeatedly from a Bernoulli distribution

T (Threshold) sum the binary map obtained with the thresholding (depending on the metric, see metric parameter).

Endemism: Choice of the method used to compute the endemism map (see Crisp et al. (2001) for more information, see reference below):

NULL No endemism map

WEI (Weighted Endemism Index) Endemism map built by counting all species in each cell and weighting each by the inverse of its range

CWEI (Corrected Weighted Endemism Index) Endemism map built by dividing the weighted endemism index by the total count of species in the cell.

First string of the character is the method either WEI or CWEI, and in those cases second string of the vector is used to precise range calculation, whether the total number of occurrences **'NbOcc'** whether the surface of the binary map species distribution **'Binary'**.

References

C. Liu, P. M. Berry, T. P. Dawson, R. & G. Pearson (2005) "Selecting thresholds of occurrence in the prediction of species distributions." *Ecography* 28:85-393 http://www.researchgate.net/publication/230246974_Selecting_Thresholds_of_Occurrence_in_the_Prediction_of_Species_Distributions

J.M. Calabrese, G. Certain, C. Kraan, & C.F. Dormann (2014) "Stacking species distribution models and adjusting bias by linking them to macroecological models." *Global Ecology and Biogeography* 23:99-112 http://portal.uni-freiburg.de/biometrie/mitarbeiter/dormann/calabrese2013globalecolbiogeogr pdf

M. D. Crisp, S. Laffan, H. P. Linder & A. Monro (2001) "Endemism in the Australian flora" *Journal of Biogeography* 28:183-198 http://biology-assets.anu.edu.au/hosted_sites/Crisp/pdfs/Crisp2001_endemism.pdf

See Also

stack_modelling to build SSDMs.

Examples

```
## Not run:
# Loading data
data(Env)
data(Occurrences)
Occ1 = subset(Occurrences, Occurrences$SPECIES == 'elliptica')
Occ2 = subset(Occurrences, Occurrences$SPECIES == 'gracilis')

# SSDM building
ESDM1 = ensemble_modelling(c('CTA', 'SVM'), Occ1, Env, rep = 1,
                           Xcol = 'LONGITUDE', Ycol = 'LATITUDE',
                           name = 'elliptica', ensemble.thresh = c(0.6))
```

```
ESDM2 = ensemble_modelling(c('CTA', 'SVM'), Occ2, Env, rep = 1,
                           Xcol = 'LONGITUDE', Ycol = 'LATITUDE',
                           name = 'gracilis', ensemble.thresh = c(0.6))
SSDM = stacking(ESDM1, ESDM2)

# Results plotting
plot(SSDM)

## End(Not run)
```

`stack_modelling` *Build an SSDM that assembles multiple algorithms and species*

Description

This is a function to build an SSDM that assembles multiple algorithm and species. The function takes as inputs an occurrence data frame made of presence/absence or presence-only records and a raster object for data extraction and projection. The function returns an S4 Stacked.SDM class object containing the local species richness map, the between-algorithm variance map, and all evaluation tables coming with (model evaluation, algorithm evaluation, algorithm correlation matrix and variable importance), and a list of ensemble SDMs for each species (see `ensemble_modelling`).

Usage

```
stack_modelling(algorithms, Occurrences, Env, Xcol = "Longitude",
  Ycol = "Latitude", Pcol = NULL, Spcol = "SpeciesID", rep = 10,
  name = NULL, save = F, path = getwd(), PA = NULL, cv = "holdout",
  cv.param = c(0.7, 1), thresh = 1001, axes.metric = "Pearson",
  uncertainty = T, tmp = F, ensemble.metric = c("AUC"),
  ensemble.thresh = c(0.75), weight = T, method = "P", metric = "SES",
  rep.B = 1000, range = NULL, endemism = c("WEI", "Binary"),
  verbose = T, GUI = F, cores = 1, ...)
```

Arguments

`algorithms`	character. Choice of the algorithm(s) to be run (see details below).
`Occurrences`	data frame. Occurrence table (can be processed first by `load_occ`).
`Env`	raster object. Raster object of environmental variables (can be processed first by `load_var`).
`Xcol`	character. Name of the column in the occurrence table containing Latitude or X coordinates.
`Ycol`	character. Name of the column in the occurrence table containing Longitude or Y coordinates.
`Pcol`	character. Name of the column in the occurrence table specifying whether a line is a presence or an absence, by setting presence to 1 and absence to 0. If NULL presence-only dataset is assumed.

Spcol	character. Name of the column containing species names or IDs.
rep	integer. Number of repetitions for each algorithm.
name	character. Optional name given to the final Ensemble.SDM produced.
save	logical. If set to true, the SSDM is automatically saved.
path	character. If save is true, the path to the directory in which the ensemble SDM will be saved.
PA	list(nb, strat) defining the pseudo-absence selection strategy used in case of presence-only dataset. If PA is NULL, recommended PA selection strategy is used depending on the algorithm (see details below).
cv	character. Method of cross-validation used to evaluate the ensemble SDM (see details below).
cv.param	numeric. Parameters associated with the method of cross-validation used to evaluate the ensemble SDM (see details below).
thresh	numeric. A single integer value representing the number of equal interval threshold values between 0 and 1 (see optim.thresh).
axes.metric	Metric used to evaluate variable relative importance (see details below).
uncertainty	logical. If set to true, generates an uncertainty map and an algorithm correlation matrix.
tmp	logical. If set to true, the habitat suitability map of each algorithms is saved in a temporary file to release memory. But beware: if you close R, temporary files will be destroyed. To avoid any loss you can save your SSDM with save.model. Depending on number, resolution and extent of models, temporary files can take a lot of disk space. Temporary files are written in R environment temporary folder.
ensemble.metric	character. Metric(s) used to select the best SDMs that will be included in the ensemble SDM (see details below).
ensemble.thresh	numeric. Threshold(s) associated with the metric(s) used to compute the selection.
weight	logical. Choose whether or not you want the SDMs to be weighted using the selection metric or, alternatively, the mean of the selection metrics.
method	character. Define the method used to create the local species richness map (see details below).
metric	character. Metric used to compute the binary map threshold (see details below.)
rep.B	integer. If the method used to create the local species richness is the random bernoulli (**B**), rep.B parameter defines the number of repetitions used to create binary maps for each species.
range	integer. Set a value of range restriction (in pixels) around presences occurrences on habitat suitability maps (all further points will have a null probability, see Crisp et al (2011) in references). If NULL, no range restriction will be applied.
endemism	character. Define the method used to create an endemism map (see details below).

verbose	logical. If set to true, allows the function to print text in the console.
GUI	logical. Don't take that argument into account (parameter for the user interface).
cores	integer. Specify the number of CPU cores used to do the computing. You can use detectCores) to automatically used all you available CPU cores.
...	additional parameters for the algorithm modelling function (see details below).

Details

algorithms 'all' allows you to call directly all available algorithms. Currently, available algorithms include Generalized linear model (**GLM**), Generalized additive model (**GAM**), Multivariate adaptive regression splines (**MARS**), Generalized boosted regressions model (**GBM**), Classification tree analysis (**CTA**), Random forest (**RF**), Maximum entropy (**MAXENT**), Artificial neural network (**ANN**), and Support vector machines (**SVM**). Each algorithm has its own parameters settable with the **...** (see each algorithm section below to set their parameters).

"PA" list with two values: **nb** number of pseudo-absences selected, and **strat** strategy used to select pseudo-absences: either random selection or disk selection. We set default recommendation from Barbet-Massin et al. (2012) (see reference).

cv **Cross-validation** method used to split the occurrence dataset used for evaluation: **holdout** data are partitioned into a training set and an evaluation set using a fraction (*cv.param[1]*) and the operation can be repeated (*cv.param[2]*) times, **k-fold** data are partitioned into k (*cv.param[1]*) folds being k-1 times in the training set and once the evaluation set and the operation can be repeated (*cv.param[2]*) times, **LOO** (Leave One Out) each point is successively taken as evaluation data.

metric Choice of the metric used to compute the binary map threshold and the confusion matrix (by default SES as recommended by Liu et al. (2005), see reference below): **Kappa** maximizes the Kappa, **CCR** maximizes the proportion of correctly predicted observations, **TSS** (True Skill Statistic) maximizes the sum of sensitivity and specificity, **SES** uses the sensitivity-specificity equality, **LW** uses the lowest occurrence prediction probability, **ROC** minimizes the distance between the ROC plot (receiving operating curve) and the upper left corner (1,1).

axes.metric Choice of the metric used to evaluate the variable relative importance (difference between a full model and one with each variable successively omitted): **Pearson** (computes a simple Pearson's correlation *r* between predictions of the full model and the one without a variable, and returns the score *1-r*: the highest the value, the more influence the variable has on the model), **AUC**, **Kappa**, **sensitivity**, **specificity**, and **prop.correct** (proportion of correctly predicted occurrences).

ensemble.metric Ensemble metric(s) used to select SDMs: **AUC**, **Kappa**, **sensitivity**, **specificity**, and **prop.correct** (proportion of correctly predicted occurrences).

method Choice of the method used to compute the local species richness map (see Calabrez et al. (2014) for more informations, see reference below): **P** (Probablity) sum probabilities of habitat suitability maps, **B** (Random Bernoulli) drawing repeatedly from a Bernoulli distribution, **T** (Threshold) sum the binary map obtained with the thresholding (depending on the metric, see metric parameter).

endemism Choice of the method used to compute the endemism map (see Crisp et al. (2001) for more information, see reference below): **NULL** No endemism map, **WEI** (Weighted Endemism Index) Endemism map built by counting all species in each cell and weighting each by the inverse of its range, **CWEI** (Corrected Weighted Endemism Index) Endemism map

built by dividing the weighted endemism index by the total count of species in the cell. First string of the character is the method either WEI or CWEI, and in those cases second string of the vector is used to precise range calculation, whether the total number of occurrences **'NbOcc'** whether the surface of the binary map species distribution **'Binary'**.

... See algorithm in detail section

Value

an S4 Stacked.SDM class object viewable with the `plot.model` function.

Generalized linear model (GLM)

Uses the `glm` function from the package 'stats', you can set the following parameters (see `glm` for more details):

test character. Test used to evaluate the SDM, default 'AIC'.

epsilon numeric. Positive convergence tolerance eps ; the iterations converge when $|dev - dev\_old|/(|dev| + 0.1) < eps$. By default, set to 10e-08.

maxit numeric. Integer giving the maximal number of IWLS (Iterative Weighted Last Squares) iterations, default 500.

Generalized additive model (GAM)

Uses the `gam` function from the package 'mgcv', you can set the following parameters (see `gam` for more details):

test character. Test used to evaluate the model, default 'AIC'.

epsilon numeric. This is used for judging conversion of the GLM IRLS (Iteratively Reweighted Least Squares) loop, default 10e-08.

maxit numeric. Maximum number of IRLS iterations to perform, default 500.

Multivariate adaptive regression splines (MARS)

Uses the `earth` function from the package 'earth', you can set the following parameters (see `earth` for more details):

degree integer. Maximum degree of interaction (Friedman's mi) ; 1 meaning build an additive model (i.e., no interaction terms). By default, set to 2.

Generalized boosted regressions model (GBM)

Uses the `gbm` function from the package 'gbm,' you can set the following parameters (see `gbm` for more details):

trees integer. The total number of trees to fit. This is equivalent to the number of iterations and the number of basis functions in the additive expansion. By default, set to 2500.

final.leave integer. minimum number of observations in the trees terminal nodes. Note that this is the actual number of observations not the total weight. By default, set to 1.

algocv integer. Number of cross-validations, default 3.

thresh.shrink integer. Number of cross-validation folds to perform. If cv.folds>1 then gbm, in addition to the usual fit, will perform a cross-validation. By default, set to 1e-03.

Classification tree analysis (CTA)

Uses the `rpart` function from the package 'rpart', you can set the following parameters (see `rpart` for more details):

final.leave integer. The minimum number of observations in any terminal node, default 1.

algocv integer. Number of cross-validations, default 3.

Random Forest (RF)

Uses the `randomForest` function from the package 'randomForest', you can set the following parameters (see `randomForest` for more details):

trees integer. Number of trees to grow. This should not be set to a too small number, to ensure that every input row gets predicted at least a few times. By default, set to 2500.

final.leave integer. Minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). By default, set to 1.

Maximum Entropy (MAXENT)

Uses the `maxent` function from the package 'dismo'. Make sure that you have correctly installed the maxent.jar file in the folder ~\R\library\version\dismo\java available at `https://www.cs.princeton.edu/~schapire/maxent/` (see `maxent` for more details).

Artificial Neural Network (ANN)

Uses the `nnet` function from the package 'nnet', you can set the following parameters (see `nnet` for more details):

maxit integer. Maximum number of iterations, default 500.

Support vector machines (SVM)

Uses the `svm` function from the package 'e1071', you can set the following parameters (see `svm` for more details):

epsilon float. Epsilon parameter in the insensitive loss function, default 1e-08.

algocv integer. If an integer value k>0 is specified, a k-fold cross-validation on the training data is performed to assess the quality of the model: the accuracy rate for classification and the Mean Squared Error for regression. By default, set to 3.

Warning

Depending on the raster object resolution the process can be more or less time- and memory-consuming.

References

M. Barbet-Massin, F. Jiguet, C. H. Albert, & W. Thuiller (2012) "Selecting pseudo-absences for species distribution models: how, where and how many?" *Methods Ecology and Evolution* 3:327-338 http://onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2011.00172.x/full

C. Liu, P. M. Berry, T. P. Dawson, R. & G. Pearson (2005) "Selecting thresholds of occurrence in the prediction of species distributions." *Ecography* 28:85-393 http://www.researchgate.net/publication/230246974_Selecting_Thresholds_of_Occurrence_in_the_Prediction_of_Species_Distributions

J.M. Calabrese, G. Certain, C. Kraan, & C.F. Dormann (2014) "Stacking species distribution models and adjusting bias by linking them to macroecological models." *Global Ecology and Biogeography* 23:99-112 http://portal.uni-freiburg.de/biometrie/mitarbeiter/dormann/calabrese2013globalecolbiogeogr.pdf

M. D. Crisp, S. Laffan, H. P. Linder & A. Monro (2001) "Endemism in the Australian flora" *Journal of Biogeography* 28:183-198 http://biology-assets.anu.edu.au/hosted_sites/Crisp/pdfs/Crisp2001_endemism.pdf

See Also

modelling to build simple SDMs.

Examples

```
## Not run:
# Loading data
data(Env)
data(Occurrences)

# SSDM building
SSDM = stack_modelling(c('CTA', 'SVM'), Occurrences, Env, rep = 1,
                       Xcol = 'LONGITUDE', Ycol = 'LATITUDE',
                       Spcol = 'SPECIES')

# Results plotting
plot(SSDM)

## End(Not run)
```

---

update,Stacked.SDM-method

Update a previous SSDM

Description

Update a previous SSDM with new occurrence data. The function takes as inputs updated or new occurrence data from one species, previous environmental variables, and an S4 Stacked.SDM class object containing a previously built SSDM.

Usage

```
## S4 method for signature 'Stacked.SDM'
update(object, Occurrences, Env, Xcol = "Longitude",
  Ycol = "Latitude", Pcol = NULL, Spname = NULL, name = stack@name,
  save = F, path = getwd(), thresh = 1001, tmp = F, verbose = T,
  GUI = F, ...)
```

Arguments

object	Stacked.SDM. The previously built SSDM.
Occurrences	data frame. New or updated occurrence table (can be processed first by load_occ).
Env	raster object. Environment raster object (can be processed first by load_var).
Xcol	character. Name of the column in the occurrence table containing Latitude or X coordinates.
Ycol	character. Name of the column in the occurrence table containing Longitude or Y coordinates.
Pcol	character. Name of the column in the occurrence table specifying whether a line is a presence or an absence, by setting presence to 1 and absence to 0. If NULL presence-only dataset is assumed.
Spname	character. Name of the new or updated species.
name	character. Optional name given to the final SSDM produced, by default it's the name of the previous SSDM.
save	logical. If set to true, the model is automatically saved.
path	character. Name of the path to the directory to contain the saved SSDM.
thresh	numeric. A single integer value representing the number of equal interval threshold values between 0 and 1 (see optim.thresh).
tmp	logical. If set to true, the habitat suitability map of each algorithm is saved in a temporary file to release memory. But beware: if you close R, temporary files will be destroyed. To avoid any loss you can save your model with save.model.
verbose	logical. If set to true, allows the function to print text in the console.
GUI	logical. Don't take that argument into account (parameter for the user interface).
...	additional parameters for the algorithm modelling function (see details below).

Value

an S4 Stacked.SDM class object viewable with the plot.model function.

See Also

stack_modelling to build SSDMs.

Examples

```
## Not run:
update(stack, Occurrences, Env, Spname = 'NewSpecie')

## End(Not run)
```

Index