

APPLICATION

BLOCKCV: An R package for generating spatially or environmentally separated folds for *k*-fold cross-validation of species distribution models

Roobeh Valavi  | Jane Elith  | José J. Lahoz-Monfort  |
Gurutzeta Guillera-Arroita 

School of Biosciences, University of Melbourne, Parkville, Victoria, Australia

Correspondence

Roobeh Valavi

Email: rvalavi@student.unimelb.edu.au

Funding information

Australian Government Research Training Program Scholarship; Rowden White Scholarship; Australian Research Council (ARC) Discovery Early Career Researcher Award, Grant/Award Number: DE160100904; ARC's Centre of Excellence for Environmental Decisions, Grant/Award Number: CE11001000104

Handling Editor: David Warton

Abstract

1. When applied to structured data, conventional random cross-validation techniques can lead to underestimation of prediction error, and may result in inappropriate model selection.
2. We present the R package BLOCKCV, a new toolbox for cross-validation of species distribution modelling. Although it has been developed with species distribution modelling in mind, it can be used for any spatial modelling.
3. The package can generate spatially or environmentally separated folds. It includes tools to measure spatial autocorrelation ranges in candidate covariates, providing the user with insights into the spatial structure in these data. It also offers interactive graphical capabilities for creating spatial blocks and exploring data folds.
4. Package BLOCKCV enables modellers to more easily implement a range of evaluation approaches. It will help the modelling community learn more about the impacts of evaluation approaches on our understanding of predictive performance of species distribution models.

KEYWORDS

block cross-validation, environmental blocking, model evaluation, spatial autocorrelation, spatial blocking, spatial leave-one-out, species distribution modelling, structured environment

1 | INTRODUCTION

Species distribution modelling (SDM) is a popular tool in Ecology, in part because it is able to produce spatial predictions of species distributions. An important component of the modelling process is evaluation of the output (Guisan, Thuiller, & Zimmermann, 2017). Is it fit for purpose? Are some models more suited to the nominated end-use than others (Guillera-Arroita et al., 2015)? Here we introduce a software package relevant to these questions.

When evaluating SDM performance, it is common to test how well predictions match observations at a set of locations (Franklin, 2010; Guisan et al., 2017; Peterson et al., 2011). Whilst early applications of SDM tended to focus on statistical measures of model fit on the data used to train the model, attention has gradually shifted towards testing on independent

data (Elith & Leathwick, 2009). Since fully independent data (such as new surveys) are rarely available, a common approach involves sub-sampling the data available for modelling. In ecology, this usually involves splitting the data into subsets for training and testing (also known as calibration and validation, respectively) (Franklin, 2010; Radosavljevic & Anderson, 2014). Training data are used for fitting the model and testing data for evaluating the performance. This is termed model validation (Hastie, Tibshirani, & Friedman, 2009), with variants based on splitting and testing strategies. For instance, one or more splits can be made, and testing can be done once or repeated in some way (e.g. *k*-fold cross-validation) (Box 1).

In recent years, some discussion has focussed on the best way to allocate data to training and testing datasets (e.g. Bahn & McGill, 2012; Radosavljevic & Anderson, 2014; Wenger & Olden, 2012). Should allocation be random, or organised in some way (hereafter

“non-random”)? This cannot be answered without knowledge of the data and the modelling task. Ecological data are often autocorrelated—i.e. observations close to each other (in space or time) are more similar than distant ones (Legendre, 1993). In species distribution modelling this is true of the response (the species data) and the predictor variables, and might result from biotic or abiotic processes. Spatially-separated training and testing datasets can help test whether the model performs as well in nearby locations as it does in more distant places (Telford & Birks, 2009). If it does not, structure might not be properly accounted for in the model or the model might be over-fitted to the training data (Dormann et al., 2007; Roberts et al., 2017). Several applications of SDMs require extrapolation to new times, and potentially new environments (Elith & Leathwick, 2009). Environmentally -separated training and testing sets allow evaluation of whether models can extrapolate usefully (Roberts et al., 2017). The package we introduce here, written in the R programming language (R Development Core Team, 2017) provides useful tools for allocating data, in non-random ways, to training and testing folds.

2 | THE BLOCKCV R PACKAGE: OVERVIEW

Other software packages do exist for creating non-random datasets for cross-validation—for instance, R packages SPERRORREST (Brenning, 2012) and ENMEVAL (Muscarella et al., 2014) and the Python-based ArcGIS SDMTOOLBOX (Brown, Bennett, & French, 2017). These have various relevant features, but as users of SDMs we found them limited in their applicability to distribution modelling across typical nuances of species data and analytical aims. Specifically, there were limited options for making splits, for allocating data to training and testing datasets, for dealing with both presence-absence and presence-background datasets, and for finding solutions where species are patchily distributed. Package BLOCKCV aims to fill that gap.

In a nutshell, package BLOCKCV provides functions to build training and testing datasets using three general strategies, described in detail below: *spatial blocks*, *buffers*, and *environmental blocks*. It offers several options to construct those blocks and to allocate them to cross-validation (cv) folds (see definitions, Box 1). It includes a function that applies geostatistical techniques to investigate the level of spatial autocorrelation in the chosen predictor variables, to inform the choice of block and buffer size. In addition, visualization tools further aid selection of block size and provide understanding of the spread of species data across generated folds. The package has been written with *species distribution modelling* in mind, though we have kept the output general enough that it is likely to be useful more generally. The functions allow for a number of common usage scenarios, including presence-absence and presence-background species data (Box 1), rare and common species, and raster data for predictor variables.

The generated output is stored in lists and vectors that identify allocation of locations to training or testing data and, where relevant, to cross-validation folds. These can be directly input to any species

BOX 1 Cross-validation, blocks, folds and species data – definitions

Cross-validation and folds

Cross-validation (cv) is a technique for evaluating predictive models. It partitions the data into k parts (*folds*), using one part for testing and the remaining ($k - 1$ folds) for model fitting. In k -fold cv the process is iterated until all the folds have been used for testing. If k is equal to the number of records, it is called n -fold, or leave-one-out cross-validation (Hastie et al., 2009, p. 241). When the folds are not randomly chosen and some sort of environmental, temporal or spatial strategy is used to construct the folds, it is called block cross-validation (Roberts et al., 2017).

Blocks and folds

Broadly speaking, *blocks* are groupings with similar characteristics. In our package they are units of geographical area (e.g. rectangles, spatial polygons and buffers of specific distance) or environmental clusters. Within these units, all species data are treated together—for instance, allocated to the same fold of cv. Several blocks could be allocated to one cv fold.

Species data

Presence-only data are locations where a species was observed. These can be coupled with a large sample of points across the landscape, known as background data (Renner et al., 2015). Presence-absence data include both the locations where the species has been observed and where it has been searched for, but not found (Guillera-Arroita et al., 2015). Following common convention, our code expects presence data to be represented by a 1 and absence or background by a 0.

modelling workflow in R, and formats for the widely-used species distribution modelling package BIOMOD2 (Thuiller, Georges, Engler, & Bréner, 2017) are also included. With the package, we provide a vignette and example data to demonstrate use of these functions in modelling. The following sections describe the functionalities of the package in more detail.

3 | INITIAL CONSIDERATIONS

Conceptually each block is a unit, within which all species data are treated in the same way. Several blocks could be allocated to one cv fold, so decisions need to be made about how to group the blocks into folds for cross-validation (Box 1). The practical implementation of this “blocks into folds” step is affected by the species data, as

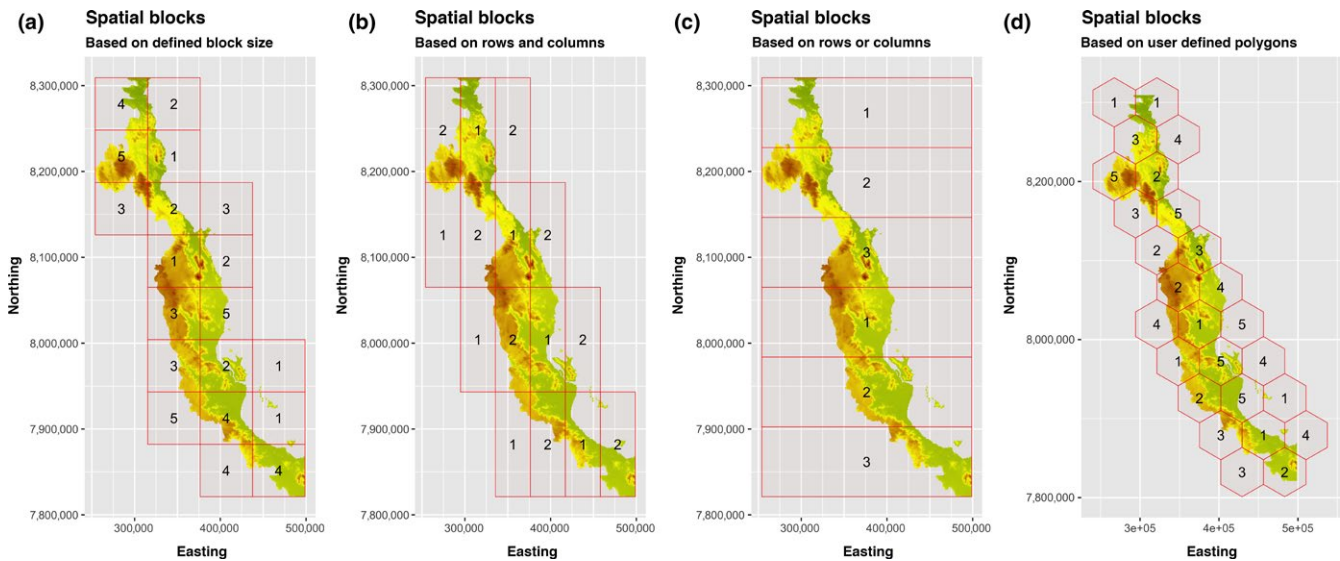


FIGURE 1 Illustration of different spatial blocking strategies. In all panels (a–d), blocks are outlined in red. The numbers in the blocks are fold numbers, showing allocation of blocks to folds

described in the following sections. The package currently explicitly deals with the two main types of species response data (Box 1): presence-only (with background points, where relevant) and presence-absence data. However all blocking strategies can be applied, with this package, to a broad range of data types, as discussed later in relation to the “species” argument.

The package assumes that species spatial data and predictor variables (rasters) have the same projection and similar extent—i.e. the predictors at least extend across all species data. By default, the package creates blocks according to the extent and shape of the study area, based on the predictors. Alternatively, the spatial blocks can be created based on the extent of the species spatial data. This is especially useful when the species data are not evenly dispersed across the whole region e.g. for rare species.

Spatial data sometimes have coordinate reference systems in degrees. Whilst the most satisfactory approach is for the user to first convert these to projected reference systems (e.g. UTM), the package provides alternatives that handle data in degrees. For instance, those blocking strategies relying on distance (size of the blocks or buffers) use an (optional) scaling factor to convert degree to metres. Computations (described later) regarding spatial autocorrelation in predictor raster data with geographic reference system use the great circle distance (the shortest distance between two points on the surface of a sphere; Longley, Goodchild, Maguire, & Rhind, 2015, p. 305).

4 | BLOCKING STRATEGIES

BLOCKCV supports three strategies to separate training and testing data: spatial blocking, spatial buffering, and environmental blocking, each explained in detail below. We first explain the strategies, and then discuss approaches for choosing block or radius size.

4.1 | Spatial blocking

A general strategy to account for spatial autocorrelation when evaluating models is to split data spatially into blocks. Package BLOCKCV provides several methods to create spatial blocks (Figure 1). One is to build square spatial blocks of a specified size (i.e. width) (Figure 1a). The spatial position of these blocks can be shifted vertically or horizontally, allowing assessment of the sensitivity of model evaluation metrics to specific block arrangements. The package also allows division of the study area into vertical or horizontal bins of a given height or width (rectangle blocks, Figure 1b,c), as used by Wenger and Olden (2012) and Bahn and McGill (2012) respectively. Finally, the blocks can be specified by a user-defined spatial polygon layer (Figure 1d).

The package allows allocation of blocks to folds (Box 1) randomly, systematically or in a checkerboard pattern. Blocks-to-folds is one of the key steps for species modelling because species data are rarely evenly dispersed over landscapes. When random selection of folds is chosen, constraints can be set to avoid folds with little or no presence or (where relevant) absence data. Techniques are also implemented for finding block-to-fold allocations that achieve most even spread of species data across folds (e.g., a similar number of presence and absence records in each fold). In systematic allocation, blocks are numbered and assigned to folds sequentially. The number of folds can be specified by the user in the systematic and random allocations and it can be equal or less than the number of blocks. The checkerboard pattern has only two folds, but is useful for enforcing no adjacent blocks in a fold. As explained later, interactive tools are provided for tabulating and visualising the placement of folds and distribution of species data over folds. We note that in all the spatial blocking scenarios, all data in the test folds (including background points, if relevant) are excluded from the training datasets (e.g., Figure 2).

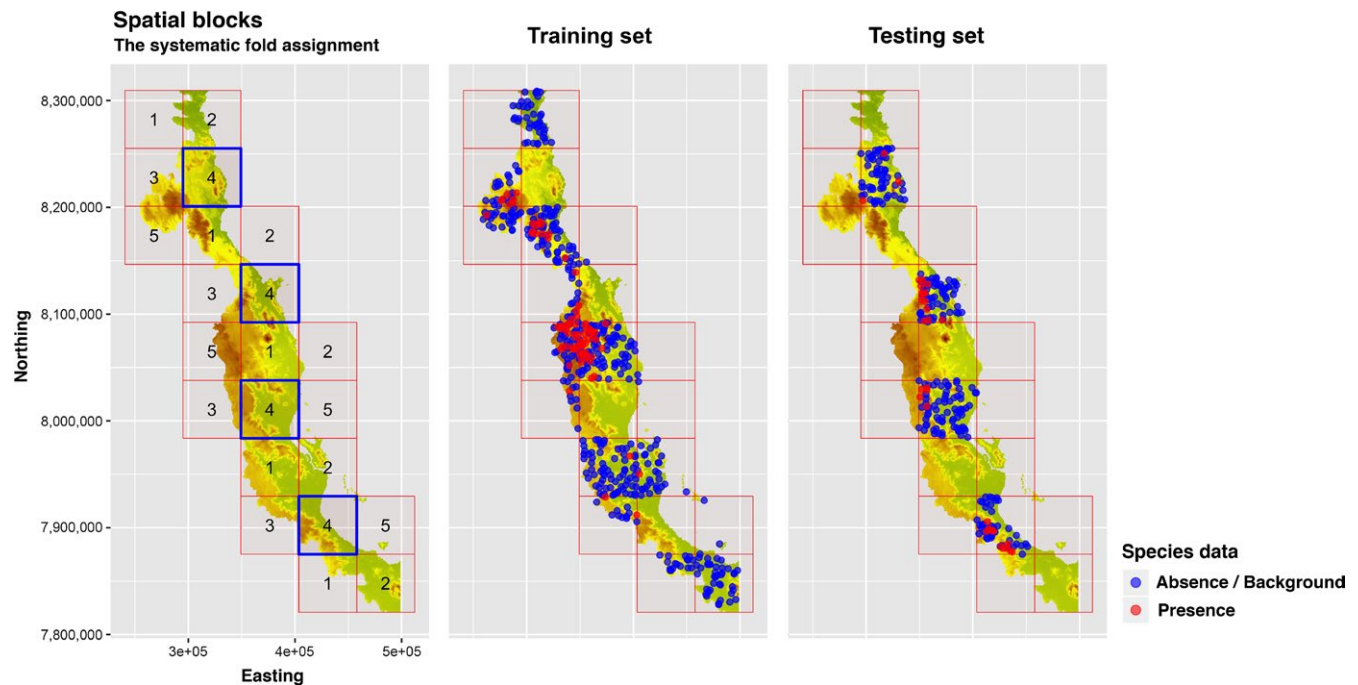


FIGURE 2 Spatial blocks (left) with systematic fold assignment. The middle and right figures show the training and testing sets in fold-4 as an example. As can be seen, no data in the testing blocks are available as training data

4.2 | Buffering

The buffering strategy generates spatially separated training and testing folds by considering circular buffers of specified radius around each observation point (Le Rest, Pinaud, Monestiez, Chadoeuf, & Bretagnolle, 2014). This approach is related to leave-one-out cross-validation (Box 1), and can be desirable if a user wants to ensure that no test data abuts training data. In this case there is no need to distinguish between blocks (buffered points) and folds, because each left-out point is equivalent to a fold.

In the following description, we envisage the method from the viewpoint of the test data. The approach varies slightly with the type of species data available (specified by *spDataType* argument).

For *presence-absence* data, folds are created based on all records, both presences and absences. Each target observation (presence or absence) forms a test point that can be considered the centre of the buffer; all other presence and absence points within the buffer are ignored. The training set comprises all presence and absence points outside the buffer.

When working with *presence-background* data, test folds are determined by the presence data. A buffer is defined around a central target presence point using the specified radius (range). By default, the testing fold comprises the target presence point and all background points within the buffer (Figure 3). Note that the background points within in the buffer are NOT available for training the model. Since some modellers may wish to deal only with presence points, there is an option (*addBG* = FALSE) for working without background data. Any non-target presence points inside the buffer are excluded from both the training and testing sets. All points (presence and

background) outside the buffer are used as the training set. The method cycles through all the presence data, each time allocating one point for testing, so the number of folds is equal to the number of presence points in the dataset.

A “species” argument is used to handle different types of species response data. The default setting (“NULL”) means each row of data (each site) is treated in the same way—this is the relevant setting for presence-only data (with no absence or background points) or other types of response data (e.g. multi-class points for classifying remotely sensed imagery, or a continuous response such as tree height). However, for presence-absence or presence-background data it should be used to direct the code to the column of data containing the 0s and 1s (see Box 1).

```
# example code of buffering with presence-background data
bf2 <- buffering(speciesData= pb_data, # presence-background data
  species= "Species", # columns with 0s and 1s
  theRange= 68000, # radius (m) of the buffer
  spDataType = "PB", # presence-background data type
  addBG = TRUE) # add background data to testing folds
```

4.3 | Environmental blocking

This strategy uses clustering methods (k-means; Hartigan & Wong, 1979) to specify sets of similar environmental conditions based on the input covariates and the chosen number of clusters in (possibly multivariate) environmental space. Species data within each cluster are assigned to a fold, so the number of folds is by default equal

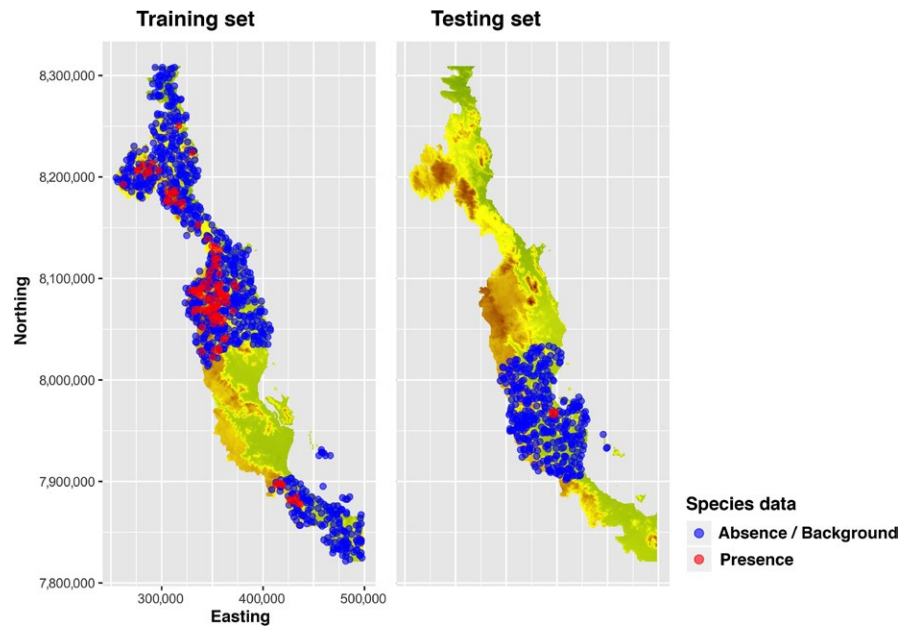


FIGURE 3 The buffering fold with the target point (presence) and background points in the testing fold

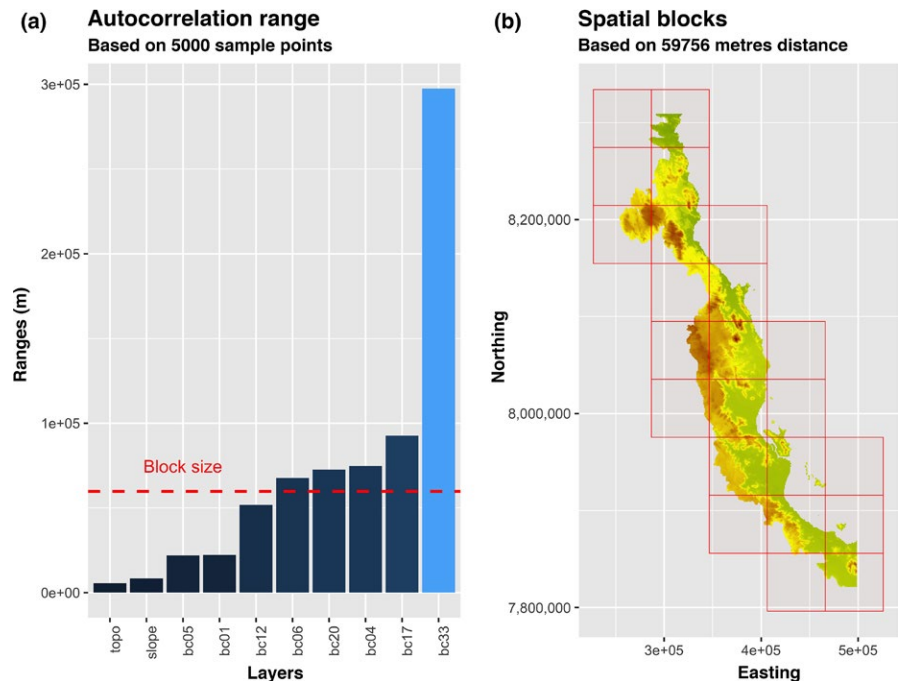


FIGURE 4 Output graph from the package BLOCKCV showing: (a) spatial autocorrelation ranges in input covariates, and (b) corresponding spatial blocks (the selected block size is based on median spatial autocorrelation range across all input data)

to the number of clusters chosen by the user. This algorithm only makes sense with continuous raster data (i.e. categorical covariates like vegetation classes should be excluded).

The clustering can be based on all raster cells or only on values at the species points. When based on the complete rasters, the clusters will be consistent throughout the region and across all species being considered in that region. However, this does not guarantee that all clusters contain species records, especially when species data are not dispersed across all environments. So, the resulting folds in practice might be fewer than the specified k . Alternatively, the clustering can be done based only on the values of the predictors at the species presence and absence (or background) points. In this case, the number

of folds will always be k . Although the clustering is repeated several times to achieve the most homogenous clusters, the resulting clusters might change in different runs. It is therefore worth setting a random seed in R before using this function, to achieve reproducibility.

5 | CHOOSING BLOCK SIZE

One of the challenges of using spatial blocks or buffered cross-validation is choosing the optimal size of blocks or buffers (Trachsel & Telford, 2016). The spatial autocorrelation range in model residuals has been used to define the optimal separation distance

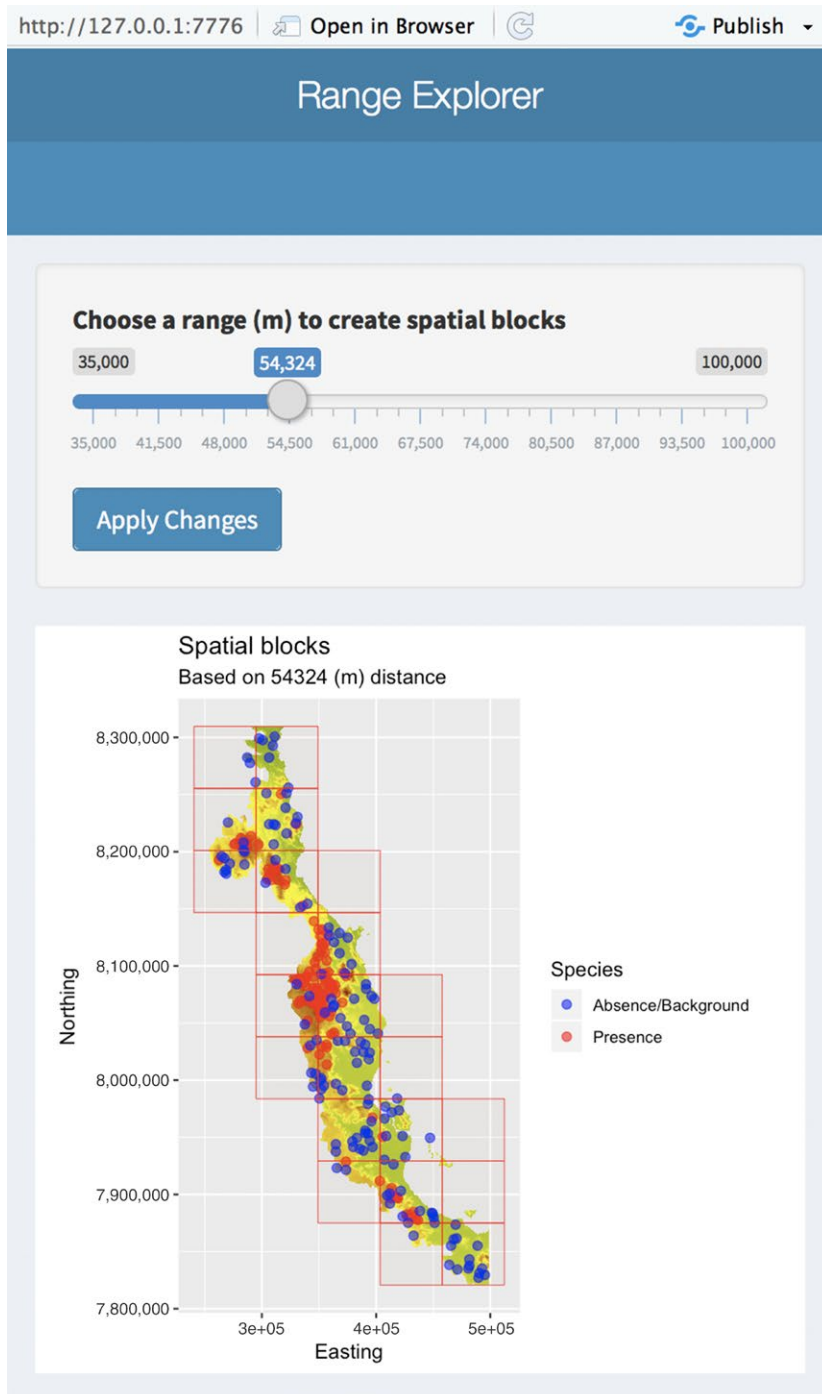


FIGURE 5 The graphical interface for choosing a suitable spatial block size

between training and testing sets (Roberts et al., 2017; Telford & Birks, 2009; Trachsel & Telford, 2016). This is the range over which residuals are approximately independent, and can be characterized using the empirical variogram, a fundamental geostatistical tool for measuring spatial autocorrelation. The empirical variogram describes the structure of spatial autocorrelation by measuring variability between all pairs of points (O'Sullivan & Unwin, 2010), and the range could be used as block size. It is easy to define residuals for presence-absence data (e.g. Bio, De Becker, De Bie, Huybrechts, & Wassen, 2002; Dunn & Smyth, 1996) but less obvious for presence-only data (Baddeley, Turner, Møller, & Hazelton, 2005).

Having the residuals, however, implies having fitted the model already, and this ties assessment to the fitted model. We might instead prefer to apply block CV methods *prior* to model fitting, raising the question of how to define autocorrelation distances and block size. One option suggested for presence-absence data is to fit a variogram to the raw species data (Roberts et al., 2017 and see Bio et al., 2002) and use the resulting distance from the analysis as block size. Alternatively, to support a first choice of block size, prior to any model fitting, package `BLOCKCV` allows the user to look at the existing autocorrelation in the predictors, as an indication of landscape spatial structure (Figure 4). The function works by

automatically fitting variogram models to each continuous raster and finding the effective range of spatial autocorrelation. A number of random points (5,000 by default) is taken from each input raster and parallel processing is used to speed up the computation. For the sake of simplicity, we used isotropic variogram (non-directional) and assumed that the data met the necessary geostatistical criteria e.g. stationarity (having constant variance). The variogram fitting procedure uses the *automap* package (Hiemstra, Pebesma, Twenhöfel, & Heuvelink, 2009). Output plots show the spatial autocorrelation ranges of input raster covariates and the spatial block that has been created based on median of these ranges (Figure 4). After model fitting, the user may want to reassess the appropriateness of the chosen block size by looking at the model residuals.

6 | INTERACTIVE VISUALISATION TOOLS

Package *BLOCKCV* provides two visualisation tools, developed as local web applications using R-package *shiny* (Chang, Cheng, Allaire, Xie, & McPherson, 2017). Through a user interface, these tools (a) allow for graphical exploration of the generated folds (the *foldExplorer* tool) and (b) assist the selection of a suitable spatial block size (the *rangeExplorer* tool). The *foldExplorer* tool displays a map where folds are overlaid, and provides a summary of the number of records in each fold. This helps the user to assess the adequacy of the distribution of training and testing folds throughout the study area. The tool is available for all three blocking strategies; in the case of the environmental blocks, it visualises the blocks in geographic space. The *rangeExplorer* tool (Figure 5) allows the user to interactively change the size of spatial blocks, visualise the blocks and assess the impact of block size on the number and arrangement of blocks in the landscape (and optionally on the distribution of species data in those blocks).

7 | FINAL REMARKS/CONCLUSION

The R package *BLOCKCV* offers a suite of tools for creating data folds via blocking for evaluation of species distribution models, with three different blocking strategies, and tools that help users deal with typical nuances in species data, and for choosing block size and allocating blocks to folds. Which strategy is best to follow in a particular situation will depend on the purpose of the modelling and the region's environmental structure (Roberts et al., 2017). For instance, both environmental blocks and horizontal spatial blocks could create distinct climatic groups in training and testing datasets, which can be useful for assessing models aiming to predict to new climatic conditions.

We recommend that users become familiar with the literature on block cross-validation, some of which we have cited in this article, so they can make appropriate choices. Different choices will have different implications on estimates of predictive performance. For instance, buffering is considered useful for enforcing spatial separation between training and testing folds (Le Rest et al., 2014; Pohjankukka, Pahikkala, Nevalainen, & Heikkonen, 2017), but—depending on the relative sizes of the buffer and

the region—it may produce training sets across repeats more similar than those produced by other blocking strategies. A disadvantage of similar training sets is that the error estimator tends to have high variance—i.e. it is sensitive to the dataset being used to estimate the error (Hastie et al., 2009, p. 242). Buffering also enforces as many training sets as presence or presence-absence points, so it can be quite computationally expensive. Other blocking strategies and choices of how to allocate blocks to folds will also have flow-on effects for estimates of performance, so it is important to think through what is most appropriate.

The package provides example data and a vignette with worked examples so users can explore the functions and learn how to use them in a species modelling workflow. The package will be actively maintained, and new features introduced as needs arise. We hope that this package enables modellers to more easily implement a range of evaluation approaches, so the modelling community learns more about the impacts of evaluation set-up on our understanding of predictive performance of SDMs.

ACKNOWLEDGEMENTS

R.V. is supported by an Australian Government Research Training Program Scholarship and a Rowden White Scholarship; G.G.-A. by an Australian Research Council (ARC) Discovery Early Career Researcher Award (DE160100904), and J.J.L.-M. and J.E. by ARC Discovery Project 160101003. J.E. appreciates the support of the ARC's Centre of Excellence for Environmental Decisions (CE11001000104). We also thank Babak Mirbagheri, Nick Golding, and reviewers and editors: Robert Anderson, David Roberts, an anonymous reviewer and Associate Editor, for their helpful suggestions and advice.

AUTHORS' CONTRIBUTIONS

R.V. conceived the idea, wrote the code, drafted all documentation, and led the writing of the manuscript. All authors contributed to the design and testing of the package, and contributed critically to the manuscript drafts.

DATA ACCESSIBILITY

No data were included in this article. Version 1.0 of the *BLOCKCV* R package is archived at <https://doi.org/10.5281/zenodo.1453551>.

ORCID

Roosbeh Valavi  <https://orcid.org/0000-0003-2495-5277>

Jane Elith  <https://orcid.org/0000-0002-8706-0326>

José J. Lahoz-Monfort  <https://orcid.org/0000-0002-0845-7035>

Gurutzeta Guillera-Aroita  <https://orcid.org/0000-0002-8387-5739>

REFERENCES

Baddeley, A., Turner, R., Møller, J., & Hazelton, M. (2005). Residual analysis for spatial point processes (with discussion). *Journal of the Royal*

- Statistical Society: Series B (Statistical Methodology), 67, 617–666. <https://doi.org/10.1111/j.1467-9868.2005.00519.x>
- Bahn, V., & McGill, B. J. (2012). Testing the predictive performance of distribution models. *Oikos*, 122, 321–331.
- Bio, A. M., De Becker, P., De Bie, E., Huybrechts, W., & Wassen, M. (2002). Prediction of plant species distribution in lowland river valleys in Belgium: Modelling species response to site conditions. *Biodiversity and Conservation*, 11, 2189–2216. <https://doi.org/10.1023/A:1021346712677>
- Brenning, A. (2012). Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: The R package *sperrorest*. In *2012 IEEE international geoscience and remote sensing symposium*, Munich (pp. 5372–5375).
- Brown, J. L., Bennett, J. R., & French, C. M. (2017). SDMtoolbox 2.0: The next generation Python-based GIS toolkit for landscape genetic, biogeographic and species distribution model analyses. *PeerJ*, 5, e4095. <https://doi.org/10.7717/peerj.4095>
- Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2017). *shiny: Web application framework for R*. R package version 1.0.5. Retrieved from <https://CRAN.R-project.org/package=shiny>
- Dormann, C. F., McPherson, J. M., Araújo, M. B., Bivand, R., Bolliger, J., Carl, G., ... Kühn, I. (2007). Methods to account for spatial autocorrelation in the analysis of species distributional data: A review. *Ecography*, 30, 609–628. <https://doi.org/10.1111/j.2007.0906-7590.05171.x>
- Dunn, P. K., & Smyth, G. K. (1996). Randomized quantile residuals. *Journal of Computational and Graphical Statistics*, 5, 236–244.
- Elith, J., & Leathwick, J. R. (2009). Species distribution models: Ecological explanation and prediction across space and time. *Annual Review of Ecology, Evolution, and Systematics*, 40, 677–697. <https://doi.org/10.1146/annurev.ecolsys.110308.120159>
- Franklin, J. (2010). *Mapping species distributions: Spatial inference and prediction*. Cambridge, UK: Cambridge University Press. <https://doi.org/10.1017/CBO9780511810602>
- Guillera-Arroita, G., Lahoz-Monfort, J. J., Elith, J., Gordon, A., Kujala, H., Lentini, P. E., ... Wintle, B. A. (2015). Is my species distribution model fit for purpose? Matching data and models to applications. *Global Ecology and Biogeography*, 24, 276–292. <https://doi.org/10.1111/geb.12268>
- Guisan, A., Thuiller, W., & Zimmermann, N. E. (2017). *Habitat suitability and distribution models: With applications in R*. Cambridge, UK: Cambridge University Press. <https://doi.org/10.1017/9781139028271>
- Hartigan, J. A., & Wong, M. A. (1979). A k-means clustering algorithm. *Journal of the Royal Statistical Society Series C (Applied Statistics)*, 28, 100–108.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed., Vol. 1). New York, NY: Springer series in statistics New York. <https://doi.org/10.1007/978-0-387-84858-7>
- Hiemstra, P. H., Pebesma, E. J., Twenhöfel, C. J., & Heuvelink, G. B. (2009). Real-time automatic interpolation of ambient gamma dose rates from the Dutch radioactivity monitoring network. *Computers & Geosciences*, 35, 1711–1721. <https://doi.org/10.1016/j.cageo.2008.10.011>
- Le Rest, K., Pinaud, D., Monestiez, P., Chadoeuf, J., & Bretagnolle, V. (2014). Spatial leave-one-out cross-validation for variable selection in the presence of spatial autocorrelation. *Global Ecology and Biogeography*, 23, 811–820. <https://doi.org/10.1111/geb.12161>
- Legendre, P. (1993). Spatial autocorrelation: Trouble or new paradigm? *Ecology*, 74, 1659–1673. <https://doi.org/10.2307/1939924>
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic information science and systems*. Hoboken, NJ: John Wiley & Sons.
- Muscarella, R., Galante, P. J., Soley-Guardia, M., Boria, R. A., Kass, J. M., Uriarte, M., & Anderson, R. P. (2014). ENMeval: An R package for conducting spatially independent evaluations and estimating optimal model complexity for Maxent ecological niche models. *Methods in Ecology and Evolution*, 5, 1198–1205. <https://doi.org/10.1111/2041-210X.12261>
- O'Sullivan, D., & Unwin, D. J. (2010). *Geographic information analysis*, 2nd ed. Hoboken, NJ: John Wiley & Sons. <https://doi.org/10.1002/9780470549094>
- Peterson, A. T., Soberón, J., Pearson, R. G., Anderson, R. P., Martínez-Meyer, E., Nakamura, M., & Araújo, M. B. (2011). *Ecological niches and geographic distributions* (Vol. 56). Princeton, NJ: Princeton University Press.
- Pohjankukka, J., Pahikkala, T., Nevalainen, P., & Heikkonen, J. (2017). Estimating the prediction performance of spatial models via spatial k-fold cross validation. *International Journal of Geographical Information Science*, 31, 2001–2019. <https://doi.org/10.1080/13658816.2017.1346255>
- R Development Core Team. (2017). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing.
- Radosavljevic, A., & Anderson, R. P. (2014). Making better Maxent models of species distributions: Complexity, overfitting and evaluation. *Journal of Biogeography*, 41, 629–643. <https://doi.org/10.1111/jbi.12227>
- Renner, I. W., Elith, J., Baddeley, A., Fithian, W., Hastie, T., Phillips, S. J., & Warton, D. I. (2015). Point process models for presence-only analysis. *Methods in Ecology and Evolution*, 6, 366–379. <https://doi.org/10.1111/2041-210X.12352>
- Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Arroita, G., & Warton, D. I. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40, 913–929. <https://doi.org/10.1111/ecog.02881>
- Telford, R. J., & Birks, H. J. B. (2009). Evaluation of transfer functions in spatially structured environments. *Quaternary Science Reviews*, 28, 1309–1316. <https://doi.org/10.1016/j.quascirev.2008.12.020>
- Thuiller, W., Georges, D., Engler, R., & Breiner, F. (2017). *biomod2: Ensemble platform for species distribution modeling*. R package version 3.3-7. Retrieved from <https://CRAN.R-project.org/package=biomod2>
- Trachsel, M., & Telford, R. J. (2016). Estimating unbiased transfer-function performances in spatially structured environments. *Climate of the Past*, 12, 1215–1223. <https://doi.org/10.5194/cp-12-1215-2016>
- Wenger, S. J., & Olden, J. D. (2012). Assessing transferability of ecological models: An underappreciated aspect of statistical validation. *Methods in Ecology and Evolution*, 3, 260–267. <https://doi.org/10.1111/j.2041-210X.2011.00170.x>

How to cite this article: Valavi R, Elith J, Lahoz-Monfort JJ, Guillera-Arroita G. BLOCKCV: An R package for generating spatially or environmentally separated folds for k-fold cross-validation of species distribution models. *Methods Ecol Evol*. 2018;00:1–8. <https://doi.org/10.1111/2041-210X.13107>