ECOGRAPHY

Software note

spectre: an R package to estimate spatially-explicit community composition using sparse data

C. E. Simpkins, S. Hanß, M. C. Spangenberg, J. Salecker, M. H. K. Hesselbarth and K. Wiegand

C. E. Simpkins (https://orcid.org/0000-0003-3212-1379) ✉ (simpkinscraig063@gmail.com), S. Hanß, M. C. Spangenberg, J. Salecker, M. H. K. Hesselbarth (https://orcid.org/0000-0003-1125-9918) and K. Wiegand (https://orcid.org/0000-0003-4854-0607), Univ. of Göttingen, Dept of Ecosystem Modelling, Göttingen, Germany. CES also at: Univ. of Auckland, School of Environment, Auckland, New Zealand. MHKH also at: Dept of Ecology and Evolutionary Biology, Univ. of Michigan, Ann Arbor, MI, USA. KW also at: Univ. of Göttingen, Centre of Biodiversity and Sustainable Land Use (CBL), Göttingen, Germany.

Ecography
2022: e06272
doi: 10.1111/ecog.06272
Subject Editor: Thiago F. Rangel
Editor-in-Chief: Miguel Araújo
Accepted 1 September 2022

An understanding of how biodiversity is distributed across space is key to much of ecology and conservation. Many predictive modelling approaches have been developed to estimate the distribution of biodiversity over various spatial scales. Community modelling techniques may offer many benefits over single species modelling. However, techniques capable of estimating precise species makeups of communities are highly data intensive and thus often limited in their applicability. Here we present an R package, `spectre`, which can predict regional community composition at a fine spatial resolution using only sparsely sampled biological data. The package can predict the presences and absences of all species in an area, both known and unknown, at the sample site scale. Underlying the `spectre` package is a min-conflicts optimisation algorithm that predicts species' presences and absences throughout an area using estimates of α-, β- and γ-diversity. We demonstrate the utility of the `spectre` package using a spatially-explicit simulated ecosystem to assess the accuracy of the package's results. `spectre` offers a simple to use tool with which to accurately predict community compositions across varying scales, facilitating further research and knowledge acquisition into this fundamental aspect of ecology.

Keywords: alpha-diversity, beta-diversity, community distribution, gamma-diversity, open-source, presence–absence, sparse data, species richness

Introduction

Understanding how biodiversity is distributed throughout space is central to ecology and conservation biology. It allows us to discern the processes causing that distribution and how best to conserve this biodiversity in the face of ongoing global change (McMahon et al. 2011). While there is an increasing availability of remotely mapped environmental data, such as topography or climatic conditions, biodiversity patterns often are only sparsely sampled. In response, there has been a rapid uptake of predictive

NORDIC SOCIETY OIKOS
www.ecography.org

modelling relating biological survey data to remotely mapped environmental attributes, aimed at predicting the distribution of biodiversity at different scales (Ferrier and Guisan 2006, D'Amen et al. 2017). While single species distribution modelling is one of the most widely undertaken biodiversity modelling practices (Elith and Leathwick 2009), community modelling has also seen rapid growth and is advantageous in many situations. Community modelling combines data from multiple species and produces estimates of the spatial pattern of biodiversity at a community level rather than at the single species level (Ferrier and Guisan 2006). Modelling at the community level may offer many benefits for situations involving high species diversity or undiscovered species (Ovaskainen et al. 2016). In addition, community modelling often makes use of all available data, including that of rare species necessarily excluded from single species distribution modelling (Guisan et al. 1999).

Early community models focused on providing simple measures of biodiversity, such as species richness and turnover (Nieto-Lugilde et al 2018). These simple biodiversity metrics can be quickly produced, requiring only a relatively small number of direct measurements from field sites and widely available remotely-sensed environmental data. These simple metrics do not, however, capture any information of the component species or makeup of the underlying communities. However, having spatially explicit estimates of community composition can be beneficial in answering many ecological questions, such as predicting the wider impacts of land-use change (Landis 2017). Therefore, several techniques capable of estimating the site-specific composition of communities have been developed.

The majority of these community composition models adopt a predict first, assemble later approach (Ferrier and Guisan 2006), such as the stacked species distribution model (SSDM) approach. SSDM predicts the distribution of individual species using niche-based environmental parameters and then stacks them to predict a species assemblage (D'Amen et al. 2017). While various SSDM and other predict-first-assemble-later methods exist they all share a key drawback; a requirement for large amounts of data and system understanding. This drawback is highly limiting in areas where only sparse sampling has occurred or that are extremely species rich, such as much of the tropics. A need exists, therefore, for an approach with the relatively small data requirements of the simpler biodiversity models, but still capable of estimating community species assemblages.

The use of simulated annealing optimisation-based algorithms has been shown capable of producing species-level community compositions using only sparse field data (Mokany et al. 2011). These optimisation algorithms combine measures of community species richness and compositional dissimilarity between sites to generate a series of spatially and species-specific community distributions (Kirkpatrick et al. 1983, Mokany et al. 2011). Correlative macroecological models and models of compositional dissimilarity are two popular approaches that relate community species richness and species turnover between pairs of sites to environmental variables (D'Amen et al. 2017). Both species richness (i.e. α-diversity) and compositional dissimilarity (i.e. β-diversity) can be easily derived using only sparsely sampled data. Mokany et al. (2011) proposed a simulated annealing optimisation algorithm, `DynamicFOAM`, that generates a target objective matrix against which to test potential community compositions, using estimates of α-diversity and β-diversity. This simulated annealing optimisation approach has been shown to generate useful estimates with a high degree of accuracy in a format that is easily shared with decision makers, using only sparse species data and easily accessible environmental data (Mokany et al. 2014). By incorporating estimates of regional gamma diversity this approach also allows for the prediction of the spatial distribution of unknown species, providing a mechanism with which to gain ecological understanding in areas where data may be limited (Mokany et al. 2011).

Several issues have, however, held back the widespread adoption of this community composition optimisation approach, despite its many advantages. Key amongst these issues is the lack of an easily accessible open-source tool with which to apply the algorithm. While some attempts have been made to implement such a tool, most notably the closed-source `DynamicFOAM Spatial` ver. 1.1 (Mokany et al. 2011), these attempts have not been widely adopted by researchers as no such tool has been developed to easily fit the modern ecologists' usual workflow which uses common scripting tools (such as R or Python). `DynamicFOAM Spatial` needs to be manually set up using a graphical user interface and requires users to calculate the α- and β-diversity estimates outside of the software itself, in very particular formats, therefore not allowing the tool to be used within the usual workflows of ecologists using scripting tools. In addition, it is difficult to fit this type of manual graphical user interface tool into automated repeatable experimentation pipelines, especially if inputs need to be generated by entirely separate pieces of software, resulting in an unproductive bottleneck (Aho and Vos 2018). These problems greatly hinder the speed and applicability of this optimisation type analysis while also reducing repeatability, shareability and overall transparency of results and may have caused an underutilization of the `DynamicFOAM` approach (Asendorpf et al. 2013). If this community composition optimisation approach is to be accessed by a wider audience these limitations need to be addressed. In this paper we present our newly developed R package, `spectre` ver. 1.0.2 (SPatially-Explicit Community disTRibution gEnerator), to overcome the above-mentioned issues.

`spectre` is an R package which implements a community composition optimisation algorithm, using a min-conflicts heuristic, in an easy-to-use way, generating fine-grain species-level community distributions. We have chosen to develop this package for R as this has been found to be the most popular coding language among ecologists (Sciaini et al. 2018, Lai et al. 2019). Additionally, R provides access to many well-established systems for biodiversity modelling (e.g. the `gdm` package (Fitzpatrick et al. 2021)) and spatial analysis (e.g. the `raster` (Hijmans

2021) and `sf` packages (Pebesma 2018)). `spectre` is the first implementation of this type of community composition optimisation algorithm to be embedded in a reproducible open-source framework. This framework allows researchers to estimate all the required inputs (α-, β- and γ-diversities) from a small set of sampled sites, run the optimisation algorithm and analyse the results for the entire landscape all within a single programming environment. This not only allows for an easier and simpler workflow, but also enhances the reproducibility and transparency of the workflow, thereby following calls for such open frameworks in the literature (Stodden et al. 2013, Etherington et al. 2019). The outputs generated by `spectre` provide estimates of the fine-grained distribution of all species (known and unknown) in an entire area, using only limited data collected from a small number of sites allowing this information to be estimated more quickly and cost effectively than other approaches such as SSDMs. Additionally, if labelled species data is provided, labelled species-specific presence and absence estimates can be generated for the entire area. By increasing the ease at which reliable estimates of community compositions can be made, `spectre` may allow for an increase in the usage of such estimates in both applied and theoretical studies, especially in areas with more limited resources, supporting further development in the field of spatial community ecology.

Underlying algorithm

Overview

`spectre` allows users to generate species presence and absence estimates for every site in a landscape using metrics derived from data collected in only a small sample of sites. The algorithm used within `spectre` uses a min-conflicts heuristic to optimise for β-diversity while keeping α-diversity and γ-diversity (i.e. the total number of species in the landscape) as constant constraints (Fig. 1) (Minton et al. 1992, Stuart and Peter 2016). The min-conflicts heuristic implements a set of constraints in a local search domain with which to search for minimum conflicts compared to the objective of the optimisation. This min-conflicts approach has been shown to be far quicker and more efficient than the classic local search approach applied in `DynamicFOAM` (Sosic and Gu 1994). Estimates of α-diversity, β-diversity and γ-diversity must be calculated prior to use within `spectre`. The β-diversity metric used in `spectre` is a pairwise Bray–Curtis dissimilarity matrix (equivalent to a Sorensen dissimilarity matrix if presence and absence data is used) for each site-by-site pair (Ricotta and Podani 2017). For a complete discussion and working guide to predicting β-diversity between site pairs using a generalised dissimilarity modelling see Mokany et al. (2022). The objective commonness matrix describing the number of species in common between each pair of sites (Fig. 1B), which needs to be provided as an input parameter, is calculated using the α- and β-diversity estimates. The goal of `spectre` is to correctly assign species presences and absences for each cell in a landscape so that a commonness matrix generated from the final solution matches the objective commonness matrix. The capability of `spectre` to use simple measures of species richness and community diversity to produce a species-level estimate of community composition is a major benefit of the underlying algorithm.

In the first step of the algorithm `spectre` creates a random presence/absence by site matrix to act as the initial candidate solution (Fig. 1C). Each site in this matrix contains a total number of species records (either present or absent) matching the estimated γ-diversity with the number of presences in a site matching that site's α-diversity estimate. For each iteration of the `spectre` algorithm, the number of shared species between each pair of sites is calculated to create a candidate solution commonness matrix (hereafter candidate commonness matrix; Fig. 1D) with the same structure as the objective commonness matrix. The candidate commonness matrix's difference with the objective commonness matrix is the optimisation algorithm's objective function. `spectre` uses the absolute distance to measure this difference between the two matrices.

At the beginning of each optimisation step, the algorithm removes a random species from a random site (Fig. 1). The algorithm then systematically and individually adds all the absent species at this site to determine the degree of change in the difference between the objective commonness matrix and the resulting candidate commonness matrix. Each added species is removed before a new species is added and tested. When all absent species have been added, tested and removed again, a species is randomly selected from the set of species that resulted in the smallest difference between the objective commonness matrix and the candidate commonness matrix and added to the site. The optimisation is repeated until the algorithm finds a perfect solution, or the maximum number of iterations is reached.

Input data

Three initial inputs are required by `spectre`. 1) A list of the estimated number of species present (α-diversity) in each site. Typically, estimates of α-diversity are created using correlative modelling (e.g. generalised linear models) to link species numbers with environmental variables (Fig. 1). 2) A pairwise (site-by-site) matrix of the predicted compositional dissimilarity between all sites (β-diversity). The β-diversity matrix can be input as either a square matrix or an ordered list. β-diversity needs to be measured as Bray–Curtis dissimilarity (Eq. 1):

$$\beta_{ij} = 1 - 2C_{ij} / \left(\alpha_i + \alpha_j\right) \quad (1)$$

where β_{ij} is the dissimilarity between sites i and j, C_{ij} is the number of species in common between the two sites and α is the number of species in each site (Fig. 1). We recommend

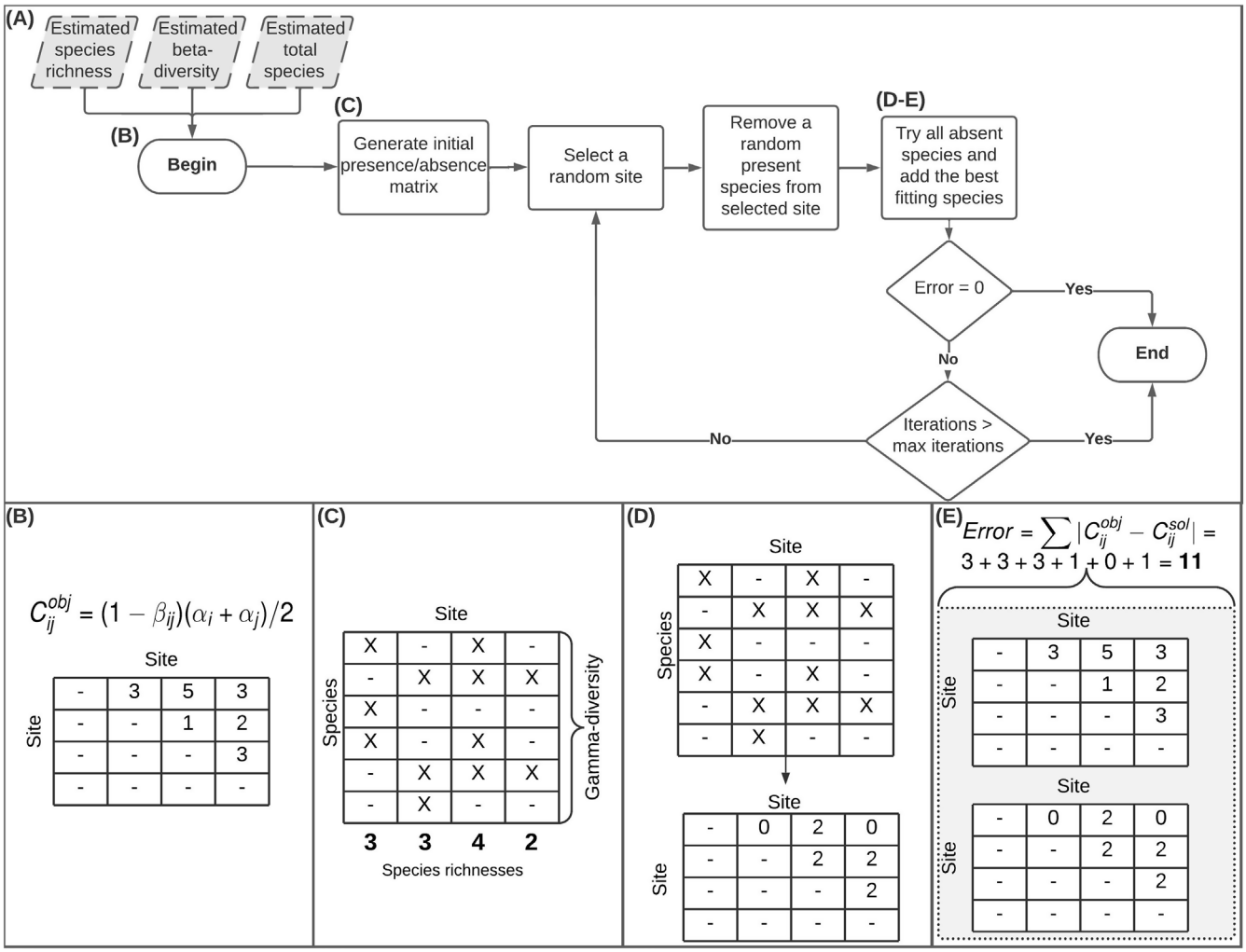

Figure 1. Steps undertaken during a single progression of the `spectre` optimisation algorithm. (A) Flowchart overview of the steps and processes involved. In this panel, inputs are displayed as grey parallelograms, start and end points as ovals, processes as boxes and decision choices as diamonds. (B) The creation of an objective commonness matrix, describing species in common between each pair of sites and created based on the species richness (i.e. α-diversity) and compositional diversity (i.e. β-diversity) estimates used as inputs. This step is the entry point into the algorithm and is only undertaken once per function call, with the objective commonness matrix acting as a target against which all candidate solutions are tested. (C) An initial candidate presence/absence matrix is generated, with the number of presences in each site matching that site's estimated species richness and the total number of species (rows) matching the overall estimated total number of species within the study area (i.e. the γ-diversity). This initial candidate presence/absence matrix can either be created entirely randomly or using known species presences. (D) For every combination of altered species presence, a candidate commonness matrix matching the format of the objective commonness matrix is created. (E) The absolute error between each candidate commonness matrix and the objective commonness matrix is calculated and the candidate with the lowest error selected. If the error equals zero or the number of total iterations has reached the user selected maximum number of iterations the corresponding presence/absence matrix is selected as the final output, otherwise a new random site is selected and the algorithm continues at (D).

estimating pairwise dissimilarity using a generalised dissimilarity model (Ferrier et al. 2007), such as is implemented by the `'gdm'` package in R (Fitzpatrick et al. 2021), though other approaches to calculate Bray–Curtis dissimilarity can also be used. These measures of α- and β-diversity are used to create a pairwise matrix containing the number of species in common between each pair of sites (Fig. 1), derived using (Eq. 2):

$$C_{ij}^{\text{obj}} = \left(1 - \beta_{ij}\right)\left(\alpha_i + \alpha_j\right) / 2 \quad (2)$$

This acts as the objective commonness matrix (C_{ij}^{obj}) against which all candidate commonness matrices are tested (Fig. 1). The last input required is (3) an estimate of the total number of species within the study area (i.e. γ-diversity), provided as a single numeric value which is used to bound the maximum number of species that may be present in any one site. The γ-diversity may be estimated using the measured number of species, prior knowledge or using sample extrapolation (Gotelli and Colwell 2001, Guillera-Arroita et al. 2019). Example data showing formatting is stored in the

'minimal_example_data' dataset provided within the package.

Initial candidate presence/absence matrix

Solutions in the optimisation algorithm are species presence/absence matrices. The number of species presences in each site is equal to the estimated α-diversity for that site (cf. Input data above). Sites with known community compositions can be included as lists of species presences and absences, with the total number of species matching the estimated γ-diversity for the study area. The inclusion of known data can increase the speed at which an accurate solution is reached. The initial candidate presence/absence matrix for use in the algorithm can also be generated completely randomly, selecting several species matching estimated α-diversity to be in site *i*, with all other species absent in this site.

spectre can have two additional optional inputs: a partial solution, in the form of a presence/absence matrix, which is used to generate a candidate solution presence/absence matrix (hereafter candidate presence/absence matrix) that replaces the randomly generated initial candidate presence/absence matrix. Additionally, a matrix indicating cells in the partial solution that can be considered fixed and thereby excluded from the optimisation algorithm. In this way, certain species can be excluded from certain sites or their occurrence in certain sites can be ensured if ecologically meaningful.

Solution optimisation

To improve the initial candidate solution and reach a more optimal estimate of community composition, a series of steps are cyclically repeated (Fig. 1D–E). One complete loop of these cyclical steps is referred to as one iteration. First, a random site is selected and a random presence value in that site switched to absent. Then, every absence value in the selected site is individually switched to presence, and the number of species in common for each new community estimate compared with the objective number of species in common between site pairs (C_{ij}^{obj}). This is done by calculating the number of species in common between site pairs in the candidate presence/absence matrix (C_{ij}^{sol}) using Eq. 3:

$$C_{ij}^{\text{sol}} = \sum O_{si} \times O_{sj} \tag{3}$$

where O_{si} and O_{sj} are the observed presence (represented as one) or absence (represented as zero) of species *s* at site *i* and *j* respectively. Thereby generating the candidate commonness matrix, a site-by-site matrix of the number of species in common between site pairs in the candidate presence/absence matrix, with the same format as the objective commonness matrix. The distance between the objective commonness matrix and each new candidate commonness matrix (hereafter referred to as error) is then calculated as (Eq. 4):

$$error = \sum \left| C_{ij}^{\text{obj}} - C_{ij}^{\text{sol}} \right| \tag{4}$$

The goal of the optimisation is to minimise *error*, with an optimal solution having a value of zero. The *error* for each switched absence value (permutation) is recorded and the permutation with the smallest *error* is retained as the new candidate presence/absence matrix if it is smaller than or equal to the best solution so far. If multiple permutations result in the same minimum *error*, a random one of these permutations is selected. Then the iteration continues with a new random site and species. The optimisation algorithm runs until either the *error* is zero, thus an optimal solution presence/absence matrix was found, or until a pre-selected maximum number of iterations have been run. We recommend that the optimisation algorithm is run with multiple initial candidate presence/absence matrices to ensure the optimisation algorithm can find the global minimum, rather than a local minimum resulting from a sub-optimal starting point.

Example application

To systematically demonstrate the effectiveness and accuracy of the spectre package we tested it using simulated community composition data sets, where known data was sampled to mimic empirical data (Zurell et al. 2010). By using this virtual species approach, we were able to directly compare estimates from spectre to the known complete virtual community datasets, while simultaneously removing any empirically generated sources of uncertainty. The simulated community datasets were built using the virtualspecies ver. 1.5.1 R package (Leroy et al. 2016), which generates spatially-explicit presence/absence matrices from habitat suitability maps. We simulated these suitability maps using Gaussian fields neutral landscapes produced using the NLMR ver. 1.0 R package (Sciaini et al. 2018). To allow for some level of overlap between species suitability maps, we divided the γ-diversity (i.e. the total number of simulated species) by an adjustable correlation value to create several species groups that share suitability maps. Using a full factorial design, we developed 81 presence/absence maps varying across four axes (Supporting information): 1) landscape size, representing the number of sites in the simulated landscape; 2) γ-diversity; 3) the level of correlation among species suitability maps, with greater correlations resulting in fewer shared species groups among suitability maps; and 4) the habitat suitability threshold of the virtual species distribution function. The latter corresponds to the level to which a species is a generalist or a specialist represented by the degree a species distribution can be outside its preferred habitat type from a suitability map. Every variable set in the factorial design was replicated three times. Species richness, pairwise dissimilarity and γ-diversity measures (used as the inputs for the spectre algorithm) were taken directly from the simulated community composition maps, thus avoiding any errors produced in the process of estimating these values. To assess the accuracy of the spectre generated estimates we calculated the relative commonness error (RCE) for each estimate, with the RCE calculated as (Eq. 5):

$$RCE = \frac{\overline{|C_{ij}^{obj} - C_{ij}^{sol}|}}{\overline{C_{ij}^{obj}}} \times 100 \quad (5)$$

where C_{ij}^{obj} is the objective commonness matrix and C_{ij}^{sol} is the candidate commonness matrix.

`spectre` is built around one primary function, *run_optimisation_min_conf()*. Wrapping all of `spectre`'s key functionality and steps into one function allows for simple parallelization of the algorithm using any of the various workflows used within the R ecosystem. Once all input data is prepared, the `spectre` optimisation algorithm is run using a single function call:

```
species_grid <- run_optimization_min_conf(alpha_list=alpha_estimate,
                                          total_gamma=gamma_estimate,
                                          target=beta_estimate,
                                          max_iterations=100000)
```

In this function, the first three arguments enter the three pieces of required input data, which for this demonstration were derived directly from the simulated data sets. `max_iterations` set the maximum number of iterations that the optimisation algorithm may be run for before stopping, set to 100 000 iterations in our example to ensure there were adequate iterations to demonstrate a decrease and stabilisation of overall error. When applied to real use cases the number of iterations can be set to smaller values that still demonstrate the stabilisation of measured error. This stabilisation can be determined by running the function multiple times as is recommended to avoid optimizing to local minima.

After optimisation the function produces an R object containing two pieces of stored data; 1) The predicted species presence and absence records for all sites across the study site, referred to above as solution presence/absence matrix. This prediction of presences and absences is the primary output of the `spectre` package. 2) A record of the decline in $|error|$ produced in the optimisation algorithm, which is useful to visualise the effective improvement to the prediction and can be plotted using the included `plot_error()` function (Supporting information).

All the estimated community compositions derived from the `spectre` algorithm had RCE values less than 20%, though a substantial impact on accuracy was caused by increases in landscape size (Fig. 2, Supporting information). Final solution community compositions most closely matched the known simulated compositions for smaller landscapes with lower numbers of species (though gamma diversity had only a limited effect). Increasing the number of sites (landscape size) had the strongest impact on estimate accuracy while also having the largest impact on

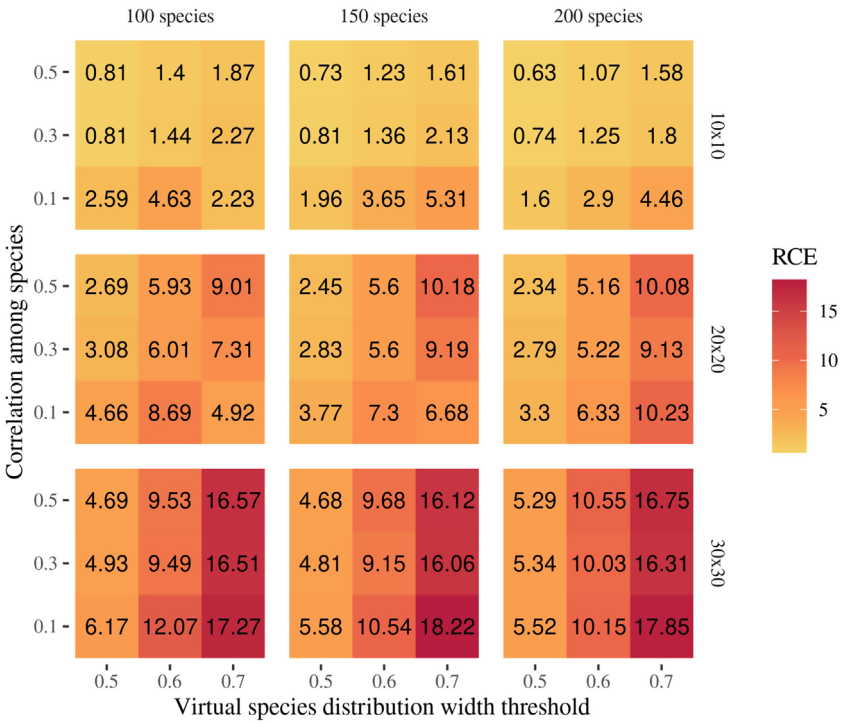

Figure 2. Relative commonness errors (RCE, %) for estimated community compositions predicted by the `spectre` package. Following the virtual species approach, inputs were derived from virtual community compositions with different total numbers of species (columns), landscape sizes (rows), species distribution width thresholds (x-axes) and the strength of correlation among species (y-axes). For each factor combination, the RCE was assessed between the commonness matrices of the community composition predicted by the `spectre` algorithm and the virtual community composition. Each factor combination was replicated three times, printed labels represent mean RCE values across replicates.

computational demand (100 sites and 100 species took 0.5 h to resolve while 900 sites and 200 species took 283 h, using a single core). These results suggest that while community estimates derived using `spectre` may be usefully accurate, this usefulness will degrade for larger landscapes. However, the sizes of landscapes normally studied in landscape ecology (such as 1000 km^2 with 50 m^2 plots or approximately 600 cells) estimates have high accuracy. Though it should be noted that given the drop off in accuracy with increasing landscape sizes `spectre` may not be recommended for continental scale applications with thousands of cells. Additionally, accuracy can be increased by including known community composition data for individual sites if available (see incorporating empirical species composition data for specific sites section below).

This example was run using an AMD EPYC Processor with 2.9 GHz, 32 cores, and 128GB RAM, though as the optimisation process currently cannot be run in parallel, only one core was used per landscape. Future developments will focus on chunking larger landscapes into multiple smaller overlapping sections. Each of these sections could be solved in parallel before rejoining. By adopting this parallel chunking approach over multiple iterations compute times will be reduced as a function of the number of simultaneous instances run.

Incorporating empirical species composition data for specific sites

Including sites with known community compositions allows the `spectre` algorithm to estimate the distribution of the specific species within that community. To illustrate this capability, we used subsamples of the Barro Colorado Island (BCI), Panama open data set (Condit et al. 2019). We created presence and absence lists for 100 tree species from 100 randomly selected sites (referred to as quadrats in Condit et al. 2019), sampled without replacement out of the 1251 sites from the BCI data. One fraction of the sampled data was included in the initial candidate presence/absence matrix, and the remaining fraction was used to test how many species in the remaining sites were correctly predicted in the solution presence/absence matrix generated by `spectre`. We analysed the changes in accuracy in estimating specific species from including between 0 and 40 sites with known community compositions, with a step size of five sites. We ran 25 replicates for each number of known sites. The code used to run this type of analysis is:

```
  species_grid <- run_optimization_min_
conf(alpha_list=alpha_estimate,
          total_gamma=gamma_estimate,
          target=beta_estimate,
          partial_solution=known_spe-
cies,
          fixed_species=known_species,
          max_iterations=100000)
```

where `known_species` is a binary site-by-species matrix with known species represented as a one and unknown species represented as a zero and the total number of species matching `total_gamma`. Note that `known_species` must be supplied as both the `partial_solution` and `fixed_species` parameters.

The proportion of correctly predicted species showed an initial large step followed by a near linear rate of increase as the number of sites with known community compositions increased (Supporting information). Including 25 sites with known community compositions roughly doubled the percentage of correctly predicted species from around 30% (no sites included) to greater than 60%.

Conclusion and outlook

The `spectre` package provides a tool to easily implement an optimisation-based community composition model. Optimisation-based community composition models have been shown capable of accurately predicting the species-level community makeup for large landscapes at fine resolutions (Mokany et al. 2011). These models have, however, not been widely adopted due to difficulties in their implementations. By simplifying the implementation and increasing the algorithms efficiency `spectre` can facilitate a wider adoption of optimisation-based community composition models in ecological research.

The `spectre` package is a continuously developing, open-source tool that has several opportunities for future development that can further broaden its scope. Currently, `spectre` is accurate at scales of hundreds of sites, though for larger landscapes in the order of several thousand sites accuracy greatly reduces, while compute times significant increase. Future work could decrease this reduction in accuracy with landscape size, while simultaneously improving compute times, by allowing landscapes to be subdivided with partial overlaps. These subdivisions could then be run simultaneously with intermediate results being shared between overlapping subdivisions and weighted by the autocorrelation of input variables. In addition, we envisage adding other measures of dissimilarity, beyond only Bray–Curtis dissimilarity, to be used as inputs into the algorithm. Beyond these planned developments as `spectre` is fully open-source, it may be used as a base from which other researchers may extend the package to use-cases not currently foreseen.

To our knowledge `spectre` is the first completely openly available software capable of implementing this type of algorithm for spatially-explicit and species-specific community composition estimation. While other algorithms such as SSDMs can provide similar community composition estimates they have far larger, often prohibitive, data requirements compared to `spectre`, making `spectre` useful in situations where other algorithms cannot be applied. Our package is designed to follow a straightforward workflow and is extensively documented, including a

vignette demonstrating and explaining all the steps necessary to undertake a complete analysis. The package is hosted on CRAN and GitHub and is freely available.

To cite `spectre` or acknowledge its use, cite this Software note as follows, substituting the version of the application that you used for 'version 1.0':

Simpkins, C. E. et al. 2022. `spectre`: an R package to estimate spatially-explicit community composition using sparse data. – Ecography 45: XX–XX (ver. 1.0).

Acknowledgements – We thank Marco Sciaini for R coding advice and Sebastian Fiedler for feedback on earlier versions of the manuscript. Additionally, we would like to thank Marcus Baum, Laura Wolf and Fabian Sigges for their input and expertise in helping guide the design of the spectre algorithm. Lastly, we would like to thank the anonymous reviewers for their contributions in improving our manuscript. Open access funding enabled and organized by Projekt DEAL.

Funding – This study was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project no. 192626868 – SFB 990 in the framework of the collaborative German – Indonesian research project CRC 990. M.C.S. was supported by the German Research Foundation (DFG) through grant no. WI 1816/18-2 (FOR2432/2).

Author contributions

Craig E. Simpkins: Conceptualization (lead); Formal analysis (lead); Methodology (lead); Project administration (lead); Software (lead); Writing – original draft (lead); Writing – review and editing (lead). **Sebastian Hanß**: Formal analysis (supporting); Investigation (supporting); Software (supporting); Writing – review and editing (supporting). **Matthias C. Spangenberg**: Data curation (supporting); Formal analysis (supporting); Methodology (supporting); Software (supporting); Visualization (lead); Writing – review and editing (supporting). **Jan Salecker**: Conceptualization (supporting); Data curation (supporting); Formal analysis (supporting); Methodology (supporting); Software (supporting); Writing – original draft (supporting); Writing – review and editing (supporting). **Maximilian H. K. Hesselbarth**: Data curation (supporting); Formal analysis (supporting); Methodology (supporting); Software (supporting); Visualization (supporting); Writing – review and editing (supporting). **Kerstin Wiegand**: Conceptualization (supporting); Funding acquisition (lead); Investigation (supporting); Methodology (supporting); Project administration (supporting); Resources (lead); Supervision (lead); Writing – review and editing (supporting).

Transparent peer review

The peer review history for this article is available at <https://publons.com/publon/10.1111/ecog.06272>.

Data availability statement

Virtual species data are available from the Dryad Digital Repository: <https://doi.org/10.5061/dryad.fbg79cnz7> (Simpkins et al. 2022). Scripts replicating use case examples and spectre package source code are available from GitHub: <https://github.com/r-spatialecology/spectre_usecase> and <https://github.com/r-spatialecology/spectre>.

Supporting information

The Supporting information associated with this article is available with the online version.

References

Aho, P. and Vos, T. 2018. Challenges in automated testing through graphical user interface. – IEEE Int. Conf. Softw. Testing 2018: 118–121.

Asendorpf, J. B. et al. 2013. Recommendations for increasing replicability in psychology. – Eur. J. Person. 27: 108–119.

Condit, R. et al. 2019. Complete data from the Barro Colorado 50-ha plot: 423 617 trees, 35 years. – Dryad Digital Repository, <https://doi.org/10.15146/5xcp-0d46>.

D'Amen, M. et al. 2015. Using species richness and functional traits predictions to constrain assemblage predictions from stacked species distribution models. – J. Biogeogr. 42: 1255–1266.

D'Amen, M. et al. 2017. Spatial predictions at the community level: from current approaches to future frameworks: methods for community-level spatial predictions. – Biol. Rev. 92: 169–187.

Elith, J. and Leathwick, J. R. 2009. Species distribution models: ecological explanation and prediction across space and time. – Annu. Rev. Ecol. Evol. Syst. 40: 677–697.

Etherington, T. R. et al. 2019. A research institution framework for publishing open code to enable reproducible science. – PeerJ Preprints 7: e27762v1.

Ferrier, S. and Guisan, A. 2006. Spatial modelling of biodiversity at the community level. – J. Appl. Ecol. 43: 393–404.

Ferrier, S. et al. 2007. Using generalized dissimilarity modelling to analyse and predict patterns of beta diversity in regional biodiversity assessment. – Divers. Distrib. 13: 252–264.

Fitzpatrick, M. C. et al. 2021. gdm: generalized dissimilarity modeling. – <https://CRAN.R-project.org/package=gdm>.

Gotelli, N. J. and Colwell, R. K. 2001. Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. – Ecol. Lett. 4: 379–391.

Guillera-Arroita, G. et al. 2019. Inferring species richness using multispecies occupancy modeling: estimation performance and interpretation. – Ecol. Evol. 9: 780–792.

Guisan, A. et al. 1999. GLM versus CCA spatial modeling of plant species distribution. – Plant Ecol. 143: 107–122.

Hijmans, R. J. 2021. raster: geographic data analysis and modeling. – <https://CRAN.R-project.org/package=raster>.

Kirkpatrick, S. et al. 1983. Optimization by simulated annealing. – Science 220: 671–680.

Lai, J. et al. 2019. Evaluating the popularity of R in ecology. – Ecosphere 10: e02567.

Landis, D. A. 2017. Designing agricultural landscapes for biodiversity-based ecosystem services. – Basic Appl. Ecol. 18: 1–12.

Leroy, B. et al. 2016. virtualspecies, an R package to generate virtual species distributions. – Ecography 39: 599–607.

McMahon, S. M. et al. 2011. Improving assessment and modelling of climate change impacts on global terrestrial biodiversity. – Trends Ecol. Evol. 26: 249–259.

Minton, S. et al. 1992. Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems. – Artif. Intellig. 58: 161–205.
Mokany, K. et al. 2011. Combining α and β-diversity models to fill gaps in our knowledge of biodiversity. – Ecol. Lett. 14: 1043–1051.
Mokany, K. et al. 2014. Identifying priority areas for conservation and management in diverse tropical forests. – PLoS One 9: e89084.
Mokany, K. et al. 2022. A working guide to harnessing generalized dissimilarity modelling for biodiversity analysis and conservation assessment. – Global Ecol. Biogeogr. 31: 802–821.
Nieto-Lugilde, D. et al. 2018. Multiresponse algorithms for community-level modelling: review of theory, applications and comparison to species distribution models. – Methods Ecol. Evol. 9: 834–848.
Ovaskainen, O. et al. 2016. Uncovering hidden spatial structure in species communities with spatially explicit joint species distribution models. – Methods Ecol. Evol. 7: 428–436.
Pebesma, E. 2018. Simple features for R: standardized support for spatial vector data. – R J. 10: 439–446.
Ricotta, C. and Podani, J. 2017. On some properties of the Bray–Curtis dissimilarity and their ecological meaning. – Ecol. Complex. 31: 201–205.
Sciaini, M. et al. 2018. NLMR and landscapetools: an integrated environment for simulating and modifying neutral landscape models in R. – Methods Ecol. Evol. 9: 2240–2248.
Simpkins, C. E. et al. 2022. Data from: spectre: an R package to estimate spatially-explicit community composition using sparse data. – Dryad Digital Repository, <https://doi.org/10.5061/dryad.fbg79cnz7>.
Sosic, R. and Gu, J. 1994. Efficient local search with conflict minimization: a case study of the n-queens problem. – IEEE Trans. Knowl. Data Eng. 6: 661–668.
Stewart, G. 2010. Meta-analysis in applied ecology. – Biol. Lett. 6: 78–81.
Stodden, V. and Miguez, S. 2013. Best practices for computational science: software infrastructure and environments for reproducible and extensible research. – J. Open Res. Softw. 2: e21.
Stodden, V. et al. 2013. Toward reproducible computational research: an empirical analysis of data and code policy adoption by journals. – PLoS One 8: e67111.
Stuart, R. and Peter, N. 2016. Artificial intelligence – a modern approach, 3rd edn. – Pearson Education Limited.
Zurell, D. et al. 2010. The virtual ecologist approach: simulating data and observers. – Oikos 119: 622–635.