

Species' range model metadata standards: RMMS

Cory Merow^{1,2}  | Brian S. Maitner³  | Hannah L. Owens^{4,5}  | Jamie M. Kass^{6,7}  |
Brian J. Enquist³  | Walter Jetz²  | Rob Guralnick⁴ 

¹Ecology and Evolutionary Biology, University of Connecticut, Storrs, Connecticut

²Ecology and Evolutionary Biology Department, Yale University, New Haven, Connecticut

³Department of Ecology and Evolutionary Biology, University of Arizona, Tucson, Arizona

⁴Florida Museum of Natural History, University of Florida, Gainesville, Florida

⁵Center for Macroecology, Evolution, and Climate, University of Copenhagen, Copenhagen, Denmark

⁶Department of Biology, City College of New York (CUNY), New York, New York

⁷Program in Biology, The Graduate Center, CUNY, New York, New York

Correspondence

Cory Merow, Ecology and Evolutionary Biology, University of Connecticut, 75 N. Eagleville Road, Unit 3043 Storrs, CT 06269-3043.

Email: cory.merow@gmail.com

Funding information

NSF, Grant/Award Number: DEB1565046, DEB1046328, and DEB1137366

Editor: Antoine Guisan

Abstract

Aim: The geographic range and ecological niche of species are widely used concepts in ecology, evolution and conservation and many modelling approaches have been developed to quantify each. Niche and distribution modelling methods require a litany of design choices; differences among subdisciplines have created communication barriers that increase isolation of scientific advances. As a result, understanding and reproducing the work of others is difficult, if not impossible. It is often challenging to evaluate whether a model has been built appropriately for its intended application or subsequent reuse. Here, we propose a standardized model metadata framework that enables researchers to understand and evaluate modelling decisions while making models fully citable and reproducible. Such reproducibility is critical for both scientific and policy reports, while international standardization enables better comparison between different scenarios and research groups.

Innovation: Range modelling metadata (RMMS) address three challenges: they (a) are designed for convenience to encourage use, (b) accommodate a wide variety of applications, and (c) are extensible to allow the research community to steer them as needed. RMMS are based on a metadata dictionary that specifies a hierarchical structure to catalogue different aspects of the range modelling process. The dictionary balances a constrained, minimalist vocabulary to improve standardization with flexibility for users to modify and extend. To facilitate use, we have developed an R package, *rangeModelMetaData*, to build templates, automatically fill values from common modelling objects, check for inconsistencies with standards, and suggest values.

Main conclusions: Range Modelling Metadata tools foster cross-disciplinary advances in biogeography, conservation and allied disciplines by improving evaluation, model sharing, model searching, comparisons and reproducibility among studies. Our initially proposed standards here are designed to be modified and extended to evolve with research trends and needs.

KEYWORDS

abundance, MAXENT, niche model, R package, reproducible research, species distribution model

1 | INTRODUCTION

Species' geographic ranges and environmental niches are fundamental units of biogeography and among the most widely used summaries in biology (Guisan & Thuiller, 2005; Jetz, McPherson, & Guralnick, 2012). Correlative range models (i.e., species distribution models, environmental niche models, resource selection models) describe how occurrence or abundance varies in environmental and/or geographic space and are applied to biodiversity assessments and forecasts, conservation planning, niche evolution, invasion biology and many other fields (Franklin, 2010; Guisan, Thuiller, & Zimmermann, 2017; Peterson, Soberón, Pearson, & Anderson, 2011). Many modelling approaches have been developed to quantitatively characterize ranges and environmental niches with different goals in each field, and user-friendly software has enabled many thousands of studies. However, differences in approaches and methodologies – some based on different study foci and others on field-specific jargon – have created barriers to communication and led to increasing isolation of scientific advances. For example, wildlife ecology has a literature on resource selection modelling that is rather distinct from environmental niche modelling in plant ecology, in spite of very similar data, concepts and objectives (Warton & Aarts, 2013). Recent calls have been made to standardize range model metadata to enable reuse of models both generally (Borba & Correa, 2015; Costa et al., 2018) and with the specific goal of estimating biodiversity patterns (Araújo et al., 2019), but detailed metadata standards remain lacking. Here, we propose range modelling metadata standards (RMMS) that aim to improve communication, reproducibility and reusability of published models.

1.1 | Why do we need RMMS?

Range modelling is a highly varied field with little consensus and calls for greater standardization and transparency (Joppa et al., 2013). Without standardized metadata that describe range models, it can be difficult to evaluate if a model has been built appropriately for its intended use or if it is suitable for reuse in subsequent studies. A number of studies have outlined clear connections between modelling decisions and resulting inferences (Guillera-Aroita et al., 2015; Guisan et al., 2017; Merow et al., 2014), and advances in biological metadata have already standardized and connected primary biodiversity data (Guralnick, Walls, & Jetz, 2017; Wiczorek et al., 2012). By specifying standards, methodologies will become more immediately transparent for peers as researchers adopt a standard metadata vocabulary. Easy-to-use metadata will considerably simplify the reviewing process by automating the reporting of decisions, which can take considerable time for reviewers and help them better understand the methodological context of a study's insights. Metadata can also help relieve manuscripts from laborious methodological descriptions, increasing valuable space to focus on results.

Range models constitute valuable information products that have been recognized as key for developing an understanding of the status and trends in species distributions. They are vital to large biodiversity modelling projects such as Botanical Information and

Ecology Network (BIEN; biendata.org) and Map of Life (MOL, mol.org) and synthetic conservation efforts such as defining species distribution essential biodiversity variables (Jetz et al., 2019; Pereira et al., 2013). The large taxonomic scale of the range models in these efforts leverages standardized approaches to improve model reliability, but such mass production places an even stronger onus to report how models were produced. The potential inclusion of range models produced by the research community in these databases necessitates metadata that enable comparisons and integration. Making range model products easily citable via searchable metadata increases accessibility to other subdisciplines of biology and environmental science and provides credit for the researchers who developed the models. Standardization also helps connect related subdisciplines that have evolved their own language or best practices but may benefit from cross-pollination. Over time, adherence to metadata standards would support a catalogue where researchers could search for modelling studies based on features of interest (e.g., data sources, model method and settings, reported evaluation metrics) that would otherwise likely be inaccessible from metadata on a published paper. Meta-analyses leveraging this resource might have applications ranging from community ecology to biogeography to methodological development.

Taken together, advancing standardized range model metadata will enable more reproducible, standardized, searchable and citable science. As these standards are meant to grow with the field, they will benefit from engagement and improvements from the user community. After an initial phase of testing and validation, we hope that RMMS can become a completely community-driven enterprise without need for management by a given entity or our research team. These gains in scientific precision and communication are well positioned to outweigh the effort required to report standardized metadata. Furthermore, our efforts will bring range modelling in line with other successful efforts in reproducible research systems (Mesirov, 2010) in other domains in the life sciences (Goecks, Nekrutenko, Taylor, & Team, 2010).

To promote adoption of our proposed metadata standard, we have designed convenient and flexible tools for its implementation, including a user-friendly interface to enable researchers to provide such descriptions with minimal effort and errors. We provide an R package, `rangeModelMetadata`, that automatically completes many required fields and can be extended to automatically fill them from common modelling objects in R.

2 | `rangeModelMetadata (rmm)` FORMAT

The `rangeModelMetadata (rmm)` format that we propose is designed to be human readable to accommodate more flexible specification of inputs, as well as ensure generality beyond specific software or present-day use cases. After sharing a minimum set of critical metadata, provision of additional information is optional. This flexibility gives researchers three advantages: (a) it is adaptable to new technologies (e.g., algorithms, applications), (b) it will ensure

relevance to a broad user base, and (c) it permits customization as needed. The standards are comprehensive enough to provide guidance and clarity, but not onerous.

The basic unit of RMMS is a single study with a single model per taxon to reduce the burden on researchers, in contrast to building a metadata object for each species or model (although this is a custom option). This follows standards from the biosciences standards community to focus on the study or experiment (Taylor et al., 2008). The structure of `rmm` objects correspond to eight top-level fields: `authorship`, `studyObjective`, `data`, `dataPrep` (data preparation), `modelFit`, `prediction`, `evaluation` and `code` (Table 1). Within each of these top-level fields are subfields, which may contain further granular reporting. The named *values* assigned to unique combinations of *fields* (e.g., `data:environment:extent`) are termed “entities” (see a subset of the metadata dictionary in Table 1 and a complete version in Supporting Information Supplement S1). Entities have values that are vectors of characters or numbers.

Our metadata dictionary includes the hierarchical structure of the metadata *entities*, provides standardized and suggested inputs, and defines all the content needed to produce an `rmm` object (Table 1; Supporting Information S1). Each row defines a single *entity* in a `rmm` object, classified by columns specifying the field hierarchy described above. Some *entities* with commonly used settings have a constrained vocabulary to standardize *values* (noted in the *constrainedValues* column of the dictionary), while others may take on any value. To balance flexibility with standardization, many entities are partially constrained such that a standardized vocabulary is available for certain common values while user-defined values are also accepted. To add further flexibility, many fields have a `:Notes` entity (e.g., `data:notes`, `dataPrep:notes`, `modelFit:notes`) to allow authors to mention any additional high-level critical information. Formatted examples as well as descriptions of guidelines for user-defined values are also included in the dictionary. All values can be entered programmatically with our R package `rangeModelMetadata` or manually into a csv file (templates provided in Supporting Information S5 and S6).

3 | STANDARDS

The standards below provide background on the predefined *entities* and guidance on how to extend them to include user-specified options.

3.1 | A case study

As an example for constructing an `rmm` object in the sections that follow, we built a simplified range model for *Bradypus variegatus*, the brown-throated sloth, in South America. Specifically, we use `MAXENT` (Phillips, Anderson, & Schapire, 2006) and `DSMO` (Hijmans, Phillips, Leathwick, & Elith, 2017) applied to occurrence data from the Global Biodiversity Information Facility (GBIF; GBIF.org, 2019) and climate data from Worldclim (Fick & Hijmans, 2017). See Supporting Information S4 for complete workflow. Various modelling decisions

are described below in the context of constructing a metadata object. Notably, we begin with a study involving only a single species and describe how to extend this below in “*Multispecies studies*”. The resulting `rmm` object is shown in Figure 1.

3.2 | Authorship

The `authorship` field provides information on citation, contact information, related studies using the models and licensing/use restrictions associated with the models. Each `rmm` object is given a unique name in the format `Author_Year_Taxa_Model_fw`. We suggest the convention that *Author* be limited to surnames and that multiple authors be included via camel case (e.g., *MerowMaitnerOwensKassEnquistJetzGuralnick*). *Year* should include a four-digit year. *Taxa* can be specified at the authors' discretion and include common or scientific names at any appropriate taxonomic level (e.g., *Sloth*, *Bradypus*, *BradypusVariegatus*). *Model* should describe the algorithm used [multiple models can be specified when using ensemble models (Araújo & New, 2007; Thuiller, Lafourcade, Engler, & Araújo, 2009)] – standardized model names can be viewed in the `modelFit:algorithm` field of the metadata dictionary. Finally, two random alphanumeric characters should be appended to the `rmm` name to prevent cases where ambiguity might arise. A complete example could take the form (Figure 1): *MerowMaitnerOwensKassEnquistJetzGuralnick_2018_BradypusVariegatus_Maxent_b3*.

3.3 | Study objective

Entities under `studyObjective`, including `:purpose`, `:rangeType`, `:invasion`, `:transfer`, and so forth, provide authors with a text field to briefly describe the intended application of their study to set the context for modelling decisions specified in other fields. In our example study, the model it was fit in the northern part of South America, and transferred to the southern part in order to determine whether there is any potentially suitable habitat in a region where no records exist:

```
studyObjective:purpose='transfer'
studyObjective:rangeType='potential'
studyObjective:transfer='detect    unoccupied
suitable habitat'
```

3.4 | Data

Information within the `data` field pertains to occurrence records (`data:occurrence`) and environmental data (`data:environment`) used to train or transfer models. The `:occurrence` field may contain taxon names (`:occurrence:taxaVector`), the type of occurrence data used (`:occurrence:occurrenceDataType`; e.g., presence-only, presence-absence, abundance), the temporal extent of the occurrence records (`:occurrence:yearMin`, `:occurrence:yearMax`), occurrence data sources (`:occurrence:sources`) and information on sample sizes. The `data:environment` field may contain information on the environmental

TABLE 1 An example of entries in the `xmm` metadata dictionary

field1	field2	field3	entity	family	examples
authorship			rmmName	base	MerowMaitnerOwensKassEnquistGuralnick_2018_Acer_Maxent_b3
authorship			license	base	CC; CC BY; CC BY-SA; CC BY-ND; CC BY-NC; CC BY-NC-SA; CC BY-NC-ND
studyObjective			purpose	base	projection
studyObjective			rangeType	base	potential; realized
studyObjective			invasion	base	native; invasive; naturalized; colonized
data	occurrence		taxon	base	Acer rubrum; Nasua nasua;
data	occurrence		dataType	base	presence only; presence-absence; abundance
data	occurrence		occurrenceType	NULL	breeding; wintering; migratory; resident
data	occurrence		yearMin	base	1900
data	occurrence		yearMax	base	2000
data	occurrence		presenceSampleSize	occurrence, po, pa	87
data	occurrence		backgroundSampleSizeSet	occurrence, po	1000
data	environment		variableNames	base	bio1, bio4, bio12, bio15
data	environment		minVal	transferEnv2	[{"bio1":273,"bio12":0,"bio15":0,"bio2":9}]
data	environment		maxVal	transferEnv2	[{"bio1":292,"bio12":10017,"bio15":246,"bio2":212}]
data	environment		yearMin	base	1970
data	environment		yearMax	base	2000
data	environment		extentSet	base	[{"xmin":-5582581.9734,"xmax":5367418.0266,"ymin":-7389365.067,"ymax":7410634.933}]
modelFit			algorithm	base	generalized linear model; generalized additive model; boosted regression trees; maxent; biotdim; Poisson point process; range bagging; all biomod2 models;
modelFit	maxent		featureSet	maxent	L; LQ; LQP; LQPT; LQPTH; H; HT
modelFit	maxent		featureRule	maxent	L for \$<\$50 presences; LQ for \$>\$= 50 presences
modelFit	maxent		regularizationMultiplierSet	maxent	1, 1.5, 2, 2.5, 3
modelFit	maxent		regularizationRule	maxent	Chosen based on fivefold cross-validation on a grid of regularization multipliers from [0;10]
modelFit	maxent		convergenceThresholdSet	maxent	1.00E-05
modelFit	maxent		samplingBiasRule	maxent	target group; offset; none;
modelFit	maxent		samplingBiasNotes	maxent	NULL
prediction	transfer	environment1	units	transferEnv1	absolute probability; relative occurrence rate; presence/absence
prediction	transfer	environment1	minVal	transferEnv1	0.001
prediction	transfer	environment1	maxVal	transferEnv1	0.97
prediction	transfer	environment1	thresholdSet	transferEnv1	0.34

(Continues)

TABLE 1 (Continued)

field1	field2	field3	entity	family	examples
prediction	transfer	environment1	thresholdRule	transferEnv1	5% quantile of training presences
prediction	transfer	environment1	extrapolation	base	clamping; extrapolate function
evaluation	trainingData-Stats		AUC	binaryClassification	0.923
evaluation	testingDataStats		AUC	binaryClassification	0.923
evaluation	evaluationData-Stats		AUC	binaryClassification	0.923
code	software		platform	base	@Manual{title = {R: A Language and Environment for Statistical Computing}, author = {{R Core Team}}, organization = {R Foundation for Statistical Computing}, address = {Vienna, Austria}, year = {2017}, url = {https://www.R-project.org/}}

Note.: AUC = area under the curve. The full dictionary is available in Supporting Information S1. Fields specify the hierarchy of the metadata object while *entities* define the quantity of interest. *Entities* are assigned values as shown in the examples (example values are separated by semicolons). *Families* specify collections of related entities used for generating templates and checking for conditionally obligate entities. The examples column shows a variety of appropriate values that might be relevant in different studies.

variables used (:environment:variableName), the temporal extent of the environmental layers (:environment:yearMin, :environment:yearMax) and the source of the environmental data (:environment:sources). For example, occurrence information for our example includes (additional *entities* in Supporting Information S4):

```
data:occurrence:presenceSampleSize=290
data:occurrence:backgroundSampleSize=5084
data:occurrence:yearMin=1970
data:occurrence:yearMax=2000
```

3.5 | Data preparation

Information within the `dataPrep` field details any changes, cleaning or validation done to the data. Errors or inherent biases (i.e., spatial) in publicly available occurrence data are common (Serra-Diaz, Enquist, Maitner, Merow, & Svenning, 2018) and may have serious consequences for modelling (Merow, Allen, Aiello-Lammens, & Silander, 2016; Phillips et al., 2009). Common reasons for excluding coordinates include: coordinates not falling in the specified political division, coordinates reflecting non-native or cultivated occurrences, coordinates representing centroids of a political division, duplicated coordinates or biased spatial clustering (Aiello-Lammens, Boria, Radosavljevic, Vilela, & Anderson, 2015; Maitner et al., 2017; Robertson, Visser, & Hui, 2016; Serra-Diaz et al., 2018). Valid points may also need to be removed if they constitute environmental outliers that may strongly bias a model (Soley-Guardia, Radosavljevic, Rivera, & Anderson, 2014).

Within the `dataPrep` field there are four subfields: `:errors`, `:biological`, `:environmental` and `:geographic`. The `:errors` field contains information regarding any removal of duplicate (`:errors:duplicate`) or suspicious points (`:errors:questionablePointRemoval`). The `:geographic` field contains information related to geographic name standardization (`:geographic:geographicStandardization`) and occurrence point validations (`:geographic:geographicOutlierRemoval`, `:geographic:centroidRemoval`, `:geographic:pointInPolygon`) on the basis of geopolitical regions as well as geographic outlier removal (`:geographic:geographicOutlierRemoval`). The `:biological` field contains information related to taxonomic name standardization (`:biological:taxonomicHarmonization`) as well as the identification of records that are likely to represent introduced or cultivated species (`:biological:nonNativeRemoval`, `:biological:cultivatedRemoval`). The `:environmental` field contains data related to changes made to the environmental layers used, as well as occurrence point exclusion on the basis of environmental data (`:environmental:environmentalOutlierRemoval`).

In our simplified example, we removed records duplicated within cells (on the 10-km grid of the environmental layers) and thinned the occurrence data to reduce the effects of spatial autocorrelation:

```
dataPrep:biological:duplicateRemoval:rule='one
observation per cell'
dataPrep:geographic:spatialThin:rule="20km used
as minimum distance between points"
```

FIGURE 1 An example rmm object with values, based on the example from the main text. Top level fields are indicated with bold. Note that some output has been omitted from the figure for space, indicated by *truncated*

```

$ authorship :List of 6
  [truncated]
$ studyObjective:List of 3
  ..$ purpose : chr "transfer"
  ..$ rangeType: chr "potential"
  ..$ transfer : chr "detect unoccupied suitable habitat"
$ data :List of 4
  ..$ occurrence :List of 7
  ...$ dataType : chr "presence only"
  ...$ yearMin : num 1970
  ...$ yearMax : num 2000
  ...$ sources :List of 16
  [truncated]
  ...$ backgroundSampleSizeSet: int 5359
  ...$ presenceSampleSize : int 122
  ...$ backgroundSampleSize : int 5359
  ..$ environment:List of 7
  [truncated]
  ..$ dataNotes : chr "WorldClim data accessed through dismo v1.1-4"
  ..$ transfer :List of 1
  ...$ environment1:List of 6
  [truncated]
$ dataPrep :List of 3
  ..$ geographic :List of 1
  ...$ spatialThin:List of 1
  ...$ rule: chr "20km used as minimum distance between points"
  ..$ biological :List of 1
  ...$ duplicateRemoval:List of 1
  ...$ rule: chr "one observation per cell"
$ modelFit :List of 5
  ..$ algorithm : chr "Maxent 3.3.3k via dismo 1.1.4"
  ..$ selectionRules : chr "highest mean test AUC"
  ..$ finalModelSettings: Factor w/ 6 levels "L_1","L_2","L_3",...: 4
  ..$ partition :List of 5
  ...$ partitionSet : chr "spatial blocks"
  ...$ partitionRule : chr "block cross validation: partitions occurrence
    localities by finding the latitude "|__truncated__
  ...$ notes : chr "background points also partitioned"
  ...$ occurrenceSubsampling: chr "k-fold cross validation"
  ...$ numberFolds : int 4
  ..$ maxent :List of 5
  ...$ featureSet : chr "LQ"
  ...$ regularizationMultiplierSet: num 1
  ...$ samplingBiasRule : chr "ignored"
  ...$ notes : chr "ENMeval was used to compare models with
    L and LQ features, each using "|__truncated__
  ...$ numberParameters : num 13
$ prediction :List of 3
  ..$ extrapolation: chr "clamping"
  [truncated]
  ..$ continuous :List of 3
  ...$ units : chr "relative occurrence rate"
  ...$ minVal: num 1.19e-12
  ...$ maxVal: num 0.0164
$ evaluation :List of 1
  ..$ testingDataStats :List of 3
  ...$ AUC : num 0.802
  ...$ omissionRate: num [1:2] 0.0501 0.2207
  ...$ notes : chr "omission rate thresholds are 1) minimum training
    presence, 2) 10% training presence"
$ code :List of 2
  ..$ software : chr "@Manual{\n title = {R: A Language and
    Environment for Statistical Computing},\n "|__truncated__
  ..$ vignetteCodeLink: chr "https://github.com/cmerow/"|__truncated__

```

3.6 | Model fitting

The `modelFit` field has the largest variety of *entities* owing to the profusion of modelling algorithms and decisions applied in their use. A subfield

specifies the algorithm name and can be user-defined to accommodate newly developed algorithms. In cases where ambiguity may exist about algorithm definitions, for example, determining whether one should define `modelFit:algorithm = 'Poisson point process'`

or 'glm' because the latter can be fit with GLM (generalized linear model) software, we leave this to the authors' discretion and provide the `modelFit:algorithmNotes` entity if needed. It is worth remembering that the intention of `rmm` objects is to be human readable and therefore subject to context and interpretation. ...Notes entities, such as `modelFit:notes`, allow users to describe this context to the desired level of detail. The `modelFit` field contains subfields for specifying data partitioning methods (e.g., *k*-fold cross-validation), specification of how covariates are treated (e.g., scaled, z-scores) and algorithm-specific settings. For MAXENT modelling, we have specified comprehensive examples, while providing only minimal recommendations for other algorithms. We leave extensions to other algorithms for their expert users to recommend as part of our efforts to engage the research community in further development. For example, users can also specify their own custom *entities* to accommodate less common metadata. This flexibility ensures that our metadata framework is not so prescriptive that it excludes less-common modelling tools or those yet to be developed.

In our simplified example, we used MAXENT via the ENMeval R package (Muscarella et al., 2014) to compare different combinations of feature classes and different regularization parameters. Models were compared based on area under the curve (AUC) evaluated on test data, obtained from spatial block cross-validation. As `rmm` objects are designed to handle a single model per species, we report the optimal model settings only and include information in the relevant ...Notes entities on the model selection strategy. Had we used ensemble averaging over these candidate models, we would have reported the attributes of the ensemble and including attributes of the component models in the ...Notes fields.

```
rmm$modelFit$partition$partitionRule='spatial
block cross validation'
rmm$modelFit$maxent$featureSet='LQ'
rmm$modelFit$maxent$regularizationMultiplier-
Set=1
rmm$modelFit$maxent$samplingBiasRule='ignored'
rmm$modelFit$maxent$notes='ENMeval was used to
compare models with L (linear) and LQ (linear
and quadratic) features, each using regular-
ization multipliers of 1,2,3. The best model
was selected based on test AUC evaluated
under spatial block cross-validation.'
```

3.7 | Prediction

The `prediction` field describes common attributes of a variety of possible output types, including the prediction in geographic space (optionally a single prediction or the mean of multiple models), predictions transferred in space or time, and prediction uncertainty. For each of these prediction types, users specify the units (e.g., binary presence/absence, abundance, absolute probability of occurrence), the maximum and minimum values, and notes associated with interpretation. For each prediction type (except uncertainty), users can optionally specify a threshold value or rule to convert continuous predictions to binary. Finally, text can be provided to describe rules for extrapolation, building ensembles of

models and other optional attributes of model reporting. In our example study, we make predictions using MAXENT's "raw" (or relative occurrence rate; Merow, Smith, & Silander, 2013) values. Note the use of functions (`raster::cellStats()`; Hijmans, 2019) to fill in entities, where `p` is the prediction raster. Further, analogous entities related to transferring predictions to a new region, are shown in Supporting Information S4 for brevity.

```
rmm$prediction$continuous$units="relative oc-
currence rate"
rmm$prediction$continuous$minVal=raster::cell-
Stats(p,min)
rmm$prediction$continuous$maxVal=raster::cell-
Stats(p,max)
rmm$prediction$extrapolation="clamping"
```

3.8 | Evaluation

The `evaluation` field stores a range of statistics used to quantify model training, testing or overall evaluation. This follows recommendations common in machine learning (Hastie, Tibshirani, & Friedman, 2009) for splitting data into three subsets before model building: training, testing and evaluation. Training statistics are evaluated on the data used to fit, or train, the model. Testing statistics are calculated on data withheld from training and describe evaluation on test data to assess generality. Such testing statistics can be used for model selection or for weighting in model ensembles, and can help determine which model settings are optimal of those tested (answering the question "of the models run, which is 'best'?"). The evaluation data are independent of both training and testing data and provide a means to assess how well the selected/average model performs with out-of-sample prediction (answering the question "how good is the best model?"). While we recommend data partitioning as the most robust option, we realize that many studies do not have sufficient data – it is thus common to use testing data for evaluation. In this case, researchers should report their statistics as testing, and provide an `evaluation:notes` that these statistics were also used for evaluation. For training, testing and evaluation a common set of names of standardized statistics are provided (e.g., AUC, TSS (true skill statistic)); users can also include their own statistics and cite them in `evaluation:references`. Notably, we have designed the `rmm` object structure to accommodate a single model per taxon; this model can either be the output of a single algorithm, or the summary (i.e., mean or median) of a single algorithm fit to subsets of the data (e.g., *k*-fold cross-validation), or multiple models [e.g., an ensemble, as from the BIOMOD2 R package (Thuiller, Georges, Engler, & Breiner, 2019)]. In studies where multiple models are relevant to report for each species, a separate `rmm` object should be used for each model type.

In our example study, only AUC evaluated on test data was used to select optimal model settings. In general, it is better practice to examine multiple metrics. Note that we fill in values directly from those stored in an ENMeval object called `e`.

```
rmm$evaluation$trainingDataStats$AUC=e@
results[i,$trainAUC
rmm$evaluation$testingDataStats$AUC=e@re-
sults[i,$avg.test.AUC
```

3.9 | Code

The `code:` field stores obligate information about software references and versions as well as optional links to scripts hosted online. As `rmm` objects are designed to be human readable, information that enables true reproducibility is stored in these scripts, for example, hosted by journals in supplemental information or on Github. We recommend these files be free of constraints beyond those used by journals to avoid a prohibitive amount of work by authors, which discourages sharing their code. As biologists continue to strive toward greater reproducibility, we hope standards do emerge, but this is beyond the current scope of our metadata standards. We do however offer entities for different types of code, which currently include `code:demoCodeLink` (for brief, reduced functionality examples), `code:vignetteCodeLink` (for commented, tutorial-styled code) and `code:fullCodeLink` (for a full reproduction of the analysis). These distinctions aim to help users better understand what to expect from the code and for authors to target different audiences needing different levels of detail. We recommend that `code:codeNotes` include information on which platforms the code has been tested. In our example study, we cite the relevant R packages with:

```
rmm=rmmAutofillPackageCitation(rmm=rmm,
  packages=c('rgbif','sp','raster','dismo','ENMeval'))
```

3.10 | Vector-valued entities

Some entities are naturally defined as vectors so we adopt JSON formatting (JavaScript Object Notation; www.json.org) to help clearly define named vector-valued entities. For example, when specifying the spatial extent of the modelling domain, it is common to use the minimum and maximum coordinates (i.e., bounding box). To specify these limits unambiguously with JSON, we use (from the example study): `[{"xmin":-125,"xmax":-32,"ymin":-56,"ymax":40}]`. (In JSON syntax, the "string" describing the name of a quantity is in double quotations and its "value" is given following a colon.) Vector-valued entities are apparent in a number of cases: `data:environment:minVal` and `:maxVal` indicate the extremes of each environmental layer in the analysis (e.g., `[{"bio1":289,"bio12":7682,"bio16":2458}]`). Even if users are not familiar with JSON, the `jsonlite` package (Ooms, 2014) provides convenient tools to convert an R `data.frame` to JSON text (see vignettes). A number of vector-valued entities already have names defined, but we expect that use cases will arise that require users to extend JSON formatting to other entities.

3.11 | Multispecies studies

Thus far we have focused on studies that have a single species; however, RMMS are readily extended to include multiple species. As many model properties can (and arguably should) be specific to a particular species, we have also designed the metadata structure to accommodate multispecies studies through the use of a *taxon-Specific* column in the metadata dictionary (Table 1). This column

defines whether a given *entity* applies to all taxa in the study (e.g., `data:occurrence:dataType`), applies to each species separately by specifying a vector with a value for each species (e.g., `data:occurrence:presenceSampleSize`) or is a single value or vector with a value for each species (e.g., `data:occurrence:backgroundSampleSizeSet`). Hence the *taxonSpecific* column indicates whether or not a vector-valued *entity* describes different taxa (e.g., `data:environment:sources` may contain a vector of references to different sources and a value of *taxonSpecific* = no indicates that these sources apply generally and not to different taxa). Note that any entity with taxon-specific values can optionally be specified as `[{"species1":value1, "species2":value2}]`, but users can also choose the simpler multispecies vector formatting with `value1,value2`, etc.

In multispecies studies, *entities* can take single or multiple values and are associated with a vector of taxon names. Thus an *entity* may have a single value if it is constant for all study taxa, or a vector of values associated with their respective taxa. This framework can also be thought of as a table with columns for taxa and rows for *entities*. For example, a study containing two species would specify their names (using R syntax for convenience) as `data:occurrence:taxon=c('taxon1','taxon2')`, and all subsequent entities can be provided with values as vectors of length 2 when the value differs among species or length 1 when the value is the same among species [e.g., `data:occurrence:presenceSampleSize=c(24137, 4520)` and `data:occurrence:yearMax=2018`, respectively]. Models of each taxon in a study are likely to have different properties, such as presence sample size, but may also have different model settings. Indeed, model evaluation and ecological reality of the response may be greatly improved by tuning parameters to individual datasets (Merow et al., 2014, 2013; Muscarella et al., 2014).

For simplicity in multispecies studies, users can specify unique values of an *entity* for each species as just described or a character string describing a methodological rule for choosing the value. Most model settings have `...Set` and `...Rule` entities that users can choose from. For example, when thresholding continuous predictions to make a binary map, `modelFit:prediction:thresholdSet=c(.003,.002)` could be used to indicate the specific threshold values or alternatively `modelFit:prediction:thresholdRule='5% training presence'` could be used to identify the rule to determine the thresholds applied to multiple taxa. In cases where multiple modelling algorithms are relevant, we recommend making a separate `rmm` object for each algorithm – this is designed to keep `rmm` objects easily human readable and avoid confusion about *entities* that might have similar inputs but different names/interpretations with different modelling approaches.

3.12 | Common use cases

To help guide users through determining which *entities* to include, we define a suite of common *families* of *entities* in the metadata dictionary that may be relevant for a given study. As a baseline,

the *base family* defines the minimum set of entities used to define a typical model. Certain *entities* are obligate, such as those relating to data sources, while others are recommended as typically sufficient to meet research community standards, and yet others are entirely optional. Researchers can modularly combine the *base family* with other *families* to represent different workflows that most closely match their study type as a template (examples in Table 1). *Entities* can then be added or removed as seen fit (except for obligate *entities*, which can be left empty but not deleted so that the decision to omit them is readily apparent). Some *entities* are conditionally obligate: for example, if any *entities* in the optional field `prediction:transfer:environment1` (defining the environmental conditions, perhaps for some data set on future conditions for model transfer) are non-NULL, related *entities* must have non-NULL values (`:yearMin`, `:yearMax`, `:resolution`, `:extent`, `:sources`). Hence someone modelling the future extinction risk of a species with MAXENT could combine the families *obligate*, *dataPrep*, *maxent* (*entities* associated with the MAXENT modelling algorithm), and *transferEnv1* (*entities* associated with environmental conditions where a model is projected) as a starting point for their metadata template.

4 | rangeModelMetadata R PACKAGE

Although our RMMS framework is software agnostic, we simplify the process of building a metadata list by providing an R package, `rangeModelMetadata`, which provides a number of user-friendly tools that define, print, autofill, query and check `rmm` S3 objects. It begins by defining the *families of entities* relevant to the study to generate an empty template. These templates are defined as lists of lists that capture the natural hierarchical structure of our metadata documentation scheme. As shown in Figure 2, this structure allows users to get or set values of particular *entities* using the format `field1$field2$field3$entity` (e.g., `model$algorithmSettings$maxent$featureSet`, or `output$prediction$units` in the case where only the first two fields are relevant). To enable flexibility for analysis outside of R, we provide tools to export `rmm` objects as csv files. A package overview is available in Supporting Information S2 and worked examples are in Supporting Information S3; these are maintained and updated as R package vignettes on Github and CRAN.

The `rangeModelMetadata` package provides a number of convenience features. `rmmPrintFull()` displays only non-NULL entities while `rmmPrintEmpty()` displays only null entities to help determine missing information. These can further be parsed into obligate and optional entities. To reduce errors and simplify information entry, we provide a number of `rmmAutoFill...`() functions that capture relevant information from commonly used R objects during modelling. For example, in the simplest case, one can provide a `raster::stack` of environmental input layers or model predictions to automatically fill in associated metadata entities. Similar functionality exists for citations, occurrence data and `ENMeval` (Muscarella et al., 2014) objects. These functions also provide useful examples

for other package developers to write `rmmAutoFill...`() functions to connect to new packages. For further refining inputs, `rmmSuggests()` has predefined options for input entities and their values.

We provide a number of automated checks to help researchers detect potential issues (e.g., misspellings) and ensure some level of standardization with a number of `rmmCheck...`() functions. Checking standards are drawn directly from terms in the metadata dictionary and hence update automatically with any changes. Multiple checks are available, including those for standardized fields (`rmmCheckNames`), standardized entities (`rmmCheckValues`), missing fields (`rmmCheckMissingNames`) and empty entities (`rmmCheckEmpty`). Each check function returns information on names that are (a) matched exactly to standardized values, (b) names of partial matches to standardized values, and (c) unmatched names. This enables users to see what changes might be relevant while allowing them the flexibility to ignore the suggestions and include their own custom names or values. Checks for empty entities can be split into obligate, recommended and optional entities to help users determine missing information. For a final check of all entities in an `rmm` object, we provide the `rmmCheckFinalize()` function that runs all the `rmmCheck...`() functions together.

The `rangeModelMetadata` package include a number of other facilities. The base R function `str()` can print an `rmm` object to different field depths. `rmmToCSV()` exports the `rmm` object into a “flat” csv format that is readable by other software platforms and more human readable. Finally, users can also specify their own custom entities to accommodate metadata for less-common or currently undeveloped tools, for example, `modelFit:algorithm:algorithmSettings:'userDefinedEntity'=x` where `'userDefinedEntity'` is a name provided by the user and `x` is its value.

5 | DISCUSSION

We propose a comprehensive framework for recording metadata on range models that enhances transparency, reproducibility and sharing. To reduce the burden on researchers to provide this information, we have developed an R package with a variety of convenience functions to fill, suggest and check metadata objects efficiently. We anticipate that these advances will enable better comparisons between studies and synthesis across disciplines, improved models based on the ability to readily check for best practices, and improved citability and sharing of knowledge products.

Our decision to make `rmm` objects extensible, rather than tightly constrained, reflects our goal of prioritizing convenience to researchers, but involves some trade-offs. A rigidly structured object with strictly predefined entities and values for those entities would ensure standardization, prevent errors due to typos, and generally be more easily searchable. However, as the field of range modelling is always growing, an exhaustively prescriptive metadata framework would be impractical to maintain and would likely involve such a lengthy manual that it would inhibit use. Hence we have elected to implement a more lightweight and flexible framework with fewer

```

> rmm1=rmmTemplate(family=c('base'))
> str(rmm1,2)
List of 8
 $ authorship :List of 9
  ..$ rmmName      : NULL
  ..$ names        : NULL
  ..$ ownership    : NULL
  ..$ license      : NULL
  ..$ contact      : NULL
  ..$ relatedReferences: NULL
  ..$ authorNotes  : NULL
  ..$ miscNotes    : NULL
  ..$ doi          : NULL
 $ studyObjective :List of 4
  ..$ purpose      : NULL
  ..$ rangeType    : NULL
  ..$ invasion     : NULL
  ..$ transfer     : NULL
 $ data :List of 4
  ..$ occurrence :List of 6
  ..$ environment:List of 9
  ..$ observation:List of 3
  ..$ dataNotes  : NULL
 $ dataPrep :List of 1
  ..$ dataPrepNotes: NULL
 $ modelFit :List of 9
  ..$ algorithm    : NULL
  ..$ algorithmCitation : NULL
  ..$ speciesNumber : NULL
  ..$ selectionRules : NULL
  ..$ finalModelSettings: NULL
  ..$ notes        : NULL
  ..$ objective    :List of 4
  ..$ partition    :List of 3
  ..$ references    : NULL
 $ prediction :List of 4
  ..$ binary       :List of 2
  ..$ extrapolation: NULL
  ..$ transfer     :List of 2
  ..$ uncertainty  :List of 4
 $ evaluation :List of 2
  ..$ references   : NULL
  ..$ notes        : NULL
 $ code :List of 8
  ..$ software     :List of 2
  ..$ demoCodeLink : NULL
  ..$ vignetteCodeLink: NULL
  ..$ fullCodeLink : NULL
  ..$ demoDataLink : NULL
  ..$ vignetteDataLink: NULL
  ..$ fullDataLink : NULL
  ..$ codeNotes    : NULL
 - attr(*, "class")= chr [1:2] "list" "RMM"

```

FIGURE 2 An example rmm object template generated in R. Note the hierarchical list structure. This example includes only the entities that are considered fundamental for use with every range model. Top level fields are indicated with bold

entities that can be more readily adapted to any range modelling workflow. It remains the responsibility of the researchers, editors,

and data providers to curate the text in these entities to ensure clarity and precision.

To enable an evolving metadata dictionary we will maintain it on Github so that contributions and suggestions are readily tracked, discussed and incorporated. We ultimately plan to follow Github vocabulary management processes similar to those used by the Darwin Core maintenance group (see <https://github.com/tdwg/dwc>). We will serve as an initial governance board to moderate proposed changes but welcome others to join our team, particularly those with different expertise. Facilitating community-moderated evolution of the metadata dictionary will be the subject of future work and will depend critically on the reception and responses to the currently proposed framework. We aim for RMMS to be a fully open source and community driven enterprise.

RMMS has the potential to improve the review process for manuscripts using range models. Journals may choose to define their own families of standards for particular applications or to adopt those we propose. These standards [and convenience functions like `rmmCheckFinalize()`] will make it easier for authors checking model details before journal submission and for journal reviewers/editors checking the compliance and completeness of submitted `rmm` objects. To allow a broader user base to easily evaluate `rmm` objects, we have developed a web-based graphical interface (with the R package SHINY; Chang, Cheng, Allaire, Xie, & McPherson, 2019) that enables users to upload an `rmm` object either as a csv or RDS (R's data format for a single object) file and check for missing fields or standardization issues (Figure 3). It can be accessed within R using the two commands:

```

library(rangeModelMetadata)
rmmCheckShiny()

```

Because SHINY applications are built over the top of R, this allows us to use the exact R code that console users would use to check `rmm` objects. The code used for checking and the results can readily be exported so that editors can share these results with authors without ambiguity. Finally, the application includes options to submit multiple `rmm` objects to report differences among them for comparing with previous studies.

Defining community standards can support reporting and help encourage best practice approaches to science. Suboptimal methodologies will become more immediately transparent and requests for metadata information will encourage researchers to conduct more comprehensive analyses and supply information that is vital for their peers to understand, evaluate and use their work. Araújo et al. (2019) recently proposed a set of best practices and reporting standards for the use of species distribution models in biodiversity models; our metadata standards and tools reflect these same ideals. For example, best practices can be established by defining a family of the entities required for biodiversity assessments (e.g., a new `biodiversity` family). By proposing standardized values associated with acceptable practices for the biodiversity use case, best practices can be clearly defined [e.g., to characterize the quality of predictor variables proposed by Araújo et al. (2019)].

Range Model Metadata (RMM) Check

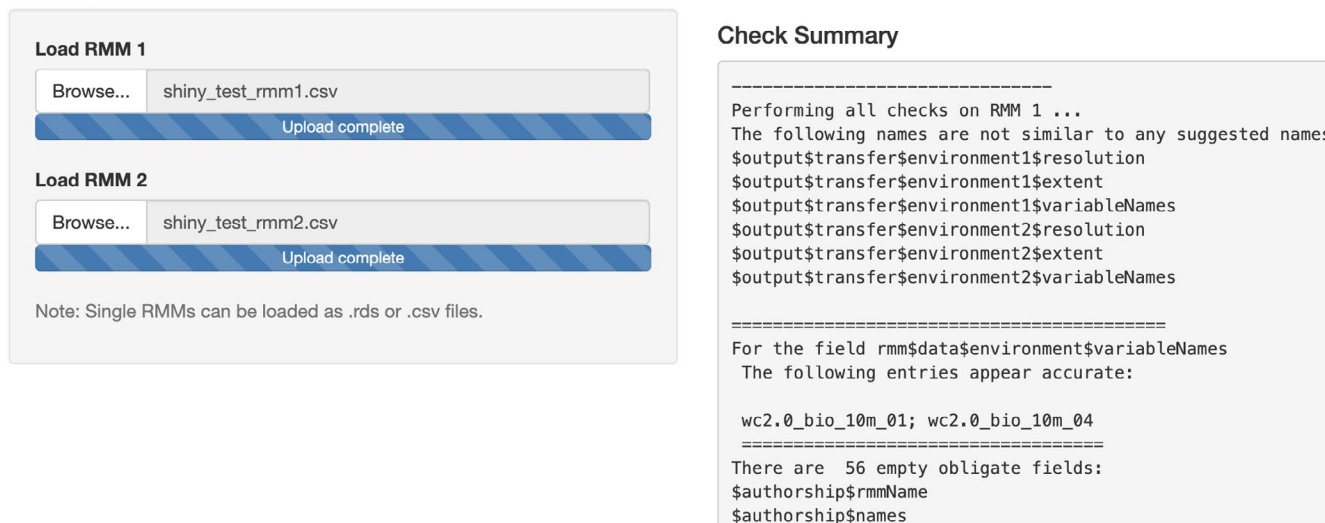


FIGURE 3 A screenshot of the web-based SHINY app that enables checking and comparing `rmm` objects without writing code [or accessible with `rmmCheckShiny()`]. The Check Summary continues on to report on a number of other comparisons, omitted here for the sake of space [Colour figure can be viewed at wileyonlinelibrary.com]

Community standards mean that both smaller-scale efforts or larger taxon-region-specific projects that produce range models can do so in a way that supports community efforts and assures that catalogues across independent efforts can be developed. Any downstream uses will benefit from the transparency enabled by the standards, which should enhance the rigor and credibility of range models for, for example, conservation application for more applied outcomes. Similar to how standards such as Darwin Core or Humboldt Core are facilitating the combination of point and inventory data of often vastly different origins (Guralnick et al., 2017; Wieczorek et al., 2012) in support of aggregators such as the GBIF, we hope that RMMS and their future evolution will set the stage for a more programmatic synthesis of range models and their products. For example, in MOL, which is integrating biodiversity information to develop a range of species distribution resources and both produces and consumes range models, RMMS open up the opportunity for a more informed visual and quantitative comparison and eventually integration of range models produced by different groups. Thus the standards open the door for contributed range models from taxon experts to enable their aggregation and integration in support of advancing the biodiversity knowledge base broadly.

As RMMS evolve and grow, we will facilitate other software developers to link their work easily to `rangeModelMetadata` and enable `rmm` objects to be largely autofilled based on the output of other R packages. For example, in complex cases involving more comprehensive range modelling workflows such as WALLACE (Kass et al., 2018), an R-based ecological modelling software, or those used by BIEN and MOL, filling in `rmm` object entities can be built into the workflow. In WALLACE an additional top level field, `wallace:`, is added to store additional information that can be used to reproduce a session. Other workflows may similarly

benefit from reading in settings from `rmm` objects to automatically select parameters, in which case `rmm` objects would serve as automatic lab notebooks to reproduce analyses. This next stage of integration with other software tools will serve as a way to further refine and maintain the metadata dictionary while engaging key range modelling teams in the process.

The range model metadata standards that we propose provide a number of tools to clarify and streamline reporting, sharing, evaluating and searching range models. We consider this the beginning of a community process rather than its endpoint, and the standards are therefore software agnostic, extensible and readily updated. If further engagement, adoption and advancement by others is successful, the proposed standards hold long-term benefits for the larger community and the impact of their work.

ACKNOWLEDGMENTS

C.M. acknowledges funding from NSF (National Science Foundation) grant DBI-1913673. C.M. and J.M.K. acknowledge funding from NSF grant DBI-1661510.

DATA AVAILABILITY STATEMENT

All code is maintained on Github (<https://github.com/cmerow/rangeModelMetadata>) and also served on CRAN (<http://cran.us.r-project.org/>).

ORCID

Cory Merow  <https://orcid.org/0000-0003-0561-053X>

Brian S. Maitner  <https://orcid.org/0000-0002-2118-9880>

Hannah L. Owens  <https://orcid.org/0000-0003-0071-1745>
 Jamie M. Kass  <https://orcid.org/0000-0002-9432-895X>
 Brian J. Enquist  <https://orcid.org/0000-0002-6124-7096>
 Walter Jetz  <https://orcid.org/0000-0002-1971-7277>
 Rob Guralnick  <https://orcid.org/0000-0001-6682-1504>

REFERENCES

- 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. <https://doi.org/10.1111/ecog.01132>
- Araújo, M. B., Anderson, R. P., Márcia Barbosa, A., Beale, C. M., Dormann, C. F., Early, R., ... Rahbek, C. (2019). Standards for distribution models in biodiversity assessments. *Science Advances*, 5, eaat4858. <https://doi.org/10.1126/sciadv.aat4858>
- Araújo, M., & New, M. (2007). Ensemble forecasting of species distributions. *Trends in Ecology and Evolution*, 22, 42–47. <https://doi.org/10.1016/j.tree.2006.09.010>
- Borba, C., & Correa, P. L. P. (2015). Application of metadata standards for interoperability between species distribution models. In E. Garoufalou, R. J. Hartley & P. Gaitanous (Eds.), *Metadata and semantics research* (pp. 113–118). Cham: Springer International Publishing.
- Chang, W., Cheng, J., Allaire, J. J., Xie, Y., & McPherson, J. (2019). shiny: Web application framework for R. R package version 1.3.2. <https://CRAN.R-project.org/package=shiny>
- Costa, W., Miranda, L., Borges, R., Saraiva, A., Imperatriz-Fonseca, V., & Giannini, T. (2018). The need of species distribution models metadata: Using species distribution model to address decision making on climate change. *Biodiversity Information Science and Standards*, 2, e25478. <https://doi.org/10.3897/biss.2.25478>
- Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: New 1-km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37, 4302–4315. <https://doi.org/10.1002/joc.5086>
- Franklin, J. (2010). *Mapping species distributions: Spatial inference and prediction*. Cambridge, UK: Cambridge University Press.
- GBIF.org. (2019). *GBIF home page*.
- Goecks, J., Nekrutenko, A., Taylor, J., & Galaxy Team. (2010). Galaxy: A comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology*, 11, R86. <https://doi.org/10.1186/gb-2010-11-8-r86>
- 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. (2005). Predicting species distribution: Offering more than simple habitat models. *Ecology Letters*, 8, 993–1009. <https://doi.org/10.1111/j.1461-0248.2005.00792.x>
- Guisan, A., Thuiller, W., & Zimmermann, N. E. (2017). *Habitat suitability and distribution models: With applications in R*. Cambridge, UK: Cambridge University Press.
- Guralnick, R., Walls, R., & Jetz, W. (2017). Humboldt Core - toward a standardized capture of biological inventories for biodiversity monitoring, modeling and assessment. *Ecography*, 41, 713–725. <https://doi.org/10.1111/ecog.02942>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. 2nd ed. New York: Springer-Verlag.
- Hijmans, R. J. (2019). Geographic Data Analysis and Modeling R package version 2.9-23. <https://CRAN.R-project.org/package=raster>
- Hijmans, R. J., Phillips, S., Leathwick, J., & Elith, J. (2017). *Dismo: Species distribution modeling*; 2013. R package version 1.1-4. <https://CRAN.R-project.org/package=dismo>
- Jetz, W., McGeoch, M. A., Guralnick, R., Ferrier, S., Beck, J., Costello, M. J., ... Turak, E. (2019). Essential biodiversity variables for mapping and monitoring species populations. *Nature Ecology and Evolution*, 3, 539–551. <https://doi.org/10.1038/s41559-019-0826-1>
- Jetz, W., McPherson, J. M., & Guralnick, R. P. (2012). Integrating biodiversity distribution knowledge: Toward a global map of life. *Trends in Ecology and Evolution*, 27, 151–159. <https://doi.org/10.1016/j.tree.2011.09.007>
- Joppa, L. N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., ... Emmott, S. (2013). Computational science. Troubling trends in scientific software use. *Science*, 340, 814–815. <https://doi.org/10.1126/science.1231535>
- Kass, J. M., Vilela, B., Aiello-Lammens, M. E., Muscarella, R., Merow, C., & Anderson, R. P. (2018). Wallace: A flexible platform for reproducible modeling of species niches and distributions built for community expansion. *Methods in Ecology and Evolution*, 9, 1151–1156.
- Maitner, B. S., Boyle, B., Casler, N., Condit, R., Donoghue, J., Durán, S. M., ... Enquist, B. J. (2017). The BIEN R package: A tool to access the Botanical Information and Ecology Network (BIEN) database. *Methods in Ecology and Evolution*, 9, 373–379.
- Merow, C., Allen, J. M., Aiello-Lammens, M., & Silander, J. A. Jr (2016). Improving niche and range estimates with Maxent and point process models by integrating spatially explicit information: Minxent. *Global Ecology and Biogeography*, 25, 1022–1036.
- Merow, C., Smith, M. J., Edwards, T. C., Guisan, A., McMahon, S. M., Normand, S., ... Elith, J. (2014). What do we gain from simplicity versus complexity in species distribution models? *Ecography*, 37, 1267–1281. <https://doi.org/10.1111/ecog.00845>
- Merow, C., Smith, M. J., & Silander, J. A. (2013). A practical guide to MaxEnt for modeling species' distributions: What it does, and why inputs and settings matter. *Ecography*, 36, 1058–1069. <https://doi.org/10.1111/j.1600-0587.2013.07872.x>
- Mesirov, J. P. (2010). Computer science. Accessible reproducible research. *Science*, 327, 415–416. <https://doi.org/10.1126/science.1179653>
- 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.
- Ooms, J. (2014). The jsonlite package: A practical and consistent mapping between JSON data and R objects. *arXiv:1403.2805 [stat. CO]*.
- Pereira, H. M., Ferrier, S., Walters, M., Geller, G. N., Jongman, R. H. G., Scholes, R. J., ... Wegmann, M. (2013). Ecology. Essential biodiversity variables. *Science*, 339, 277–278. <https://doi.org/10.1126/science.1229931>
- Peterson, A. T., Soberón, J., Pearson, R. G., & Anderson, R. P. (2011). *Ecological niches and geographic distributions* (1st ed.). NJ: Princeton University Press.
- Phillips, S. J., Anderson, R. P., & Schapire, R. E. (2006). Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, 190, 231–259. <https://doi.org/10.1016/j.ecolmodel.2005.03.026>
- Phillips, S. J., Dudík, M., Elith, J., Graham, C. H., Lehmann, A., Leathwick, J., & Ferrier, S. (2009). Sample selection bias and presence-only distribution models: Implications for background and pseudo-absence data. *Ecological Applications*, 19, 181–197. <https://doi.org/10.1890/07-2153.1>
- Robertson, M. P., Visser, V., & Hui, C. (2016). Biogeo: An R package for assessing and improving data quality of occurrence record datasets. *Ecography*, 39, 394–401. <https://doi.org/10.1111/ecog.02118>
- Serra-Diaz, J. M., Enquist, B. J., Maitner, B., Merow, C., & Svenning, J.-C. (2018). Big data of tree species distributions: How big and how good? *Forest Ecosystems*, 4, 30. <https://doi.org/10.1186/s40663-017-0120-0>

- Soley-Guardia, M., Radosavljevic, A., Rivera, J. L., & Anderson, R. P. (2014). The effect of spatially marginal localities in modelling species niches and distributions. *Journal of Biogeography*, 41, 1390–1401. <https://doi.org/10.1111/jbi.12297>
- Taylor, C. F., Field, D., Sansone, S.-A., Aerts, J., Apweiler, R., Ashburner, M., ... Wiemann, S. (2008). Promoting coherent minimum reporting guidelines for biological and biomedical investigations: The MIBBI project. *Nature Biotechnology*, 26, 889–896. <https://doi.org/10.1038/nbt.1411>
- Thuiller, W., Georges, D., Engler, R., & Breiner, F. (2019). biomod2: Ensemble platform for species distribution modeling [R package version 3.3-7.1]. <https://CRAN.R-project.org/package=biomod2>
- Thuiller, W., Lafourcade, B., Engler, R., & Araújo, M. B. (2009). BIOMOD—A platform for ensemble forecasting of species distributions. *Ecography*, 32, 369–373. <https://doi.org/10.1111/j.1600-0587.2008.05742.x>
- Warton, D., & Aarts, G. (2013). Advancing our thinking in presence-only and used-available analysis. *The Journal of Animal Ecology*, 82, 1125–1134. <https://doi.org/10.1111/1365-2656.12071>
- Wieczorek, J., Bloom, D., Guralnick, R., Blum, S., Döring, M., Giovanni, R., ... Vieglais, D. (2012). Darwin Core: An evolving

community-developed biodiversity data standard. *PLoS ONE*, 7, e29715. <https://doi.org/10.1371/journal.pone.0029715>

BIOSKETCH

Cory Merow is a quantitative ecologist interested in forecasting biological responses to global change. He likes pina coladas and getting caught in the rain.

How to cite this article: Merow C, Maitner BS, Owens HL, et al. Species' range model metadata standards: RMMS. *Global Ecol Biogeogr*. 2019;28:1912–1924. <https://doi.org/10.1111/geb.12993>