Ecological Informatics xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

MIAT: Modular R-wrappers for flexible implementation of MaxEnt distribution modelling

Sabrina Mazzoni [a], Rune Halvorsen [a], Vegar Bakkestuen [a,b]

[a] Geo-ecological Research Group, Dept. of Research and Collections, Natural History Museum, Univ. of Oslo, PO Box 1172 Blindern, Norway
[b] Norwegian Institute for Nature Research (NINA) Gaustadalléen 21, 0349 Oslo, Norway

ARTICLE INFO

Article history:
Received 30 December 2014
Received in revised form 5 June 2015
Accepted 2 July 2015
Available online xxxx

Keywords:
Distribution modelling practice
Integrated modelling framework
Object-orientation
Operational workflow
Practical MaxEnt toolbox
Embedded metadata

ABSTRACT

The maximum entropy (MaxEnt) method has gained widespread use for distribution modelling, mostly because of the practical simplicity offered by the maxent.jar software. Whilst MaxEnt was originally described as a machine learning method, recent studies have shown that the method can be explained in terms of maximum likelihood estimation. This opens for using MaxEnt with new settings and options, such as new model selection and model assessment criteria, and improved user control of the variable selection process. New practical tools are needed to explore the new opportunities and assess if they enhance model performance and ecological interpretability of the models. We present a new conceptual framework, the Modular and functionally Integrated component-based Approach (MIA) framework for practical distribution modelling by which the core components of the DM process are decoupled and then wrapped together more flexibly into component-based functional modules. Computational object-oriented and workflow approaches are integrated with ecological, statistical and modelling theory in order to handle the complexity associated with the full modelling process in a practical way. Objects (variables, functions, results, etc.) are defined according to specific modelling parameters. Properties (e.g., identities and content) are inherited between objects and new objects are created in a flexible and automated, yet traceable way. We operationalise this framework for MaxEnt by the MIA Toolbox (MIAT), a set of flexible, modular R-scripts (available in supplementary appendices) wrapped around maxent.jar and existing R-functions. MIAT covers the full range of options and settings for the maximum likelihood implementation of MaxEnt and provide flexible guidance of users through the DM process. A trail of models of increasing complexity is built to enhance traceability and interpretability, and to suit different modelling purposes. We briefly outline research questions that can be addressed by the MIAT.

1. Introduction

Distribution modelling (DM) has experienced a rapid rise since the paper by Guisan and Zimmermann (2000), and has developed into a separate branch of ecological and biogeographical science (Franklin, 2009). Along with this rise, an explosion of theoretical and conceptual frameworks, new methodologies and practical guidelines for their use, and software developments, have been published (see, e.g., Engler et al., 2012; Halvorsen, 2013; Loehle, 2012; and Thiele et al., 2012; and comprehensive reviews by, e.g., Franklin, 2009 and Peterson et al., 2011). Furthermore, guidelines have been provided for improving the quality and properties of the data used for DM, both for the modelled target and the environmental predictors (Gottschalk et al., 2011; Hanberry, 2013; Heikkinen et al., 2012; Heinänen et al., 2012; Roberts and Hamann, 2012). Paradoxically, these advances have made the DM process more complex and, thus, also increased the risk of suboptimal implementation of modelling practice (Aguirre-Gutiérrez et al., 2013; Austin, 2007; Guillera-Arroita et al., 2015; Halvorsen, 2013).

One of the most widely used methods for DM is Maximum Entropy Modelling (MaxEnt). MaxEnt's popularity amongst distribution modellers is, amongst others, due to the user-friendly software maxent.jar (Phillips, 2011; Phillips and Dudík, 2008; Phillips et al., 2004; Phillips et al., 2006); note the distinction used throughout this paper between MaxEnt the method and maxent.jar the software. The user-friendliness of the software is achieved by the integration of distinct modelling steps into one composite methodological procedure, implemented as a compiled tool with fixed choices of options that users can specify. Maxent.jar employs a "black-box" like approach to manage the complex computational and theoretical requirements of the MaxEnt method. This approach thus trades simplicity for limitations on user control and, apparently, understanding of the method (Halvorsen, 2013; Yackulic et al., in press). Even when used in conjunction with other programming packages such as DISMO, BIOMOD, or ENMTools (Hijmans and Elith, 2013; Thuiller et al., 2009; Warren et al., 2010), the systematic

E-mail addresses: Sabrina.Mazzoni@nhm.uio.no (S. Mazzoni), Rune.Halvorsen@nhm.uio.no (R. Halvorsen), Vegar.Bakkestuen@nhm.uio.no, Vegar.Bakkestuen@nina.no (V. Bakkestuen).

http://dx.doi.org/10.1016/j.ecoinf.2015.07.001
1574-9541/

exploration and tracking of the large number of models tested in search for the final model is impractical. This has contributed to most users accepting default settings and options (the 'default MaxEnt practice'), and exploring alternatives very minimally (Halvorsen, 2013; Halvorsen et al., 2015; Merow et al., 2013; Yackulic et al., in press).

Recent studies have brought theoretically and ecologically more intuitive understandings of MaxEnt modelling, and MaxEnt practitioners have been urged to draw more explicit links between the structure of the model, properties of the data and the ecological knowledge (Elith et al., 2011; Halvorsen et al., 2015; Merow et al., 2013; Renner and Warton, 2013). These developments have also opened for new model selection methods and several other options and functionalities which should be explored and used in a systematic and flexible way. However, this requires user control of all steps in the DM process (Dormann et al., 2007; Hastie et al., 2009; Leathwick et al., 2006; Reineking and Schröder, 2006; Reineking, 2006) and implementation of alternatives to the currently fixed shrinkage (Tibshirani, 1994) model selection method (Halvorsen et al., 2015; Renner and Warton, 2013; Warren and Seifert, 2010).

The Maximum Likelihood (ML) explanation of MaxEnt offers enhanced user control and flexibility of MaxEnt options without reducing accessibility and interpretability (see Halvorsen et al., 2015 for more details), but is practically difficult to perform in practise using existing tools. A flexible toolbox that could implement these options in a way that is as simple and as user-friendly as possible, whilst still giving users the added control and overview of the complexity, would greatly improve the uptake of the proposed new options. Development of such a toolbox is likely to benefit from explicit reframing and integration of theoretical and applied concepts from the DM and informatics fields—i.e., a new framework for practical distribution modelling, before the operationalisation of this framework for the ML interpretation of MaxEnt.

In this paper we first present a flexible integrating framework to the practise of distribution modelling. Our aim with this framework is to provide practitioners with the conceptual and practical control needed to explore and understand more fully the modelling process, whilst maintaining as much simplicity and accessibility as possible in practise. Secondly, we use this framework to build a flexible toolbox to implement the options opened for by the ML explanation of MaxEnt. The general concepts behind the framework may, in principle, be extended to DM methods other than MaxEnt.

2. MIA—a flexible, modular framework for practical distribution modelling

Theoretically, the DM process can be described as a set of general procedures that are organised into steps, which typically are carried out sequentially. Some of these steps are mandatory, whilst others are optional. The major steps of the DM process may be arranged into three 'core components of DM': modelling purpose, 'ecological model', and 'data properties and statistical methods' (Austin, 2007), to which Halvorsen (2012) added 'applications' (Fig. 1). Which steps are carried out, and in what order, to some extent depend on the purpose of the DM project, the data used, the method chosen, and the expertise of the practitioner (see, e.g., Franklin, 2009; Guisan and Zimmermann, 2000; Halvorsen, 2013; Peterson et al., 2011). Nevertheless, practical

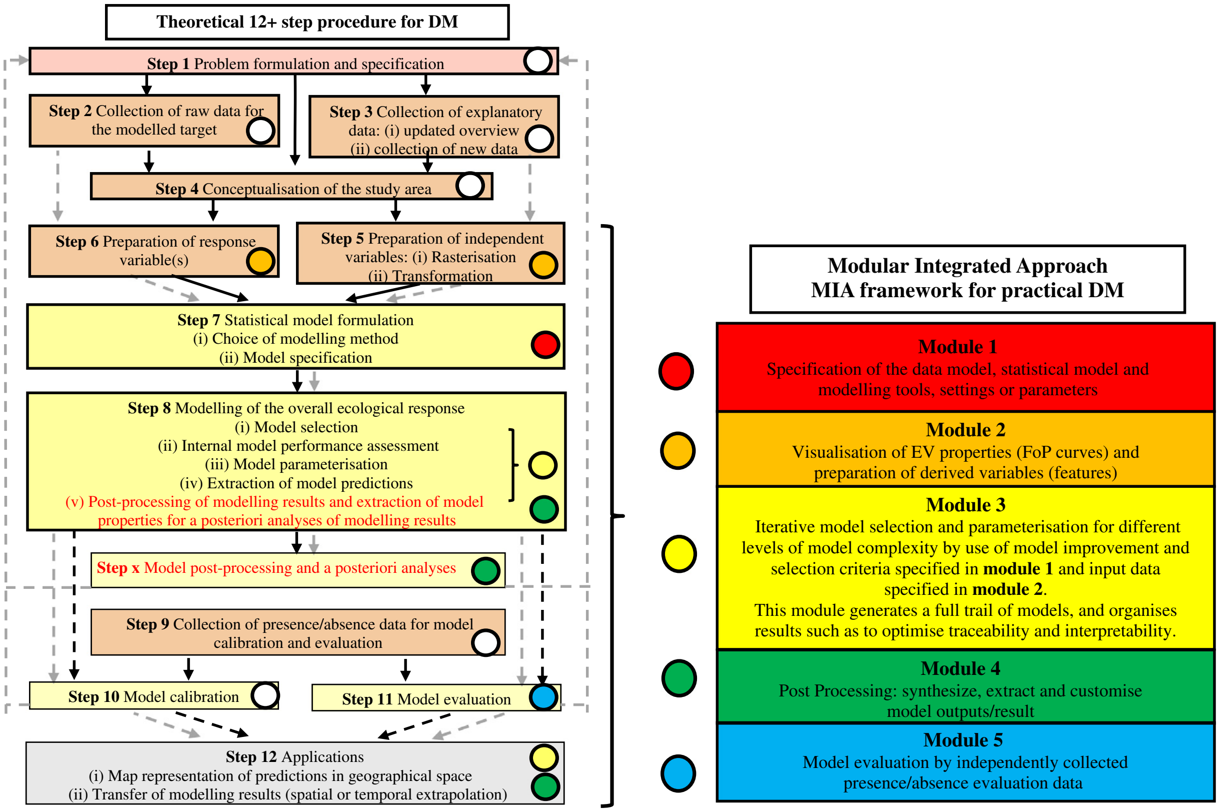

Theoretical 12+ step procedure for DM
Step 1 Problem formulation and specification
Step 2 Collection of raw data for the modelled target
Step 3 Collection of explanatory data: (i) updated overview (ii) collection of new data
Step 4 Conceptualisation of the study area
Step 6 Preparation of response variable(s)
Step 5 Preparation of independent variables: (i) Rasterisation (ii) Transformation
Step 7 Statistical model formulation (i) Choice of modelling method (ii) Model specification
Step 8 Modelling of the overall ecological response (i) Model selection (ii) Internal model performance assessment (iii) Model parameterisation (iv) Extraction of model predictions
(v) Post-processing of modelling results and extraction of model properties for a posteriori analyses of modelling results
Step x Model post-processing and a posteriori analyses
Step 9 Collection of presence/absence data for model calibration and evaluation
Step 10 Model calibration
Step 11 Model evaluation
Step 12 Applications (i) Map representation of predictions in geographical space (ii) Transfer of modelling results (spatial or temporal extrapolation)
Modular Integrated Approach
MIA framework for practical DM
Module 1
Specification of the data model, statistical model and modelling tools, settings or parameters
Module 2
Visualisation of EV properties (FoP curves) and preparation of derived variables (features)
Module 3
Iterative model selection and parameterisation for different levels of model complexity by use of model improvement and selection criteria specified in module 1 and input data specified in module 2.
This module generates a full trail of models, and organises results such as to optimise traceability and interpretability.
Module 4
Post Processing: synthesize, extract and customise model outputs/result
Module 5
Model evaluation by independently collected presence/absence evaluation data

Fig. 1. Relationships between the 12 steps of the theoretical distribution modelling (DM) process recognised by Halvorsen (2013) (left) and the five modules (some of) these steps are reorganised into for practical DM according to the MIAT framework. Dots with similar colours are grouped together in an MIA module. Steps indicated by red font are not recognised by Halvorsen (2012). The 12 steps are grouped into three composite steps, 'ecological model' (red background), 'data model' (orange background), and 'statistical model' (yellow background), in accordance with Austin (2002). Steps that are mandatory for a study to be distribution modelling are indicated by thick borders. Steps involved in re-iteration of the model are indicated by grey lines. Broken lines indicate optional pathways.

DM implies that several steps are performed in an order that may differ from the sequence of theoretical steps. We propose and describe the MIA (Modular and functionally Integrated component-based Approach) framework for practical DM which integrates the ecological and statistical theory of DM with object-oriented or functional modelling and workflow in two ways: *i*) by first dissolving and then reassembling the single components of the DM process into a set of operational modules that guide the core elements of the modelling practice more intuitively and computationally efficiently, and *ii*) by providing an object-oriented workflow environment (Barseghian et al., 2010; Michener et al., 2007; Michener and Jones, 2012; Reichman et al., 2011) for the modelling procedure which ensures flexibility and user control of the modelling process. The framework balances requirements for the overview of the components of the DM process, whilst at the same time being flexibly adapted to different DM purposes. The MIA framework thus links DM core components with DM theory (Fig. 1) and guides modellers through the practical implementation of the DM process more explicitly and flexibly.

An object-oriented modular approach was chosen because it provides intuitive, traceable automation and offers scalable implementations and integration (Holst, 2013; Parr, 2005; Pereira et al., 2006; Silvert, 1993), for a diverse range of users and applications (Steiniger and Hay, 2009; Thiele et al., 2012; Thuiller et al., 2009). Additionally, the flexible regrouping of the components (Bentlage and Shcheglovitova, 2012; Cushing et al., 2007) that is enabled with object orientation provide the explicit and flexible link between the theory and the practice. The framework's full value lies in understanding it as a flexible guide through the DM process, both conceptually (verbally) and as practical tools.

3. MIAT—a flexible toolbox for practical MaxEnt modelling

We operationalise the MIA framework as a modular integrated toolbox, MIAT, to provide a practical workflow wrapper around MaxEnt modelling practice. Table 1 is the descriptive overview of the toolbox and guides the modelling process. The structure of the toolbox reflects the hierarchical nested modularity of the MIA framework (see the right-hand side of Fig. 1 and Fig. 2), and is implemented as a set of coded scripts using the R programming environment (R Development Core Team, 2013). These R-scripts (wrappers) carry out the core steps of the MaxEnt modelling process and directly operationalise the options offered by the ML implementation of MaxEnt (Halvorsen, 2013; Halvorsen et al., 2015), using existing packages and tools, such as maxent.jar (Phillips et al., 2004; Phillips et al., 2006) and Windows batch files.

The MIAT toolbox contains several files, all starting with the code MIAT_. The next two letters of the filename indicate the module and eventual further letters, the component(s) addressed. The scripts are provided in Appendix 1 (see Table 1 for overview).

Module 1 of the MIAT toolbox for MaxEnt modelling (Table 1) first identifies the core components of DM by MaxEnt and then breaks these components down into smaller objects to which properties are assigned and amongst which relationships are defined. Modules 2 and 3 regroup these entities into functional components and using existing tools (maxent.jar, .bat files) produce a wide range of automated outputs in the form of objects. Modules 4 and 5 provide auxiliary analyses of modelling results (model post-processing) options. A detailed description of the main components of the MIAT, with “vignettes” (screen shots from practical runs) is provided in Appendix 2.

The MIAT toolbox offers hierarchical nested modularity at different levels, and gives the user control and adds flexibility, traceability and interpretability to the entire process DM process by MaxEnt. Flexibility is achieved by creating a trail of models for which a range of model selection criteria and other specific settings and options can be explored, amongst others by comparative analyses. Traceability (and improved interpretability) is achieved within each module by providing the necessary level of information and passing it on by an iterative nested approach (Maley and Caswell, 1993) by which objects (and their names) contain metadata to help users keep track of which model settings were used (Fig. 2).

Each module produces objects outside R, such as files and folders with embedded metadata referring to the source (which module/script it came from), the type of information included, and the necessary object parameter codes. Most files produced are comma separated tables (.csv), R-generated graphics (.jpg or.wmf) and Windows batch files (.bat) that are used to run maxent.jar and to assemble, copy and rename results files in a traceable manner. All scripts end with saving key objects separately as R-datafiles. These files contain input to, and output from, the script, for improved traceability and interpretability of the results (Stock et al., 2012; Villa et al., 2009). The files can easily be exported to spreadsheet or graphics software, folders, R-datafiles, etc., and file names can be reused or as titles/legend in graphical output.

All modules start with a commented “title section” and some descriptive background. Each script within the module file is separated by a line comment that says “Next script” to facilitate searches. All scripts start with listing the key component(s) necessary to start off with (usually output from the previous module/script), and the *scriptnamecode* object gets populated accordingly.

The toolbox is extensively commented. The comment lines are meant to guide the process for interested users, and to provide metadata for the modelling process. General comments for sections are preceded by one or more (#) or (#_DesCode) whilst 'commented-out inspection codes' are preceded by (#_InsCode). This is a series of code such as 'str(object name)' or 'head(object name)', provided both for convenience and as guidance to enable the user to inspect the process along the way. Furthermore, optional codes (#_OptCode), as well as alternative lines of code that can be used to test/develop further tools (#_AltCode), are provided.

The modules create and in turn use two main types of input objects (as well as other minor objects) throughout the process: a ‘starting’ parameters object (MIAPar); and a ‘starting’ data objects (M1_N_SWD, M3_N_SWD_RV). These two objects are used to produce the main output objects specific to that particular stage of the modelling process, which in turn become the starting objects for the next module/stage.

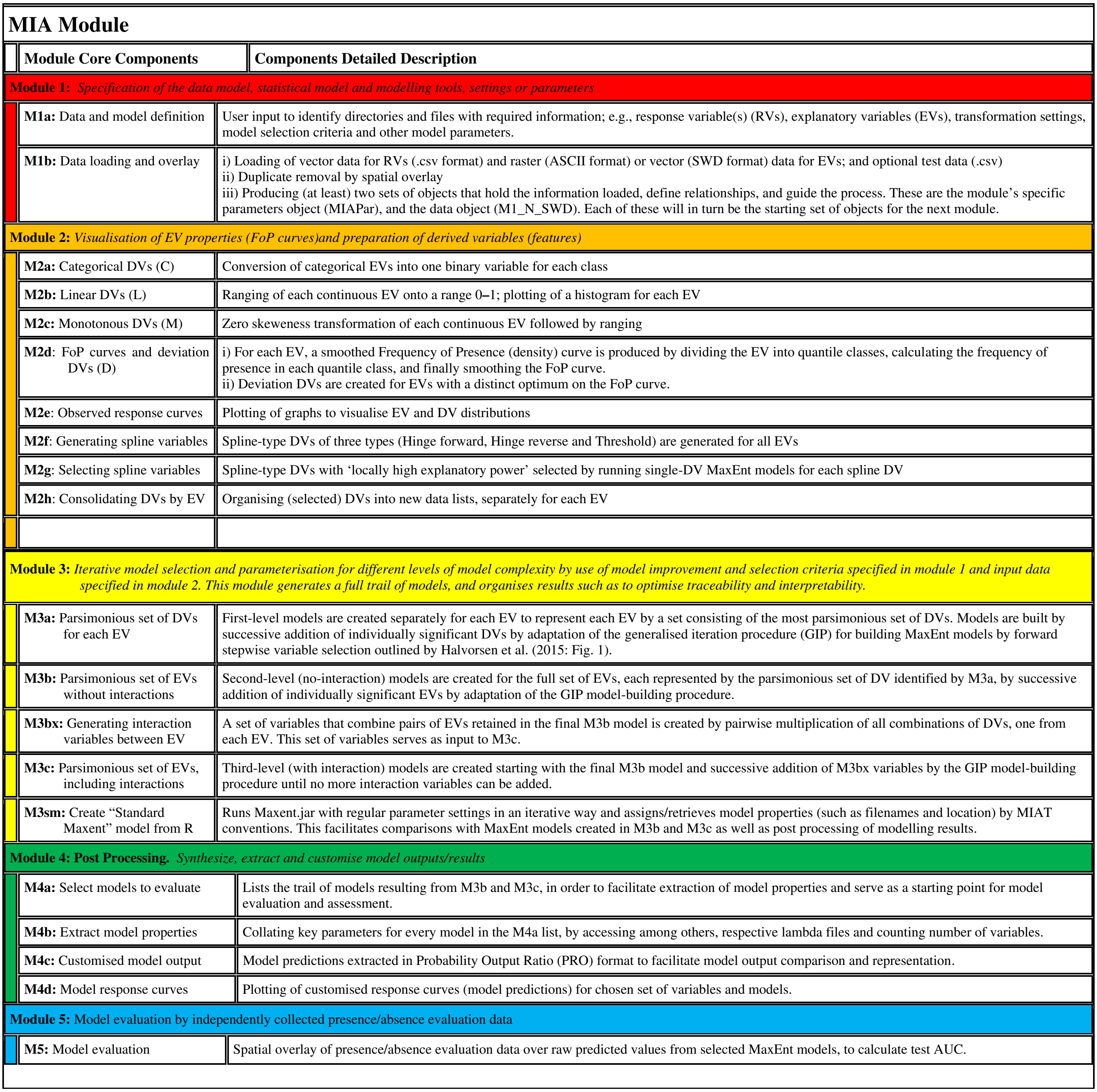
The R-object type ‘list’ is used throughout as lists give the flexibility necessary for building an increasing trail of information, that varies in size and dimension. Lists are open for making use of the tag (names) structure that gets built along the way, to create new objects and link them, and access them accordingly. Access to ‘embedded metadata’ is then made possible at different stages of the modelling process. The lists may each contain several R dataframes, which typically store two kinds of information: ‘parameter’ lists and ‘data’ lists. The parameter lists (such as MIAPar) specify all key parameters internal to R required by the scripts themselves, statistical model parameters (such as the internal model performance assessment criterion and other specifications of the model selection procedure), modelling tools-specific parameters (such as required by maxent.jar, Windows or R), and also data-specific parameters (such as location and number and types of response and explanatory variables) that are specified throughout the process.

The use and structure of lists can be exemplified by the M2_N_SWD_RV data lists, specific for the last stage of MIAT Module 2 which holds training and background data to be used in Module 3 for the MaxEnt model selection and parameterisation process. The syntax of these lists is

M2_N_SWD_RV[[rv]] [[ev]][row,columns]

The list object contains two nested lists, indicated by double square brackets, and a dataframe for each. The highest level is “RV”, containing 1 or more response variable(s) for the modelled target. The dimension of this level (the length of this list) equals the number of response variables. The next level is the “EV” level which contains the environmental variables, the length of which depends on the number of EV considered

Table 1
Descriptive overview and guide of the **MIA Toolbox**, a practical workflow wrapper for flexible implementation of the maximum likelihood explanation of MaxEnt.



MIA Module

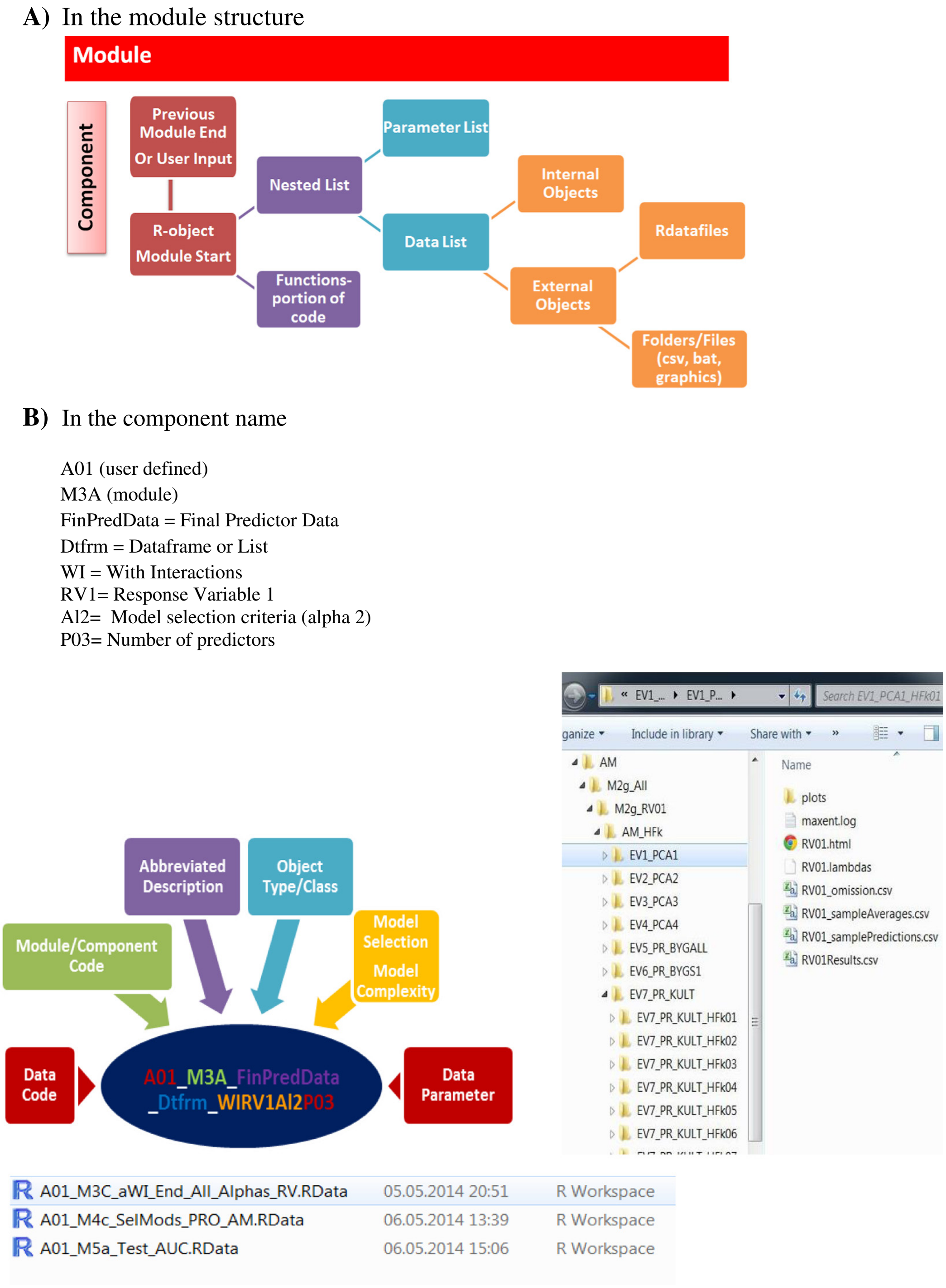
Module Core Components	Components Detailed Description
Module 1: *Specification of the data model, statistical model and modelling tools, settings or parameters*	
M1a: Data and model definition	User input to identify directories and files with required information; e.g., response variable(s) (RVs), explanatory variables (EVs), transformation settings, model selection criteria and other model parameters.
M1b: Data loading and overlay	i) Loading of vector data for RVs (.csv format) and raster (ASCII format) or vector (SWD format) data for EVs; and optional test data (.csv) ii) Duplicate removal by spatial overlay iii) Producing (at least) two sets of objects that hold the information loaded, define relationships, and guide the process. These are the module's specific parameters object (MIAPar), and the data object (M1_N_SWD). Each of these will in turn be the starting set of objects for the next module.
Module 2: *Visualisation of EV properties (FoP curves)and preparation of derived variables (features)*	
M2a: Categorical DVs (C)	Conversion of categorical EVs into one binary variable for each class
M2b: Linear DVs (L)	Ranging of each continuous EV onto a range 0–1; plotting of a histogram for each EV
M2c: Monotonous DVs (M)	Zero skeweness transformation of each continuous EV followed by ranging
M2d: FoP curves and deviation DVs (D)	i) For each EV, a smoothed Frequency of Presence (density) curve is produced by dividing the EV into quantile classes, calculating the frequency of presence in each quantile class, and finally smoothing the FoP curve. ii) Deviation DVs are created for EVs with a distinct optimum on the FoP curve.
M2e: Observed response curves	Plotting of graphs to visualise EV and DV distributions
M2f: Generating spline variables	Spline-type DVs of three types (Hinge forward, Hinge reverse and Threshold) are generated for all EVs
M2g: Selecting spline variables	Spline-type DVs with 'locally high explanatory power' selected by running single-DV MaxEnt models for each spline DV
M2h: Consolidating DVs by EV	Organising (selected) DVs into new data lists, separately for each EV
Module 3: *Iterative model selection and parameterisation for different levels of model complexity by use of model improvement and selection criteria specified in module 1 and input data specified in module 2. This module generates a full trail of models, and organises results such as to optimise traceability and interpretability.*	
M3a: Parsimonious set of DVs for each EV	First-level models are created separately for each EV to represent each EV by a set consisting of the most parsimonious set of DVs. Models are built by successive addition of individually significant DVs by adaptation of the generalised iteration procedure (GIP) for building MaxEnt models by forward stepwise variable selection outlined by Halvorsen et al. (2015: Fig. 1).
M3b: Parsimonious set of EVs without interactions	Second-level (no-interaction) models are created for the full set of EVs, each represented by the parsimonious set of DV identified by M3a, by successive addition of individually significant EVs by adaptation of the GIP model-building procedure.
M3bx: Generating interaction variables between EV	A set of variables that combine pairs of EVs retained in the final M3b model is created by pairwise multiplication of all combinations of DVs, one from each EV. This set of variables serves as input to M3c.
M3c: Parsimonious set of EVs, including interactions	Third-level (with interaction) models are created starting with the final M3b model and successive addition of M3bx variables by the GIP model-building procedure until no more interaction variables can be added.
M3sm: Create "Standard Maxent" model from R	Runs Maxent.jar with regular parameter settings in an iterative way and assigns/retrieves model properties (such as filenames and location) by MIAT conventions. This facilitates comparisons with MaxEnt models created in M3b and M3c as well as post processing of modelling results.
Module 4: Post Processing. *Synthesize, extract and customise model outputs/results*	
M4a: Select models to evaluate	Lists the trail of models resulting from M3b and M3c, in order to facilitate extraction of model properties and serve as a starting point for model evaluation and assessment.
M4b: Extract model properties	Collating key parameters for every model in the M4a list, by accessing among others, respective lambda files and counting number of variables.
M4c: Customised model output	Model predictions extracted in Probability Output Ratio (PRO) format to facilitate model output comparison and representation.
M4d: Model response curves	Plotting of customised response curves (model predictions) for chosen set of variables and models.
Module 5: Model evaluation by independently collected presence/absence evaluation data	
M5: Model evaluation	Spatial overlay of presence/absence evaluation data over raw predicted values from selected MaxEnt models, to calculate test AUC.

at the end of M2. The third level is the R dataframe object, consisting of rows and columns for the actual data being considered, in.csv table format referred to by maxent.jar as "sample with data". Each column in this dataframe specifies *i*) the geographical coordinates of each record, *ii*) observed presences for that specific response variable or unobserved background, and *iii*) as many columns of data values for each background data in the form of derived variables at that location.

Comprehensive examples of DM modelling using the MIAT scripts, with real data, are provided by Halvorsen et al. (2015, Supplementary Material, ECOG-00565), and Bendiksby et al. (2014, Supplementary Material, jbi12347-sup-0001-AppendixS1–S3).

4. Discussion

The modular MIAT toolbox for DM by MaxEnt presented here, which contains R-scripts produced in accordance with the MIA framework, allows the user to tackle the complexity of distribution modelling by MaxEnt in a flexible and practical way. It enhances user control over the rapidly accumulating, detailed, information produced during the DM process whilst maintaining overall simplicity and integration in a nested modular structure. Practical decoupling of components is then made possible. Necessary linkages are provided via the explicit definition of each object/component's properties, identities and relationships,



A) In the module structure

B) In the component name

A01 (user defined)
M3A (module)
FinPredData = Final Predictor Data
Dtfrm = Dataframe or List
WI = With Interactions
RV1= Response Variable 1
Al2= Model selection criteria (alpha 2)
P03= Number of predictors

```
◢ AM
  ◢ M2g_All
    ◢ M2g_RV01
      ◢ AM_HFk
        ▷ EV1_PCA1
        ▷ EV2_PCA2
        ▷ EV3_PCA3
        ▷ EV4_PCA4
        ▷ EV5_PR_BYGALL
        ▷ EV6_PR_BYGS1
        ◢ EV7_PR_KULT
          ▷ EV7_PR_KULT_HFk01
          ▷ EV7_PR_KULT_HFk02
          ▷ EV7_PR_KULT_HFk03
          ▷ EV7_PR_KULT_HFk04
          ▷ EV7_PR_KULT_HFk05
          ▷ EV7_PR_KULT_HFk06
```

```
plots
maxent.log
RV01.html
RV01.lambdas
RV01_omission.csv
RV01_sampleAverages.csv
RV01_samplePredictions.csv
RV01Results.csv
```

```
A01_M3C_aWI_End_All_Alphas_RV.RData   05.05.2014 20:51   R Workspace
A01_M4c_SelMods_PRO_AM.RData          06.05.2014 13:39   R Workspace
A01_M5a_Test_AUC.RData                06.05.2014 15:06   R Workspace
```

Fig. 2. Hierarchical nested modularity at different levels. Objects' (modules, scripts, components, files, etc) structures and names that reflect their source, relationship, purpose and identities (embedded metadata) improving traceability and interpretability. As many levels or parameter components as necessary, flexibly decided by user and by parameters themselves. Colour coding is being used to match this traceability.

within and across each of these different levels, by use of the metadata embedding concept for the naming of objects. Automated outputs are saved both inside and outside R so they can be accessed with other software, in a variety of formats, both tabular and graphical. The MIAT scripts thus may serve as both a guide as well as a conduit for carrying out the full modelling process, and make results more interpretable. We believe that the component-based modular MIAT toolbox thus has the flexibility needed for practical implementation of the broadened scope of MaxEnt modelling provided by the maximum likelihood explanation, as called for by Halvorsen (2013) and Halvorsen et al. (2015). Its practicability is further enhanced by the use of open source approaches and accessible software.

The basic MIA framework itself loosely draws on, and integrates concepts from several disciplines and, accordingly, is interdisciplinary in its nature (Richardson and Whittaker, 2010; Store and Kangas, 2001). Construction of the framework is guided by the same main aim as modelling itself, to simplify in a way that balances complexity and simplicity. The coding of each script in the MIAT toolbox is therefore kept rather simple and only a limited amount of R-object classes and functionalities are employed. Accordingly, the scripts themselves require a minimum of user programming skills whilst, of course, understanding of the MaxEnt method and the DM process is mandatory.

The scripts are intended to act on two levels at the same time: they operationalise automated production of DM models, and they record the process that leads to these models. Thus, the scripts themselves function as an additional metadata embedding wrapper of the entire process, in accordance with the principle that "the best time to collect metadata is whilst the data is being developed" (quote from the FGDC Metadata workbook version 2, FGDC-std-001-1998). This also reduces the number of potential sources of error.

The modularity of the MIA toolbox opens for better integration and interactivity, in exploring different but related modelling purposes, in the exploration of the data, and in the development of DM methodology itself, all of which may be approached within one single modelling exercise. However, in contrast to previous frameworks for DM, the MIA framework does not seek to meet all desired functionalities in one linear sequence of steps. Instead, core components in the DM process are identified to which properties are assigned and relationships defined based both on general principles as well as on specific rules. Then, based on the component's identity (class, type, and dimension) and the context, the functions perform actions in a flexible and modular way and produce results that are organised in a nested, hierarchical, manner. The MIA framework and scripts thus make a direct link between the modelling purpose, the statistical tools and the practical tools, further enabling a tighter integration of theory and practice, from which the discipline is likely to benefit (Austin, 2002; Guisan et al., 2006; Hirzel and Le Lay, 2008; Peterson et al., 2011). The MIAT tools presented here, created using the MIA framework, enable practical testing of the new options and settings for MaxEnt, opened up by the ML explanation of MaxEnt, such as alternative transformations of predictor variables and subset selection methods with new model performance assessment criteria (Halvorsen, 2013; Halvorsen et al., 2015). The framework may also be generalised to other DM methods.

The MIAT consists of separate R-scripts rather than functions. The reason for this is that separate scripts enable users to carry out their analysis with minimal additional coding, and allows them to "visibly" follow (via loops and if statements) the process and add more flexibility in their implementation. In the long term, after extensive experience has been gained from practical use, our ambition is to build an R library for practical DM by MaxEnt.

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.ecoinf.2015.07.001.

Acknowledgments

Thanks go to fellow Geo-ecology group members, who have contributed in various ways to the development of this framework and the toolbox by providing data, theoretical and practical discussions. Particular thanks go to Thijs C. van Son and Knut Anders Hovstad for sharing R knowledge and R-scripts; to Inger Auestad and John Wirkola Dirksen (and their data) for being the main guinea pigs during the never ending testing process. Also thanks to the anonymous reviewers and to the Editor. Finally, special thanks first and foremost to Katrina Roen, as well as Bente Støa, Anders K. Wollan, Desalegn Chala Galete and Louis Boumans for their valuable feedback on this manuscript.

This work has been carried out as part of a doctoral research grant at the Natural History Museum, University of Oslo, with additional support from the Norwegian Directorate for Nature Management.

References

Aguirre-Gutiérrez, J., Carvalheiro, L.G., Polce, C., van Loon, E.E., Raes, N., Reemer, M., Biesmeijer, J.C., 2013. Fit-for-Purpose: Species Distribution Model Performance Depends on Evaluation Criteria – Dutch Hoverflies as a Case Study. PLoS ONE 8 e63708.

Austin, M.P., 2002. Spatial prediction of species distribution: an interface between ecological theory and statistical modelling. Ecol. Model. 157, 101–118.

Austin, M., 2007. Species distribution models and ecological theory: a critical assessment and some possible new approaches. Ecol. Model. 200, 1–19.

Barseghian, D., Altintas, I., Jones, M.B., Crawl, D., Potter, N., Gallagher, J., Cornillon, P., Schildhauer, M., Borer, E.T., Seabloom, E.W., Hosseini, P.R., 2010. Workflows and extensions to the Kepler scientific workflow system to support environmental sensor data access and analysis. Ecological Informatics 5, 42–50.

Bendiksby, M., Mazzoni, S., Jørgensen, M.H., Halvorsen, R., Holien, H., 2014. Combining genetic analyses of archived specimens with distribution modelling to explain the anomalous distribution of the rare lichen Staurolemma omphalarioides: long-distance dispersal or vicariance? J. Biogeogr. 41, 2020–2031.

Bentlage, B., Shcheglovitova, M., 2012. NichePy: modular tools for estimating the similarity of ecological niche and species distribution models. Methods in Ecology and Evolution 3, 484–489.

Cushing, J., Nadkarni, N., Finch, M., Fiala, A., Murphy-Hill, E., Delcambre, L., Maier, D., 2007. Component-based end-user database design for ecologists. Journal of Intelligent Information Systems 29, 7–24.

Dormann, C.F., McPherson, J.M., Araújo, M.B., Bivand, R., Bolliger, J., Carl, G., Davies, R.G., Hirzel, A., Jetz, W., Kissling, W.D., Kühn, I., Ohlemüller, R., Peres-Neto, P.R., Reineking, B., Schröder, B., Schurr, F.M., Wilson, R., 2007. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. Ecography 30, 609–628.

Elith, J., Phillips, S.J., Hastie, T., Dudík, M., Chee, Y.E., Yates, C.J., 2011. A statistical explanation of MaxEnt for ecologists. Divers. Distrib. 17, 43–57.

Engler, R., Hordijk, W., Guisan, A., 2012. The MIGCLIM R package – seamless integration of dispersal constraints into projections of species distribution models. Ecography 35, 872–878.

Franklin, J., 2009. Mapping Species Distributions: Spatial Inference and Prediction. Cambridge University Press, Cambridge.

Gottschalk, T.K., Aue, B., Hotes, S., Ekschmitt, K., 2011. Influence of grain size on species–habitat models. Ecol. Model. 222, 3403–3412.

Guillera-Arroita, G., Lahoz-Monfort, J.J., Elith, J., Gordon, A., Kujala, H., Lentini, P.E., McCarthy, M.A., Tingley, R., 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.

Guisan, A., Zimmermann, N.E., 2000. Predictive habitat distribution models in ecology. Ecol. Model. 135, 147–186.

Guisan, A., Lehmann, A., Ferrier, S., Austin, M., Overton, J.M.C., Aspinall, R., Hastie, T., 2006. Making better biogeographical predictions of species' distributions. J. Appl. Ecol. 43, 386–392.

Halvorsen, R., 2012. A gradient analytic perspective on distribution modelling. Sommerfeltia 35, 1–165.

Halvorsen, R., 2013. A strict maximum likelihood explanation of MaxEnt, and some implications for distribution modelling. Sommerfeltia 1.

Halvorsen, R., Mazzoni, S., Bryn, A., Bakkestuen, V., 2015. Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt. Ecography 38, 172–183.

Hanberry, B.B., 2013. Finer grain size increases effects of error and changes influence of environmental predictors on species distribution models. Ecol. Inform. 15, 8–13.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. 2nd ed. Springer, New York.

Heikkinen, R.K., Marmion, M., Luoto, M., 2012. Does the interpolation accuracy of species distribution models come at the expense of transferability? Ecography 35, 276–288.

Heinänen, S., Erola, J., von Numers, M., 2012. High resolution species distribution models of two nesting water bird species: a study of transferability and predictive performance. Landsc. Ecol. 27, 545–555.

Hijmans, R.J., Elith, J., 2013. Species Distribution Modelling with R. The R foundation for statistical computing (http://cran.r-project.org/web/packages/dismo/vignettes/sdm.pdf).

Hirzel, A.H., Le Lay, G., 2008. Habitat suitability modelling and niche theory. J. Appl. Ecol. 45, 1372–1381.

Holst, N., 2013. A universal simulator for ecological models. Ecological Informatics 13, 70–76.

Leathwick, J.R., Elith, J., Hastie, T., 2006. Comparative performance of generalized additive models and multivariate adaptive regression splines for statistical modelling of species distributions. Ecol. Model. 199, 188–196.

Loehle, C., 2012. Relative frequency function models for species distribution modeling. Ecography 35, 487–498.

Maley, C.C., Caswell, H., 1993. Implementing i-state configuration models for population dynamics: an object-oriented programming approach. Ecol. Model. 68, 75–89.

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.

Michener, W.K., Jones, M.B., 2012. Ecoinformatics: supporting ecology as a data-intensive science. Trends in ecology & evolution (Personal edition) 27, 85–93.

Michener, W., Beach, J., Jones, M., Ludäscher, B., Pennington, D., Pereira, R., Rajasekar, A., Schildhauer, M., 2007. A knowledge environment for the biodiversity and ecological sciences. Journal of Intelligent Information Systems 29, 111–126.

Parr, C.S., Espinosa, R., Dewey, T., Hammond, G., Myer, P., 2005. Building a biodiversity content management system for science, education, and outreach. Data Science Journal 4, 1–11.

Pereira, A., Duarte, P., Norro, A., 2006. Different modelling tools of aquatic ecosystems: A proposal for a unified approach. Ecological Informatics 1, 407–421.

Peterson, A.T., Soberón, J., Pearson, R.G., Anderson, R.P., Martínez-Meyer, E., Nakamura, M., Araújo, M.B., 2011. Ecological Niches and Geographic Distributions (MPB-49). Princeton University Press, Princeton.

Phillips, S.J., 2011. A Brief Tutorial on MaxEnt. AT&T Research, Princeton, NJ.

Phillips, S.J., Dudík, M., 2008. Modeling of species distributions with MaxEnt: new extensions and a comprehensive evaluation. Ecography 31, 161–175.

Phillips, S.J., Dudík, M., Schapire, R., 2004. A Maximum Entropy Approach to Species Distribution Modeling. Anonymous (Anonymous) Anonymous), Proceedings of the 21st International Conference on Machine Learning. ACM Press, New York, pp. 655–662.

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

Reichman, O.J., Jones, M.B., Schildhauer, M.P., 2011. Challenges and Opportunities of Open Data in Ecology. Science 331, 703–705.

Reineking, B., Schröder, B., 2006. Constrain to perform: regularization of habitat models. Ecol. Model. 193, 675–690.

Renner, I.W., Warton, D.I., 2013. Equivalence of MAXENT and Poisson point process models for species distribution modeling in ecology. Biometrics 69, 274–281.

Richardson, D.M., Whittaker, R.J., 2010. Conservation biogeography — foundations, concepts and challenges. Divers. Distrib. 16, 313–320.

Roberts, D.R., Hamann, A., 2012. Method selection for species distribution modelling: are temporally or spatially independent evaluations necessary? Ecography 35, 792–802.

Silvert, W., 1993. Object-oriented ecosystem modelling. Ecological Modelling 68, 91–118.

Steiniger, S., Hay, G.J., 2009. Free and open source geographic information tools for landscape ecology. Econ. Inf. 4, 183–195.

Stock, K., Stojanovic, T., Reitsma, F., Ou, Y., Bishr, M., Ortmann, J., Robertson, A., 2012. To ontologise or not to ontologise: an information model for a geospatial knowledge infrastructure. Comput. Geosci. 45, 98–108.

Store, R., Kangas, J., 2001. Integrating spatial multicriteria evaluation and expert knowledge for GIS-based habitat suitability modelling. Landsc. Urban Plan. 55, 79–93.

Thiele, J.C., Kurth, W., Grimm, V., 2012. RNetLogo: an R package for running and exploring individual-based models implemented in NetLogo. Methods Ecol. Evol. 3, 480–483.

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

Tibshirani, R., 1994. Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B 267–288.

Villa, F., Athanasiadis, I.N., Rizzoli, A.E., 2009. Modelling with knowledge: a review of emerging semantic approaches to environmental modelling. Environ. Model. Softw. 24, 577–587.

Warren, D.L., Seifert, S.N., 2010. Ecological niche modeling in MaxEnt: the importance of model complexity and the performance of model selection criteria. Ecol. Appl. 21, 335–342.

Warren, D.L., Glor, R.E., Turelli, M., 2010. ENMTools: a toolbox for comparative studies of environmental niche models. Ecography 33, 607–611.

Yackulic, C.B., Chandler, R., Zipkin, E.F., Royle, J.A., Nichols, J.D., Grant, E.H.C., Veran, S., 2013. Presence-only modelling using MAXENT: when can we trust the inferences? Methods Ecol. Evol. 4 (in press).