Distribution modelling by MaxEnt: from black box to flexible toolbox

Sabrina Mazzoni

**Dissertation presented for the degree of
Philosophiae Doctor**

2016

Geo-Ecological Research Group
Department of Research and Collections
Natural History Museum
University of Oslo, Norway

© **Sabrina Mazzoni, 2016**

*Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo
No. 1736*

ISSN 1501-7710

All rights reserved. No part of this publication may be
reproduced or transmitted, in any form or by any means, without permission.

Cover: Hanne Baadsgaard Utigard.
Print production: Reprosentralen, University of Oslo.

Dedicata a papa', Ramin e la mia taonga

*"What we call the beginning is often the end.
And to make an end is to make a beginning. The end is where we start from."*

from the poem Little Gidding by T.S. Eliot, 1943.

CONTENTS

Abstract.....	5
List of Papers.....	7
Introduction.....	9
Background and Motivation	9
Aims and Objectives.....	11
Basic Theoretical Foundation.....	12
Opening up the theoretical considerations for practitioners (aim 1)	12
Proposing a flexible modelling practice (aim 2)	17
Accessible flexible toolbox for practical implementation (aim 3)	21
Empirical exploration of the practice of Distribution Modelling by MaxEnt	25
Materials and Methods.....	25
The study objects and basic dataset properties (aim 4)	26
Modelled Targets and Response Variables	26
Study Area and Explanatory Variables	30
Detection and mitigation of potential sampling bias (aim 5)	33
Effects of spatial autocorrelation in the response variable (aim 6).....	34
Model selection: importance of options and settings (aim 7)	34
Relating model complexity, performance and modelling purpose (aim 8)	37
Model Complexity	37
Modelling Purpose	37
Overall model performance assessment and evaluation.....	38
Results	39
The study object and basic dataset properties (aim 4).....	39
Detection and mitigation of potential sampling bias (aim 5).....	41
Effects of spatial autocorrelation in the response variable (aim 6).....	41
Model selection: options and settings (aim 7)	42
Relating model complexity, performance and modelling purpose (aim 8)	42
Discussion	44
The study object and basic dataset properties (aim 4).....	44
Detection and mitigation of potential sampling bias (aim 5).....	46
Effects of spatial autocorrelation in the response variable (aim 6).....	46
Model selection: options and settings (aim 7)	47
Relating model complexity, performance and modelling purpose (aim 8)	48
Concluding remarks and recommendations.....	50
Acknowledgements	53
References.....	57
Supplementary Materials	67
S1. MIAT Toolbox Code.....	68
Papers	

Abstract

The easier access to increasingly powerful computational approaches and tools in the field of distribution modelling, has contributed to a proliferation of data, applications, practitioners, guidelines, and novel theoretical understandings. Recognising the dynamic link in how these elements influence one another is critical as the discipline and practices develop. The challenge of how to implement the statistically and computationally complex theory behind the MaxEnt modelling method has been overcome by the practical simplicity of the powerful, platform independent and free Java™ tool, maxent.jar. Lowering this computational, and accessibility threshold, has meant the increased use and further development of relevant digital ecological data, such as biodiversity/occurrence records held in natural history collections worldwide (GBIF -Global Biodiversity Information Facility) and GIS layers of spatio-temporal environmental background layers being developed across a diverse range of fields.

However, the computational advantages of the fixed options offered by the software have come at the expense of a full exploration of the potentials of this statistical method. Over time, the popularity of the practical shortcuts have resulted in an uncritical acceptance of the defaults, a conflation of the statistical method with the software's black box approach, and a disconnection between theoretical and practical implications of the modelling process. A more flexible and explicit integration of these two, facilitates a much needed comparison between, and testing of, these theoretical and practical defaults, options and settings.

The aim of this thesis is to reduce the gap between the how practitioners can work with these practical tools, their understanding the body of DM theory, and MaxEnt in particular. PAPER 1 lays out the theoretical description of a novel interpretation of MaxEnt, with new settings and options, such as a new model selection and model assessment criteria, and improved user control of the variable selection process. To test this new theory in a practical way, new informatics driven approaches and tools were developed. PAPER 2 provides their detailed description and presents them as a modular toolbox in the form of a set of flexible R-scripts and functions. This new MaxEnt modelling approach and toolbox are used in PAPER 3, which looks specifically at how to identify and tackle the potential effects of sampling bias in presence only (PO) data obtained from museum collections. The application value of this alternative MaxEnt modelling procedure (aMp) is further explored and tested in PAPERS 4 and 5, where conservation management issues are addressed, as well as model purpose, model fitting and properties of the data. PAPER 4 explores how distribution modelling can be combined with phylogeographic analysis to address spatial temporal conservation issues. PAPER 5 makes use of fine grained remotely sensed LiDAR data, to explore issues related both to data properties (accuracy, spatial autocorrelation) and model complexity (variable and model selection, and model improvement). All MaxEnt models are evaluated against an independently collected field dataset, and theoretical and practical implications are discussed. PAPER 6 makes full use of this new theoretical approach and practical toolbox, and addresses MaxEnt model selection strategy by testing eight different combinations of model complexity and data properties. Finally, the paper discusses additional benefits these tool enhancements of the MaxEnt model performance and also the ecological interpretability are discussed.

In modelling, there is no single or best approach that works for everyone. There are always alternative approaches owing to our individual differences as practitioners, not solely based on the modelling tools or purposes alone. This thesis makes explicit use of both Ecological and Informatics approaches to perform a broad-scoped assessment of the relative performance of different combinations of MaxEnt options and their settings for DM with different modelling purposes, including of the specific properties of the data. By adding a flexible and traceable way to tackle this both theoretically and practically, I've attempted the reduce gap between the how the practitioners can work with the tools and the body of theory.

List of Papers

PAPER 1. Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt.

Rune Halvorsen, Sabrina Mazzoni, Anders Bryn, and Vegar Bakkestuen.
Ecography (2015), **38**, 172-183. DOI: 10.1111/ecog.00565

PAPER 2. MIAT: Modular R-wrappers for flexible implementation of MaxEnt distribution modelling.

Sabrina Mazzoni, Rune Halvorsen and Vegar Bakkestuen.
Ecological Informatics (2015), **30**, 215-221.
Open Access: <http://dx.doi.org/10.1016/j.ecoinf.2015.07.001>

PAPER 3. Sampling bias in presence-only data used for species distribution modelling: Assessment and effects on models.

Bente Støa, Rune Halvorsen, Sabrina Mazzoni, and Vladimir I. Gusarov.
Sommerfeltia. (In press)

PAPER 4. 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?

Mika Bendiksby, Sabrina Mazzoni, Marte Holten Jørgensen, Rune Halvorsen and Håkon Holien.
Journal of Biogeography (2014), **41**, 2020–2031. DOI: 10.1111/jbi.12347

PAPER 5. How important are choice of model selection method and spatial autocorrelation of presence data for distribution modelling by MaxEnt?

Rune Halvorsen, Sabrina Mazzoni, John W. Dirksen, Erik Næsset, Terje Gobakken and Mikael Ohlson.
Ecological Modelling (2016), **328**, 108-118.
DOI: dx.doi.org/10.1016/j.ecolmodel.2016.02.021

PAPER 6. Optimal MaxEnt model selection strategy evaluated by use of independent test data.

Sabrina Mazzoni, Rune Halvorsen, Vegar Bakkestuen, Inger Auestad, Trine Bekkby, Johannes Breidenbach, Desalegn Chala, John W. Dirksen, Anette Edvardsen, Dag Endresen, Lars Erikstad, Hege Gundersen, Einar Heegaard, Knut Anders Hovstad, Eli Rinde and Anders K. Wollan.
(Manuscript currently under review at *Global Ecology and Biogeography*)

Introduction

Background and Motivation

As an almost fashionable trend, the field of distribution (also known as: prediction, niche, species prediction, habitat suitability) modelling, has received growing interest, both within the scientific community (Guisan & Zimmermann 2000; Austin 2007; Franklin 2009; Peterson *et al.* 2011) and among the policy and management professionals (Mörtberg, Balfors & Knol 2007; Mazzoni *et al.* 2011; Guisan *et al.* 2013; Polce *et al.* 2013; Gould *et al.* 2014). This is not surprising, considering the growing urgency to tackle ongoing human induced threats to biodiversity, locally and globally, in particular increased land use pressure and climate change (del Barrio *et al.* 2006; Thuiller *et al.* 2008; Heller & Zavaleta 2009; Buckland *et al.* 2014). This trend in use of DM has exploded so rapidly, that a wide range of conceptual frameworks, statistical methods and, analytical and computational tools, have proliferated, reflecting the diverse nature of this inherently interdisciplinary field (Elith & Leathwick 2009; Franklin 2009; Peterson *et al.* 2011). The pace and complexity of how each of these individual components have themselves developed, is equally dynamic and diverse.

Because of the strong results in comparative studies, particularly with regards to modelling performance and modelling purpose (Hernandez *et al.* 2006; Gibson, Barrett & Burbridge 2007; Elith & Graham 2009; Tognelli *et al.* 2009), and of the practical simplicity offered by the *maxent.jar* tool in implementation, distribution modelling by Maximum Entropy (MaxEnt) (Jaynes 1957), has become very popular amongst ecologists. As a non-linear statistical modelling method, MaxEnt (Graham *et al.* 2004; Phillips, Dudík & Schapire 2004; Dudík, Phillips & Schapire 2007; Phillips 2010) can use presence-only occurrence data from existing natural history or research collections, something that in the last decade has become increasingly available through digital portals such as the Global Biodiversity Information Facility (www.gbif.org; Telenius 2011), and has also been shown to produce good prediction models with small sample sizes (Hernandez *et al.* 2006; Wisz *et al.* 2008; Mateo, Felicísimo & Muñoz 2010). Furthermore, the free Java™ compiled tool implementing this method requires minimal ecological or technical expertise to produce a wide range of automated outputs (graphs, tables, maps, reports, html files) that appeal to a broad range of users. Despite the apparent simplicity of use, or perhaps as a result of it, not all results generated have been either: accessed, adequately reported or understood. Users commonly misrepresent the resulting “default map” as the MaxEnt model itself, and often include little details on final models’ parameters. There’s also likely been a general lack of exploration of all options and settings offered by either the software or other statistical interpretations of this method itself. In fact, over time, there’s been a conflation of the two terms, so that most MaxEnt studies perform model selection and model complexity control only via the shrinkage method of the ℓ_1 -regularisation approach (Tibshirani 1996; Phillips, Anderson & Schapire 2006; Hastie, Tibshirani & Friedman 2009), implemented by the software. This widespread acceptance of the defaults (from here onwards called default MaxEnt practice – dMp), was initially explained by ecologists’ general lack of familiarity with machine-learning and Bayesian statistical concepts (Elith *et al.* 2011; Merow, Smith & Silander 2013). The fact that

machine learning concepts are not easily translatable into the ecological realities, may have led to MaxEnt being described as a “*black box*” and inspired independently several researchers (Elith *et al.* 2011; Fitzpatrick, Gotelli & Ellison 2013; Halvorsen 2013; Renner & Warton 2013) to open it up to alternative statistical interpretations.

The ability to derive the MaxEnt method through principles of strict maximum likelihood estimation (sMLE) (Halvorsen 2013), allows such opening up of both theoretical and practical considerations. The conceptually simpler and more intuitive approach is also more familiar to ecologists, and a more explicit link between the methods and its ecological interpretation can be made. The sMLE interpretation of MaxEnt offers flexible options for models selection methods, such as decoupling of the model selection from the model improvement criteria and more control of the variable selection process. This opening however, is in practice more complicated to implement with existing set of tools, particularly in view of a full exploration that is similarly accessible to existing MaxEnt modellers. Thus new, more flexible approaches are needed to be able to untangle the process in a way that is both guided and informed by its theoretical and practical considerations.

More recently, a growing number of studies documenting how this established dMp practice has produced highly complex models, both in terms of the number of parameters and the number of environmental variables included (Anderson & Gonzalez 2011; Warren & Seifert 2011; Auestad *et al.* 2012; Halvorsen *et al.* 2015), prompting theoretical and practical questions about the appropriateness of these models. The practitioners’ choices of model selection procedures, regularization method, and strictness of the criterion used to compare alternative models (Reineking & Schröder 2006) control the degree of model complexity. However, exercising this control is practically impossible in the dMp models, as these options are fixed into one single parameter (ℓ_1 -regularisation) rather than decoupled, as proposed above, by the more open approach to model selection and parameterisation of MaxEnt.

Finally, another common source of suboptimal model performance may be strictly due to the properties of the data set itself, such as its inherent sampling bias (Vaughan & Ormerod 2003b; Kadmon & Allouche 2007; Phillips *et al.* 2009; Fourcade *et al.* 2014) or spatial autocorrelation (Peres-Neto 2006; Dormann *et al.* 2007; Santika & Hutchinson 2009; Thibaud *et al.* 2014). As a Presence Only (PO) method, resulting MaxEnt models are particularly susceptible to both (Veloz 2009; Anderson & Gonzalez 2011; Merckx *et al.* 2011). Use of the background target-group approach (BTG, Phillips & Dudík 2008) to mitigate for sampling bias has become really popular (Bystrakova *et al.* 2012; Millar & Blouin-Demers 2012; Crall *et al.* 2013), despite the fact that it relies on assumptions that are practically impossible to validate (such as that the presence and BTG sets of observations contain similar bias (Mateo *et al.* 2010). Additionally, thorough evaluations of this approach have not yet been performed (but see Stokland, Halvorsen & Støa 2011; Heibl & Renner 2012; Fourcade *et al.* 2014). How spatial autocorrelation affects the performance of distribution models is also still not well known or understood (Dormann *et al.* 2007; Santika & Hutchinson 2009). Furthermore, the current practice of data splitting to evaluate model performance means that any bias contained in the training dataset will be passed onto the

test data, further limiting the ability to appropriately assess these models (Edwards *et al.* 2005; Veloz 2009; Edvardsen, Bakkestuen & Halvorsen 2011a; Halvorsen 2012).

Aims and Objectives

As such, the aims and objectives of this thesis are presented as theoretical and empirical, with papers ordered and discussed accordingly. PAPERS 1 and 2 detail the basic foundations worked out empirically in PAPERS 3-6. The important underlying topic of how to assess model performance is addressed throughout.

Specifically, in establishing the basic **theoretical foundation** the aims are to:

Aim 1: Opening up the theoretical options by presenting the theoretical considerations of new methodological opportunities offered by the strict Maximum Likelihood estimation (sMLE) interpretation of MaxEnt; from both an ecological, and informatics perspective.

Aim 2: Propose a flexible modelling practice to guide and inform the DM process in a more open, accessible, and integrated way.

Aim 3: Develop an accessible **toolbox** for its practical implementation.

The **empirical** exploration of the **practice** of DM by MaxEnt is structured according to the three main components identified by Austin (2007):

- i.* Properties of the *ecological model*. Idiosyncratic properties of the objects studied *as such* which are outside the control of the modeller i.e., the biological properties of the modelled target and the climatic, geological, and geomorphological characteristics of the study area.
- ii.* Properties of the *data model*, i.e., of the empirical data sets as such, resulting from the filtering implicit in the design of any study (rasterization of the study area, sampling of the response and predictor variables, etc.).
- iii.* Properties of the *statistical model* such as how the modelling procedure is specified, including choice of modelling method, options and their settings.

To which, explicit consideration of the *practical tools* to explore these is also added.

Thus, with respect to model performance, the aims are to determine the importance of:

Aim 4: The study objects and **basic dataset properties** (components *i*, *ii*).

Aim 5: Detecting and mitigating **potential sampling bias** in presence-only data (*ii*).

Aim 6: Effects of **spatial autocorrelation** in the response variable (*ii*).

Aim 7: Model selection method, including statistical and practical options and settings, and corresponding methods for variable selection (*iii*).

Aim 8: Relating **model complexity, performance** and **modelling purpose** for **overall model assessment** and **evaluation**.

Though presented last, the eighth aim guides the entire process, and is fundamental in achieving the others.

Basic Theoretical Foundation

Laying out theoretical foundation is an essential element of building sound tools or methodologies that in turn seek to explore novel theoretical questions. This is particularly important in a multi-disciplinary field such as DM. The practical testing of the opportunities offered by the sMLe interpretation of MaxEnt proposed by Halvorsen 2013 requires just such a detailed consideration. Whilst intuitively resonant with ecologists, implementing this approach poses practical challenges that can be made more accessible by a drawing on simplified theoretical informatics concepts also covered in this section.

All the papers presented here also draw from the following theoretical and statistical concepts: principles of parsimony as they apply to model selection (Legendre & Legendre 2012), ecological gradient analytic perspectives (Whittaker 1967; Ter Braak & Prentice 1988; Halvorsen 2012), object orientation and operational workflows (Jørgensen 1993; Petzoldt & Rinke 2007), distribution modelling practice in general (Austin 2002; Franklin 2009; Peterson et al. 2011; Halvorsen 2012), maximum entropy (Jaynes 1957) and maximum likelihood principles (Pawitan 2001; Plant 2012; Sokal & Rohlf 2012).

A variety of analytical and computational tools were used throughout. See the empirical section, for more details on these.

Opening up the theoretical considerations for practitioners (aim 1)

PAPER 1 reviews the theoretical basis of the non-parametric Maximum Entropy method and provides a simplified mathematical derivation of the more statistically familiar sMLe provided by Halvorsen (2013). This work draws on gradient analytical perspectives and is intended to provide a more ecologically intuitive understanding of the statistical of an otherwise less accessible machine learning modelling approach. A real practical example of how to implement this new interpretation is worked out in details (PAPER 1), and opens up, at least theoretically, the entire MaxEnt modelling practice to a broader range of opportunities. The new options offered by this approach also include incorporation of more user control of the variable transformation and selection process; improved variable contribution measures and options for variation partitioning; and improved output prediction formats (see PAPERS 1 and 2).

The MaxEnt principle (Jaynes 1957), laid out by Phillips et al. (2006) enables you to estimate a target probability by finding the probability distribution that is most spread out, or closest to uniform, or of maximum entropy (hence the name MaxEnt) given a set of constraints (in DM typically represented by a set environmental variables, recorded for a set presence and a set of background observations). Della Pietra, Della Pietra and Lafferty (1997) demonstrated that the best estimates for the MaxEnt distribution can be obtained by parameterisation of a Gibbs function. Recently, Halvorsen (2013) has shown that the MaxEnt model can also be derived by principles of strict Maximum Likelihood Estimation (sMLe), and Renner and Warton (2013) have demonstrated a close relationship of MaxEnt to Poisson point processes. In the context of DM, Maximum likelihood estimation implies identifying the model that maximises the likelihood of the observations, given a species set of conditions

(Hastie & Fithian 2013). In PAPER 1 we thus describe in practical terms MaxEnt as an sMLe method, to an audience of ecologists currently using maxent.jar software in their distribution modelling research. Furthermore, we show how the sMLe explanation of MaxEnt opens for more user control over the entire modelling process, from transformation of explanatory variables, via model selection, to model assessment and evaluation.

Drawing on more standard statistical methods, such as those offered by GLM, the approach that we present opens up MaxEnt modelling to a broader range of statistical tools and options well known to ecologists. Theoretically “opening up” the core elements of the modelling process we first suggest decoupling the model selection and parameterisation process, at all levels, starting with variable selection itself. This contrasts the *black box* approach employed by the lasso penalty approach whereby the model selection and parameterisation procedures are fixed together into a single shrinkage ℓ_1 -regularisation term and the user has limited control, particularly over the individual variable selection process.

Rather than the *standard* iterative process of pre-variable selection and “shrinking” the final models to the desired level of model complexity, the *alternative* approach we propose is to make use of the subset stepwise selection procedure, which allows us to separate model selection from model improvement criteria, and, through an iterative process of nested model comparison, build models of increasing levels of complexity (Reineking & Schröder 2006; Halvorsen 2013). The log-loss interpretation of MaxEnt, allows us to measure and compare these nested models in terms of *i*. Variation Accounted for (equivalent to Phillips, Anderson and Schapire (2006)’s “gain”), *ii*. Residual Variation (Variation not accounted for), and *iii*. Fraction of the Variation Accounted for, in other words explained by the model. We can then use these statistics against a pre-set internal model performance assessment criterion, such as the test significance level α (Halvorsen 2012) to make a statistical decision as to whether to accept or reject the null hypothesis, stating that the model of increased complexity does not *significantly* improve the predictions of the modelled. This approach can be applied at different levels of complexity, starting with improved control of single variable transformation and selection, to multi-variable models with or without considering interactions between variables and different levels of model strictness (see the empirical section for examples of this).

Thus, prior to the MaxEnt modelling itself, all continuous explanatory variables may be transformed to one of eight classes of derived variables [linear (L), monotonic (M), quadratic (Q), deviation (D) splines: forward and reverse hinges (FH and RH), and threshold (T), and categorical variables into binary ones (B)]. These roughly correspond to Maxent.jar’s *features* (Phillips 2010), which are, on the other hand, generated at once, in full modelling process, and users have limited control on which are selected in the final MaxEnt model (Table 1 and Table 3: Module 2).

Table 1. Transformation of explanatory variables (EVs) into derived variables (DV): DV main types (DVMs) and types (DVTs) relevant for MaxEnt modelling. Transformation is carried out in two steps, of which only the first step, transformation into ‘raw’ derived variables (rDVs) X_k' , is shown in the rightmost column of the table. The proper DVs X_k are obtained by linear ranking of rDVs onto a [0,1] scale. * = DVTs not currently implemented in Maxent.jar. Source PAPER 1: Appendix 3

DVMT	DVT		Description	Interpretation	Transformation function for DVs
	Code	Term			
continuous	L	Linear	the continuous EV Z_j itself	models the response to the EV itself	$x_{ik}' = h_L(z_{ij}) = z_{ij}$
continuous	M	Monotonous	a monotonous, continuous transformation of the continuous EV Z_j	models the response to a nonlinear transformation of the EV; quadratic (Q) variable obtained as the square of Z_j is a special case	$x_{ik} = h_M(z_{ij}) = f(z_{ij})$ where f is a continuous function
continuous	D*	Deviation	the continuous EV Z_j , centred on the mean for observed presence grid cells, raised to the power a	takes the tolerance of the species with respect to the EV explicitly into account by modelling the response to the spread of z_{ij} around the mean value for observed presence grid cells, $\bar{z}_j^*$; the V (variance) variable, which is obtained for $a = 2$, is a special case	$x_{ik}' = h_D(z_{ij}) = z_{ij} - \bar{z}_j^* ^a$
spline	HF	Forward hinge	a continuous EV Z_j transformed to a linear spline of order two	models the response to a piecewise linear spline with one knot (the point z_{0j}) above which X_k is a linear function of Z_j and below which X_k is set equal to 0	$x_{ik}' = h_{HF}(z_{ij}) = \begin{cases} 0 & \text{if } z_{ij} < z_{0j} \\ \frac{z_{ij} - z_{0j}}{\max(z_{ij}) - z_{0j}} & \text{if } z_{ij} \geq z_{0j} \end{cases}$
spline	HR	Reverse hinge	a continuous EV Z_j transformed to a linear spline of order two	models the response to a piecewise linear spline with one knot (the point z_{0j}) below which X_k is a linear function of Z_j and above which X_k is set equal to 0	$x_{ik}' = h_{HR}(z_{ij}) = \begin{cases} \frac{z_{0j} - z_{ij}}{z_{0j} - \min(z_{ij})} & \text{if } z_{ij} \leq z_{0j} \\ 0 & \text{if } z_{ij} > z_{0j} \end{cases}$
spline	T	Threshold	binary transformation of a continuous EV Z_j	piecewise constant spline with one knot (discontinuity point z_{0j}) below which X_k is set equal to 0 and above which $X_k = 1$; models the proportion (frequency) of presence grid cells with $z_{ij} \geq z_{0j}$	$x_{ik}' = h_T(z_{ij}) = \begin{cases} 1 & \text{if } z_{ij} \geq z_{0j} \\ 0 & \text{if } z_{ij} < z_{0j} \end{cases}$
interaction	I	Interaction	the product of two continuous EVs Z_j and Z_v	models the response to the product of two continuous DVs	$x_{ik}' = h_P(z_{ij}, z_{iv}) = z_{ij} \cdot z_{iv}$
binary set	B	Binary	m_j binary DVs, one for each factor level u of a categorical EV Z_j ; each DV expresses if factor level u is recorded in cell i or not	models the proportion (frequency) of presence grid cells for each factor level u	m_j binary DVs, one for each factor level u : $x_{ik} = h_C(z_{ij}) = \begin{cases} 1 & \text{if } z_{ij} = u \\ 0 & \text{if } z_{ij} \neq u \end{cases}$

The iterative process of model selection and parameterisation, and additional control on variable selection and transformation, opened up by the ML explanation of MaxEnt, enables more comparability between models, as well as a closer match between the properties of the data and the model output.

Finally, the probability-ratio output (PRO) for representing model probability is derived and presented as more “assumption free” alternative to the default logistic format (Halvorsen 2013). This is because it does not rely on knowledge of the modelled target’s prevalence. This is obtained by multiplying the raw model output by the number of background points (N) used in the model. Being a ratio and having a mean value of roughly 1, it makes all models more comparable, regardless of number of background points considered. Figure 1, shows how PRO values from 5 different models compare.

Figure 1. Map representations of predictions from five MaxEnt models for *Scorzonera humilis* in SE Østfold, SE Norway, using the PRO (probability-ratio output). The five models are: Model 1 = the model with 1 EV (DIR); Model 2 = the model with 2 EVs (DIR + DIA); Model 3 = the model with 3 EVs (DIR + DIA + ALT); and Model 4 = the model with 4 EVs (DIR + DIA + ALT + SOE); Model 5 = the final model with 4 EVs and interactions between three pairs of EVs. Insert maps show mapped predictions for the small area. Source PAPER 1's supplementary materials.

These options allow users not only a more intuitive understanding of the ecological implications of the models being produced, but also an ability to trace and understand the theoretical and practical implications entailed in building these models. Using general principles of parsimony, whereby simplest models that explain the most variation are best and applying gradient analytical ecological theory, we can better assess and interpret the ecological processes we're modelling. Theoretically, most variation in nature is explained by a few complex gradients, and by comparing our theoretical with the observed modelled target's response curve, and we can better distinguish if these contributions are due to specific properties of the data model, the ecological model or the statistical method itself.

We extend this concept of opening up the model selection process and the mathematical understanding of MaxEnt, to the practice of DM in general, bringing in also more explicitly on informatics theory and gradient analytic perspectives.

Proposing a flexible modelling practice (aim 2)

Halvorsen (2012) presents a detailed gradient analytic perspective on DM and lays it out as a theoretical 12+ step procedure. These are grouped into Austin (2002) three main components, the ecological, data and statistical models. Theoretically, reshuffling the contents of an object (conceptually or physically) can result in novel access and understanding (Beck 2000), taking care that the broader context or purpose is not lost. Drawing on informatics concepts such as object orientation and computational workflows (Silvert 1993; Barseghian et al. 2010), in PAPER 2, I open up these steps and components, extract the right level of detail, and reassemble them into conceptual and operational modules that guide (and carry out) the core elements of distribution modelling practice more intuitively and computationally efficiently, giving users more practical control and automation (Figure 2). Additionally, Table 2 has been included to demonstrate how this conceptual and practical reshuffling also relates to other key literature in this field.

This Modular flexible Integrated component-based modelling practice arose as a direct result of “opening up” the theoretical opportunities of the strict Maximum Likelihood explanation of the MaxEnt model. As such, much of the framework and flexible toolbox revolve around this more ecologically intuitive, alternative MaxEnt practice (aMp), and comparing it (where relevant) to the existing default practice (dMp) of the options offered by the maxent.jar software. The aMp procedure was performed by applying a forward stepwise variable-selection procedure, using an F-ratio test (Halvorsen 2013) to compare the models and a significance alpha value (α), and generates a full trail of models, for different levels of model complexity. These are in contrast to the shrinkage default (dMp) method offered by the Maxent.jar software, whereby model selection and parameterisation are fixed by ℓ_1 -regularisation. Please see the previous section for details on how these options were derived, and the next one for details on the practical toolbox.

In the full context of the MaxEnt modelling practice, we present how this added flexibility and control, such as separating the model selection criteria from the model improvement criteria, for example, improves our theoretical understanding of the underlying ecological process. This flexible functional oriented practice entails 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 is practical and accessible to a broad range of users and purposes.

The framework balances requirements for overview of the components of the DM process (Figure 2 and 3), whilst at the same time guiding the modellers explicitly through the practical details of implementing the particular choices required by the model selection or data component selected. This flexible framework can be both a conceptual (descriptive) and a practical (computationally) guide throughout the practice of DM. In the next section, I present how this flexible practice is implemented as a modular, integrated, open and flexible toolbox.

Figure 2. Relationships between the 12 steps of the theoretical distribution modelling (DM) process recognised by Halvorsen (2012) (left) and the five modules (some of) these steps are reorganised into for practical DM according to the MIA framework. Dots with similar colours are grouped together in an MIA module. Steps indicated by red font are not recognised by Halvorsen (2012). 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. Source *PAPER 2: Fig.1*

Modular Integrated Approach	Halvorsen 2012 (Figure 8)	Peterson et al. 2011 (Figure 4.1)	Franklin 2009 (Figures 1.1, 1.2, 10.1)	Guisan and Zimmermann 2000 (Figure 1)
Module 1: Specification of the data model, statistical model and modelling tools, settings or parameters	<ul style="list-style-type: none"> Steps 1-5i, 6: Problem formulation, specification and conceptualisation, spatial overlay of species and environmental data Step 7 ii: Model specification Step 8ii: <i>Specification of internal model performance assessment criteria</i> 	<p><i>Set up methods from information found in:</i></p> <ul style="list-style-type: none"> Step 1: Species and Environmental Data Step 2: Niche modelling and model calibration 	<ul style="list-style-type: none"> Choose statistical model and parameters Species locations and environmental data matrix 	<ul style="list-style-type: none"> Conceptual model/theory Statistical model formulation: choice of (i) a suited algorithm for predicting a particular type of response variable and estimating the model coefficients, and (ii) an optimal statistical approach with regard to the modelling context.
Module 2: Preparing all derived variables from individual explanatory variables, including visualising distribution and frequency of presence curves	<ul style="list-style-type: none"> Step 5ii: Transformation of explanatory variables into derived variables 	<ul style="list-style-type: none"> Step 1: Data preparation <ul style="list-style-type: none"> species Process Environmental layers to generate predictor variables 	<ul style="list-style-type: none"> Species data selection Environmental data transformations 	<ul style="list-style-type: none"> Preparing species and environmental data <ul style="list-style-type: none"> All possible variable transformation, through primary identification of species' response curves, polynomial terms, smoothed empirical functions etc.
Module 3: Iterative process of model selection and parameterisation at different levels of model complexity. Model improvement and selection criteria specified in the first module	<ul style="list-style-type: none"> Steps 7,ii, 8,i-iii Model specification and modelling of the overall ecological response Step 8i: Model selection Step 8ii: Internal model performance assessment Step 8iii: Variable contribution to model 	<ul style="list-style-type: none"> Step 2: Niche modelling and model calibration <ul style="list-style-type: none"> a. Apply ecological statistical model algorithm b. Model calibration 	<ul style="list-style-type: none"> Model fitting or estimation (<i>based on these from above</i>): <ul style="list-style-type: none"> Choose model type Variable selection Response functions Model realization(s) 	<ul style="list-style-type: none"> Implementation of suitable statistical model algorithm through: <ul style="list-style-type: none"> Model construction: selection of explanatory variable, and Model calibration: estimation and adjustment of model parameters and constants to improve the agreement between model output and a data set
Module 4: Model post processing. Synthesize, extract and customise outputs and results of the modelling process	<ul style="list-style-type: none"> Step 8iv: Interpretation and transformation of model predictions 	<ul style="list-style-type: none"> Step 3a. Model projection into current geographic space. (<i>Mostly unaddressed</i>) 	<ul style="list-style-type: none"> Model validation: <ul style="list-style-type: none"> Accuracy measures Evaluate response functions Residuals, plot, map 	<ul style="list-style-type: none"> Model prediction/cartographic representations of: <ol style="list-style-type: none"> probability of occurrence, the most probable abundance, the predicted occurrence based on non-probabilistic metrics or the most probable entity
Module 5: Model evaluation	<ul style="list-style-type: none"> Step 11: Model evaluation by independently collected data 	<ul style="list-style-type: none"> Step 3b. Evaluation of model performance by independent data or data splitting 	<ul style="list-style-type: none"> Prediction map evaluation: <ul style="list-style-type: none"> Validation by expert Intermodal comparison 	<ul style="list-style-type: none"> Model evaluation <ul style="list-style-type: none"> Data splitting: Cross validation, Jack-knife or bootstrap Independent evaluation data

Table 2. Rough correspondence between the Modular Integrated Approach and key works in the literature of Distribution Modelling. Edits to text from each corresponding source is kept to a minimum, so as to ensure original meaning and use. Additional comments italicised and in parentheses.

Table 3. Descriptive overview and guide of the **MIA Toolbox**, a practical workflow wrapper for flexible implementation of the sMLE of MaxEnt. *From PAPER 2: Table 1.*

MIA Module	
Module Core Components	Components Detailed Description
Module 1: <i>Specification of the data model, statistical model and modelling tools, settings or parameters</i>	
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: <i>Visualisation of EV properties (FoP curves) and preparation of derived variables (features)</i>	
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 skewness 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: <i>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 in module 2. This module generates a full trail of models and organises results optimising traceability and interpretability.</i>	
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 <i>Standard Maxent</i> 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: <i>Post Processing. Synthesize, extract and customise model outputs/results</i>	
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 selected variables and models.
Module 5: <i>Model evaluation by independently collected presence/absence evaluation data</i>	
M5: Model evaluation	Spatial overlay of presence/absence evaluation data over raw predicted values from selected MaxEnt models, to calculate test AUC.

Accessible flexible toolbox for practical implementation (aim 3)

At the core of this theoretically simpler modelling practice are statistical techniques (manual forward selection) and complex processes and information produced (trail of models, model selection and parameterisation) that can be quite laborious and impractical to implement. This can greatly hinder its full exploration, not only in terms of actual testing of the proposed options to test, but also in terms of applied feedback and further developments of this novel theoretical practice. Thus, an equally open and intuitive informatics approach that can be scaled to integrate the different components being proposed by this practice is needed. With properties such as encapsulation, polymorphism and inheritance, object orientation and functional programming enable just such flexible regrouping of the components without sacrificing traceability and computational efficiency (Cushing *et al.* 2007; Matloff 2011; Bentlage & Shcheglovitova 2012).

The object oriented and computational workflow approaches are used to fully automate the alternative procedures and provide a practical guideline in its implementation. These are also simplified to more closely match the needs of the broad range of existing MaxEnt users and applications. Thus, in the context of this thesis, the Modular, Integrated, Approach Toolbox (MIAT) refers to both the descriptive guidance of the process, as well as the actual computational tools currently developed. See Table 3 for the descriptive guidance to the fully documented scripts also included as supplementary material. Though we exemplify this only in the context of MaxEnt modelling, this flexible toolbox and practice can also be applied to other statistical and modelling methods (Table 2).

PAPER 2 presents the object-oriented and computational workflow approach and how it is integrated with ecological, statistical and modelling theory in order to handle the complexity associated with the full modelling process in a practical way. MIAT objects (variables, functions, scripts, models, results, etc.) have been defined according to specific properties and modelling parameters. MIAT properties (e.g., identities and content) can be interactively assigned or inherited between objects and new objects created in a flexible and automated way. This concept of inheritance is taken a step further by coding the names of each object themselves in a way that enables users to more easily trace their origins. This embedded metadata approach is employed throughout the framework and resulting toolbox. An example of this is renaming and grouping all models results files (both those generated uniquely by the MIAT toolbox, as well as Maxent.jar) in different ways. First by replacing the first part of the file name with a code that combines key elements of the data properties, model selection and level combined. Secondly by changing the three letter suffix, making them more accessible to other analysis tools (such as other database or spatial analysis tools). Finally, by copying them and regrouping them in the traceable folders (Figure 3).

A. In the module and file structure

B. In the component and file name

A01 (user defined)
 M3A (module)
 FinPredData = Final Predictor Data
 Dtfrm = Dataframe or List
 WI = With Interactions
 RV1= Response Variable 1
 A12= Model selection criteria (λ_2)
 P03= Number of predictors

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

Figure 3: Hierarchical nested modularity at different levels. Objects' (modules, scripts, components, files, etc) structure and names that reflect their source, relationship, purpose, 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. *Source PAPER 2: Fig. 3.*

Through its Data or Model Parameter Tracking table (Figure 4), the MIAT toolbox enables access to any of these files (resulting from both the maxent.jar and the entire process), with minimal user input. Interactive user input is required in the first module to define core initial properties. Thereafter the process of renaming files and objects in a way that reflect key properties of the object at hand, and makes them more accessible is automated by the toolbox. The coding is kept simple and presented in the form of detailed scripts rather than higher level functions, to match the accessibility and flexibility of the framework and maintain the explicit link to its theoretical source (Adler 2010; Matloff 2011; Teetor 2011; Van der Loo 2012).

Network ▶ Pcnhm329 ▶ D ▶ IDA ▶ A03_RanKA ▶ AM ▶

Burn New folder

Name	Date modified	Type	Size
M2g_All	22.05.2014 14:08	File folder	
M2h_FullSetDV	22.05.2014 14:10	File folder	
M3A_All	22.05.2014 14:44	File folder	
M3B_All	22.05.2014 14:53	File folder	
M3C_All	22.05.2014 15:25	File folder	
M4_All	26.05.2014 12:12	File folder	
M5_All	26.05.2014 12:15	File folder	
A03_M4b_MaxModPar_AM_ModName1st...	26.05.2014 12:12	Microsoft Excel Co...	2 KB
A03_M4sm_MaxModPar_SM_Dtfrm_1st.csv	26.05.2014 12:15	Microsoft Excel Co...	1 KB
A03_M4sm_MaxModPar_SM_Dtfrm_all.csv	26.05.2014 12:15	Microsoft Excel Co...	2 KB
A03_M5a_FD_AUC_ModAll.csv	26.05.2014 12:15	Microsoft Excel Co...	1 KB
A03_ResAll.xlsx	26.05.2014 12:23	Microsoft Excel W...	19 KB
sMaxRes_SM_All.csv	26.05.2014 12:13	Microsoft Excel Co...	14 KB

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	ModName	DAC	RV_Num	SelMeth	Lambda	Alpha	FTCode	DVs	EVs	IEV	FTVE_PV	MaxRes_AITest_AUC	MaxRes_n	numBackgr	GPName		
2	A03_aNIRV1AI1P06	A03_	1 a		0	0.05 aWl		58	13	6	0.542178	0.9774	0	54	10054	DEM10xSLOPE DEM10x	
3	A03_aNIRV1AI2P07	A03_	1 a		0	0.05 aWl		26	7	0	0.481758	0.9713	0	54	10054	ATIL DEM10 JORDART	
4	A03_aNIRV1AI2P04	A03_	1 a		0	1.00E-06 aWl		21	9	4	0.481203	0.9706	0	54	10054	DEM10xASPECT_ASPECT	
5	A03_aNIRV1AI2P03	A03_	1 a		0	1.00E-06 aWl		14	6	0	0.464233	0.9677	0	54	10054	ATIL DEM10_ASPECT	
6	A03_aNIRV1AI2P04	A03_	1 a		0	1.00E-06 aWl		11	4	0	0.453461	0.9636	0	54	10054	ATIL DEM10_ASPECT	
7	A03_aNIRV1AI2P03	A03_	1 a		0	1.00E-06 aWl		7	3	0	0.421721	0.9463	0	54	10054	ATIL DEM10_ASPECT	
8	A03_aNIRV1AI2P02	A03_	1 a		0	1.00E-06 aWl		5	2	0	0.40387	0.9325	0	54	10054	ATIL DEM10	
9	A03_aNIRV1AI2P01	A03_	1 a		0	1.00E-06 aWl		3	1	0	0.345957	0.9056	0	54	10054	ATIL	

« Pcnhm329 ▶ D ▶ IDA ▶ A03_RanKA ▶ AM ▶ M4_All ▶ Proj_CSV

Open Print Burn New folder

GP04

GrPr04pv1

M4_All

HTML

Lambda

ModPar

Pred_ASC

maxent.cache

Pred_ASCPRO

Pred_CSV

Proj_CSV

ProjData_SWD

Res_Max

Name	Date modified
aNIRV1AI1P07ProjPred.csv	22.05.2014 14:4
aNIRV1AI2P01ProjPred.csv	22.05.2014 14:5
aNIRV1AI2P02ProjPred.csv	22.05.2014 14:5
aNIRV1AI2P03ProjPred.csv	22.05.2014 14:5
aNIRV1AI2P04ProjPred.csv	22.05.2014 14:5
aNIRV1AI2P06ProjPred.csv	22.05.2014 14:5
aWIRV1AI1P06ProjPred.csv	22.05.2014 15:2
aWIRV1AI2P04ProjPred.csv	22.05.2014 15:2

Figure 4: Model Parameter Tracking table (highlighted), which contains detailed information of model characteristics extracted from module 3 internal R objects (i.e. type and strictness of model selection criteria, training and background data parameters), and maxent.jar results files (lambda, maxent.res) produced. Final models selected in module 4 get collated into one folder, using the model tracking table and the script. This table is also accessible manually via other spreadsheet tools.

MIAT now covers a range of options and settings for the maximum likelihood implementation of MaxEnt and provides flexible guidance of users through the DM process. Through the use of the alternative interpretation of MaxEnt and the Flexible Framework and Toolbox built to implement it, a trail of models of increasing complexity can be built to suit different modelling purposes, which further enhances traceability and interpretability.

This accessibility is exemplified through a series of descriptive vignettes (PAPER 2: Appendix 2), where the key elements of the MIAT toolbox, and how they were developed, are presented in a very accessible manner. Vignettes 15-18 provide nice examples of the flexibility and integrated traceability that mirror and guide the theoretical foundations of the full modelling process.

Additional specifications were implemented into the MIAT toolbox and used to automate the practical parameterisation of all the models presented in the next section (Empirical exploration of the practice), as well as guide the analysis and testing of two novel frameworks to tackle potential sampling bias, presented in PAPER 3. These two frameworks entail themselves a dynamic interplay between theory and empirical data, which, despite the automation in producing the results, the analysis still relies on careful inspection and assessment on the part of the practitioner. The ability to trace and interpret the vast number of model results, achieved in a guided manner through the toolbox, is thus a critical part of this resulting practice.

Empirical exploration of the practice of Distribution Modelling by MaxEnt

PAPERS 3-6 are all empirical explorations of the proposed *Flexible, Integrated Modular* MaxEnt *practice*, with each paper addressing specific aspects of the previously stated aims:

- Aim 4: The study objects and basic dataset properties (aim 4).
- Aim 5: Detecting and mitigating potential sampling bias in presence-only data.
- Aim 6: Effects of spatial autocorrelation in the response variable.
- Aim 7: Model selection method, including statistical and practical options and settings, and corresponding methods for variable selection.
- Aim 8: Relating model complexity, performance and modelling purpose for overall model assessment and evaluation.

The theoretical and practical flexibility of the MIAT toolbox is used to guide this exploration. The simplicity and accessibility of the scripts make it also possible to interactively adjust them so as to fit the specific aim being explored, whilst maintaining the overall structure of the toolbox, thereby ensuring a common strategy of this approach. The conceptual and computational traceability, through the use of R-lists objects and the *embedded metadata concept*, amongst other things, leaves also a detailed trail of the process, making it possible to tackle many disparate questions at once, and maintaining a dynamic link between them. This proved particularly effective in the practical development and testing of a novel theoretical understanding of sampling bias in presence-only data (PAPER 3), where several conceptual and practical elements could be confidently explored at once, by a group of researchers with different expertise and knowledge.

Following the modular integration and flexibility of the MIAT practice, the Materials and Methods, Results sections of each of these papers will be grouped and explored under the specific empirical aims they address, and beginning with the general specification of each modelling components, add increasing level of detail as required.

Materials and Methods

A range of desktop and field mapping tools, notably ESRI's ArcGIS Desktop and Mobile v.9.3 to 10.1 (Anonymous 2012). Furthermore I made extensive use of several programming and data analysis tools such R (R Core Team 2014), including several R-packages (see Appendix 2) and R-Studio v.0.99.473 to develop, document and test the MIAT toolbox. The current version of the toolbox makes use of several DOS/Windows' shell operating system commands, accessed via the MS-DOS command prompt window, either directly from the R scripts, or via the executable batch text files (*.bat*). These are equivalent to Unix/Linux's Shell commands and Bash Scripts. Though less familiar to Windows' users, use of this command-line interface is a very powerful way to simplify and automate repetitive tasks, including creating pipelines between files and other tools (Phillips 2010). The maxent.jar software, version 3.3.3.k, (Phillips 2010) is used throughout.

The study objects and basic dataset properties (aim 4)

Distribution modelling can be thought of quite simply as estimating the spatial distribution of a given taxon by correlating its occurrence to the environmental information about this taxon. This can be described in the form of the mathematical relationship between the dependent variable (modelled target/response variable) and the independent variables (explanatory variables). The statistical assessment of the validity of this model entails considering all the properties of this relationship. The study object(s) of distribution modelling are the modelled target(s) and the area in which the modelled target is studied. These are represented (described) by an empirical dataset which is what is actually subjected to distribution modelling, containing a response variable and a set of (mostly environmental) explanatory variables. Since the empirical dataset results from filtering of the properties of the study object(s) through a sampling procedure (Økland 2007; Halvorsen 2012), the basic dataset properties such as rasterization of the study area, determination of extent, grain, choice of, and sampling of, variables are addressed separately in the different aims. Though listed separately, these dataset properties, and the properties of the study objects are collectively referred to as “*data model object*” in the MIAT framework and toolbox (Figure 2 and Table 3).

Modelled Targets and Response Variables

The identity of the modelled target can be a species, a habitat or nature type (Stokland *et al.* 2008; Mazzoni *et al.* 2011), or the genetic lineage of a particular taxon. The target for modelling can be represented by more than one variable, e.g., as the result of sampling by different strategies (see PAPER 5), using different representations of the study area (grains and extents) (Figure 5). These choices are often made (but not necessarily) in conjunction with determination of modelling purpose (Thibaud *et al.* 2014). Irrespective of modelling method or modelling purpose (see final aim for more on this), it is important to have a sense of the ecological requirements of the modelled target when making initial decisions about which other datasets to access and properties of the datasets to specify (Guisan *et al.* 2006; Burger & Page 2007; Elith & Leathwick 2009; Swab *et al.* 2012).

Figure 5. The map on the left shows the same modelled target (semi-natural grassland) represented as six different variables (from top left to bottom right): as points derived from the mapped polygons of the nature-type (for three different grain sizes), or as points derived from presence occurrences of three different selected indicator species. Field images of the same modelled target are shown on the right (top to bottom): as nature types (semi-natural grasslands) or as indicator species (*Gentianella campestris* and *Scorzonera humilis*). Study area: Oppdal, Sør Trøndelag county, Norway.

Nine species belonging to three different insect orders were used as modelled targets in PAPER 3. Each response variable comprised of geo-referenced presence records over mainland Norway, with a precision of ± 1000 m or better, extracted from the online GBIF datasets. Multiple records for the same species from the same grid cell were treated as one presence. After inspection of their theoretical frequency of presence curves (FoP) and taking information about the species ecology and distribution into account (Ahti, Hämet-Ahti & Jalas 1968; Hansen & Larsson 1973; Ehnström & Holmer 2007; Aarvik, Hansen & Kononenko 2009; Bartsch *et al.* 2009), these were classified according to the “CURS” categories (Collins, Glenn & Roberts 1993) – core, urban, rural, and satellite species. See Figure 6 in the thesis for and more details on each species (Table 4 and Figure 6).

Table 4. Species included in the example dataset: numbers of presence observations and distribution in Norway. CURS category refers to the classification of distributions into core, urban, rural and satellite according to Collins et al. (1993), ‘intermediate’ indicates that the species in question does not fit into any of the four categories. Source PAPER 3: Table 2.

Species	Family (BTG)	Order	No. of obs.	No. of obs. BTG	Distribution in Norway	CURS category	Trophic level
<i>Meligethes aeneus</i>	Nitidulidae	Coleoptera	60	443	As far north as Troms county	Intermediate	Herbivore
<i>Dioctria hyalipennis</i>	Asilidae	Diptera	95	375	Mostly southeast Norway	Intermediate	Predator
<i>Glaucopsyche alexis</i>	Lycaenidae	Lepidoptera	84	1541	South-east Norway along the coast	Satellite	Herbivore
<i>Rhagium mordax</i>	Cerambycidae	Coleoptera	111	1228	The whole country, except the high north	Core	Herbivore
<i>Volucella bombylans</i>	Syrphidae	Diptera	103	598	The whole country below the tree line	Core	Adults herbivore, larvae scavenger
<i>Parnassius apollo</i>	Papilionidae	Lepidoptera	88	211	The inland of southeast Norway	Urban	Herbivore
<i>Rhagonycha limbata</i>	Cantharidae	Coleoptera	113	312	The whole country below the tree line	Core	Predator
<i>Eristalis intricaria</i>	Syrphidae	Diptera	194	598	The whole country below the tree line	Core	Adults herbivore, larvae detritivore
<i>Pieris napi</i>	Pieridae	Lepidoptera	346	926	The whole country	Core	Herbivore

Figure 6. Theoretical frequency-of-presence curves for typical core (a–b), rural (c–d), urban (e–f) and satellite (g–h) species. The species' aggregated performance (given on the ordinate axes) is the relative frequency of a species' presence along the environmental gradient, i.e., ratios between counts or densities of presence and background observations for successive intervals or at successive points along the gradient. Curves are shown for broad (a, c, e, g) and fine (b, d, f, h) spatial scales, i.e., for broad (e.g., 100 km²) and fine (e.g., 100 m²) grid-cell rasterizations of the study area, respectively

In PAPER 4, three different response variables were derived for the rare lichen *Staurolemma omphalarioides* (Anzi) P.M. Jørgensen & Henssen. The mostly lowland and coastal lichen has a very broad and disjunct distribution, prompting questions about whether the species should actually be split into two distinct lineages that have been obtained through a phylogenetic analysis of selected samples (PAPER 4: Fig. 1). Thus, three modelled targets were devised to “test the null hypothesis that the two lineages did not differ with respect to distribution” (and environmental correlates of distribution), and thus contribute significant information for its conservation efforts. The response variable representations of the three modelled targets were as follows: (1) the full set of records to represent the species, $n=55$, and (2, 3) each of the Northern, $n=8$ and the Mediterranean, $n=11$, to represent the intraspecific lineages.

In PAPER 5 the presence of swamp-forests is set as the modelled target. Nutrient rich boreal swamp forests are of considerable conservation interest (Ohlson *et al.* 1997; Hörnberg *et al.* 1998). Forests larger than 100 m² were systematically searched in the Østmarka Nature Reserve (located in SE Norway, 59°50′ N, 11° 02′ E, 190–368 m.a.s.l.), geo-referenced by use of a handheld Garmin eTrex H Global Positioning System (GPS) receiver and plotted manually onto a topographic map at scale 1:1000 in the field (Dirksen 2013). These were then digitised as polygons into a geographical information system project, using ArcMap 9.3 (Anonymous 2008). Two types of response variables were obtained from these polygons (PAPER 5: Fig. 1). These differed in the number of presence observations considered: RV1 ($n=121$) as the centroid of each mapped swamp forest polygon, RV2 ($n=7175$) as the centre point of each 5-m pixel, resulting from the rasterization of the same polygons.

Twenty-three different modelled targets, recorded in 14 study areas, were used in PAPER 6. This diverse set included 19 species (12 plants, 5 fungi, 1 animal and 1 brown alga) and four nature types (semi-natural grasslands, types of forest). The study areas varied greatly in extents (from 5.8 km² to more than 100,000 km²), grain size (from 5 m² to 1,000 m²), and number of background cells used in the modelling. All except two of the modelled targets were located in Norway: data set 20 was derived from a global study of fungal infections on wheat and data set 23 was collected in 3 Ethiopian alpine mountains (see Table 4 for more details, including full data reference material).

Study Area and Explanatory Variables

The first two axes of the PCA ordination of 54 climatic, topographical, hydrological and geological variables, rasterised as 1x1 km grid cells, provided by Bakkestuen, Erikstad and Halvorsen (2008), were used as source of the explanatory variables in PAPER 3, and for several datasets in PAPER 6. In their study, they show that these axes represent the continuous oceanicity-continentality (PCA1) and temperature (PCA2) gradients identified by Moen (1999) and the vegetation sections and zones by Ahti, Hämet-Ahti and Jalas (1968). The study extent encompassed mainland Norway, thereby making them even more suitable for the gradient analytic perspective of DM.

From these two gridded explanatory variables, 2 types of background data were prepared: Background Target Group (BTG) and Uninformed Background (UB). The former are obtained by selecting 10,000 random background points only from those grid cells where all species

belonging to the same family as the respective model target species have been detected (BTG). Uninformed background points will instead be randomly drawn from anywhere in the same study area (UB).

PAPER 4 used the 19 variables of the BioClim dataset (downloaded from the WorldClim database; <http://www.worldclim.org/>; Hijmans *et al.* (2005)), and three terrain variables – digital elevation model (SRTM DEM, Jarvis *et al.* 2008), slope and aspect. Slope (as degree of an angle) and aspect (as an eight-class categorical variable: N, NE, E, SE, S, SW, W, NW), were derived using ArcGIS 10's Spatial Analyst.

Since the ecology and current distribution of the species has been well documented, background observations were selected from an area that only included the known distribution of the species and a buffer (Stokland, Halvorsen & Støa 2011; Acevedo *et al.* 2012). With a spatial resolution (grain size) of 2.5 arcminutes (c. 5 km), and a coastal buffer of 1.7°, the study area comprised coastal Europe, including northwest Africa to the Cape Verde Islands, and Iceland. The total number of grid cells covering the study area was 387,533.

PAPER 5 used eleven terrain variables derived from a LiDAR-based digital elevation model (DEM) as the independent explanatory variables (PAPER 5: Table 1). The grain of the original DEM was set at 5x5 m, and the variables were processed according to procedures described in Dirksen (2013). The extent of the study area comprised of an 8 km² section of the Østmarka Nature Reserve (59°50' N, 11° 02' E, 190–368 m.a.s.l.), located in the southern boreal zone of SE Norway (Moen 1999). Whilst the area is dominated by main ridges and valleys in the N-S direction and the topography is broken with structures on several scales, the fine grain was chosen to address the fine-scaled spatial variation in presence or absence of swamp forests in the area. The total number of grid cells in the area was 233 774, including the training presence (cells entirely located in lakes were excluded). See PAPER 5: Fig. 1.

PAPER 6 include data and field work carried out by a wide range of research participants, with differing methodologies, tools and modelling purposes. The study areas varied greatly in both extent, from 5.8 km² (data set 3) to more than 100,000 km² (data set 23), grain size, from 5 m² (data set 3) to 1,000 m², and number of background cells used in the modelling. See TABLE 1 in PAPER 6 for a detailed list of each specific data reference material, study area and dataset properties, such as grain size, or types and number of variables used. All except two of study areas were located in Norway: data set 20 was derived from a global study of fungal infections on wheat and data set 23 was collected in 3 Ethiopian Alpine Mountains.

Table 4. Characteristics of the 23 response variables modelled in this study and the datasets used. # = Response variable number (from 1 to 23). Modelled target; type [A = Animalia; C = Chromalveolata (brown alga); F = Fungi; P = Plantae; N = Nature type]. Study area: Grain = resolution; # of grid cells=total number of grid cells, # of background grid cells = number of background cells. Explanatory variables (EVs): class(es) of EVs represented [LC = land cover, C = climate, T = topography (derived from DEM), LE = other local environmental variables (bedrock, quaternary deposits, etc.), RE = regional environmental gradients (other than climate)] and the number of continuous (cont.) and categorical (cat.) variables. Evaluation data: sampling scheme (RM = randomisation within model-based strata, R = random, S = systematic) and the number of observed presences and absences. *Source PAPER 6: Table 1.*

#	Modelled target	Study area	Conceptualisation of study area		Response variable	Explanatory variables (EVs)			Evaluation data			Reference/datasource		
			Grain (m)	# of grid cells		# of background cells	# of pres. obs.	Class	# of contin.	# of cat.	Sampling scheme		Grain (m)	# of pres.
1	N: Semi-natural grassland	Østfold county, SE Norway	500	17479	11059	1059	T, LC, LE, C	22	12	RM	500	46	76	(Norderhaug <i>et al.</i> 2012)
2	N: Semi-natural grassland	Oppdal municipality, C Norway	500	4096	4096	281	T, LC, LE, C	14	8	RM	500	32	50	(Mazzoni <i>et al.</i> 2011)
3	N: Swamp forests	Østmarka, Akershus county, SE Norway	5	233774	10115	115	T	11	0	RM	5	209	80	(Dirksen 2013)
4	P: <i>Knautia arvensis</i>	Levanger municipality, Nord-Trøndelag county, C Norway	10	205578	10054	54	LC, T, LE	5	3	RM	10	394	18	(Hovstad, Grenne & Warda 2012)
5	P: <i>Scorzonera humilis</i>	SE Østfold county, SE Norway	25	3462378	10456	465	LC, T, LE	7	1	RM	25	528	13	(Edvardsen, Bakkestuen & Halvorsen 2011b)
6	P: <i>Fallopia japonica</i>	Hordaland and Sogn og Fjordane counties, W Norway	1000	9025	9025	151	T, LC, C	9	0	RM	1000	93	30	(Auestad <i>et al.</i> 2012)
7	P: <i>Lupinus polyphyllus</i>	Hordaland and Sogn og Fjordane counties, W Norway	1000	9025	9025	114	T, LC, C	9	0	RM	1000	93	29	(Auestad <i>et al.</i> 2012)
8	P: <i>Bunias orientalis</i>	SE Norway	1000	2275	2275	234	T, LC, C	9	0	RM	1000	95	31	(Auestad <i>et al.</i> 2012)
9	P: <i>Fallopia japonica</i>	SE Norway	1000	2275	2275	91	T, LC, C	9	0	RM	1000	95	10	(Auestad <i>et al.</i> 2012)
10	P: <i>Lupinus polyphyllus</i>	SE Norway	1000	2275	2275	101	T, LC, C	9	0	RM	1000	95	44	(Auestad <i>et al.</i> 2012)
11	P: <i>Pastinaca sativa</i>	SE Norway	1000	2275	2275	122	T, LC, C	9	0	RM	1000	95	12	(Auestad <i>et al.</i> 2012)
12	P: <i>Solidago canadensis</i>	SE Norway	1000	2275	2275	381	T, LC, C	9	0	RM	1000	95	37	(Auestad <i>et al.</i> 2012)
13	C: <i>Saccharina latissima</i>	S Norwegian coastline	25	365626936	10474	474	T, LE	5	0	RM	25	260	463	(Bekkby & Moy 2011; Gundersen <i>et al.</i> 2012)
14	A: <i>Crassostrea gigas</i>	The Oslo fjord, SE Norway	25	1978108	10183	183	T, LE	17	1	RM	25	34	379	(Rinde <i>et al.</i> 2015)
15	F: <i>Nephroma bellum</i>	Parts of W Norway	500	35115	10021	21	RE	2	0	RV	N/A	8	110	Heegaard <i>et al.</i> (unpubl.)
16	F: <i>Nephroma laevigatum</i>	Parts of W Norway	500	35115	10060	60	RE	2	0	RV	N/A	31	48	Heegaard <i>et al.</i> (unpubl.)
17	F: <i>Nephroma parile</i>	Parts of W Norway	500	35115	10036	36	RE	2	0	RV	N/A	46	31	Heegaard <i>et al.</i> (unpubl.)
18	F: <i>Nephroma resupinatum</i>	Parts of W Norway	500	35115	10016	16	RE	2	0	RV	N/A	3	48	Heegaard <i>et al.</i> (unpubl.)
19	N: <i>Picea abies</i> forest	Vestfold county, SE Norway	16	9956114	10040	40	T, RE	8	0	S	250	200	156	(Breidenbach & Astrup 2012)
20	F: Infection of wheat by <i>Puccinia rust</i>	The World	N/A	4615	4615	430	C, RE	19	0	RM	N/A	552	2868	(Endresen <i>et al.</i> 2011)
21	P: <i>Actinos arvensis</i>	Inner Oslofjord area, SE Norway	25	56	10056	56	LC, C, T, LE	5	1	S	25	63	20	(Bakkestuen <i>et al.</i> 2014)
22	P: <i>Seseli libanotis</i>	Inner Oslofjord area, Norway	25	112	10112	112	LC, C, T, LE	5	1	S	25	66	20	(Bakkestuen <i>et al.</i> 2014)
23	P: <i>Lobelia rhynchopetalum</i>	Three Ethiopian Alpine Mount.	100	10913616	10025	25	T, LC, C	26	2	S	20	87	111	(Chala <i>et al.</i> Submitted)

Detection and mitigation of potential sampling bias (aim 5)

In the context of this thesis a presence data-set is said to contain sampling bias if “the frequency distribution of the presence observations along its major environmental complex-gradients deviates from the frequency distribution of the modelled target’s true presence in the environmental space”(PAPER 3) . Based on this definition and using MIAT toolbox, two different sampling bias frameworks were tested:

1. the framework for **detection** of sampling **bias** in presence-only data; and
2. the framework for assessment of potential **effects** of sampling **bias** on species distribution models in total (Figure 7).

These frameworks rest firmly on the gradient analytic perspective and applying this in a Distribution Modelling context as presented by Halvorsen (2012) and addressed in PAPER 1. Both require the user to compare a theoretical distribution of the sample data in question, against the actual observed distribution obtained by either the response of the training data itself (*framework 1, bias detection*), or of the response of the modelled results (*framework 2, bias effects*). After selecting a set of presence observations (as the response variable) and the major ecological gradients (background data) a series of graphs are plotted and potential bias is assessed both before and after the distribution modelling process. Broadly speaking, Theoretical Frequency of Presence (TFoP; see PAPER 3 for more details) curves, which are based upon known properties of the modelled target, are used as a reference with which Observed FoP curves (Frequency of Presence, based upon the training data) in framework 1, and Modelled FoP curves (showing the Predicted Relative Frequency of Presence, PRFP) in framework 2, are compared. This comparison is carried out by statistical methods or, alternatively, by visual inspection, of the resulting curves. Deviations of the observed FoP curve from the reference true FoP curve indicate potential presence of sampling bias. The practical application of the testing on the examples used in PAPER 3 was guided and automated by the MIAT framework and toolbox.

The main purpose of Ecological Response Modelling (ERM) is to determine the general ecological characteristics (explanatory variables) of what the modelled target (response) responds best to, in a theoretical sense. The purpose of Spatial Prediction Modelling (SPM) on the other hand, is to determine the patterns of spatio-temporal distribution of the modelled target, given the observed characteristics of the modelled target and its environmental information. Please refer to the last aim for more detailed description and discussion of modelling purpose.

For each of the nine species in PAPER 3, 2×2 MaxEnt models were generated: two by these two different modelling purposes (ERM and SPM) and two by types of Background Data (Uninformed Background and Background Target Group). The ERM models were specified using the manual stepwise forward selection of the aMp approach and the SPM by using the shrinkage selection of the dMp approach.

Figure 7. Proposed frameworks for assessment of sampling bias in presence-only data (L) and for assessment of effects on species distribution models (R). Source PAPER 3: Figure 1.

Effects of spatial autocorrelation in the response variable (aim 6)

The effects of spatial autocorrelation on modelling results are addressed in PAPER 5 by obtaining parallel MaxEnt Models for the different response variables representing the same modelled target. These differed in the number of presence observations considered: RV1 (n=121) as the centroid of each mapped swamp forest polygon, RV2 (n=7175) as the centre point of each 5-m pixel, resulting from the rasterization of the same polygons. A matrix of Kendall's rank correlation coefficients τ (Kendall 1938) between all eleven EV pairs is used to describe their correlation structure.

Model selection: importance of options and settings (aim 7)

Options for model selection (shrinkage selection in dMp and manual stepwise forward selection in aMp) and settings for these were explored by use of the MIAT toolbox.

In PAPER 3, for each of the nine species, and the two types of background, parallel aMp and dMp MaxEnt models were generated. These models were referred to in PAPER 3 as *Ecological Response Model (ERM)* and *Species Prediction Model (SPM)*, respectively, with reference to the different modelling purposes addressed.

In PAPER 4 we prepared separate ERM MaxEnt models for each of the three modelled targets, parameterising them using the aMp approach. Because of the lack of independent

test data, and the very low number of presence observations, we used a Monte Carlo randomisation approach to parameterise a null model of each modelled target. See model evaluation section for more on this. See PAPER 4's Appendix for more details and scripts.

In the case study contained in PAPER 5, in which swamp-forests in SE Norway were the modelled target, different levels of comparable options for the two model selection (dMp and aMp) methods were explored. Nine alpha levels ($\alpha = 0.25, 0.1, 0.05, 0.01, 1.0e-03, 1.0e-04, 1.0e-05, 1.0e-07, \text{ and } 1.0e-11$) for the F-ratio test were set for aMp, and six ℓ_1 -regularisation parameter values ($\lambda = 1, 2, 4, 8, 16, 32$) for dMp. For each of these model specifications, 2 levels of (model) complexity were considered, one allowing product variables to be considered (with interactions), and one with no interactions, for a total of 30 different models.

PAPER 6 contains a thorough examination of these two model selection methods (aMp and dMp), where 23 different Datasets (D) were used to parameterise four sets of comparable options of each: 2 levels of model strictness (S1= relaxed vs S2= very strict), 2 levels of variable complexity, allowing or not allowing product variables to be considered, (I1=no interactions, I2=Interactions). Strictness values for aMp and dMp respectively were: $\alpha = 0.05$ and $\lambda = 1$ (S1) and $\alpha = 1.0e-06$ and $\lambda = 16$ (S2). A total of 184 MaxEnt models were thus specified (Figure 8).

Model selection involves also deciding the type of variable transformation (derived variables) and complexity to choose (interactions). See earlier sections for more details on model specification. Default settings were chosen for dMp, varying only with respect to whether product variables (interactions) would be allowed or not. Unless otherwise specified, all models parameterised for ERM modelling purpose did not include interaction variables.

Relating model complexity, performance and modelling purpose (aim 8)

Several concepts and methods were used to achieve these aims.

Model Complexity

The first measure of model complexity was obtained by summing up the number of explanatory variables (EVs) and variables derived from these (derived variables; DVs) in each final MaxEnt model. The number of derived variables selected for each final model was read from the Maxent.jar output file *NN.lambdas* as the number of nonzero model parameters (Warren & Seifert 2011). Thus the choice of data selected to train the models is the first factor in determining model complexity.

After data selection, model complexity is also controlled by the properties of the model selection methods (options and settings) chosen for each model, including whether interactions between variables are considered (product variables). Strictness for aMp models was set by the α value in the F-ratio test, whereas the regularisation parameter λ was used to determine strictness of dMp models.

The MIAT toolbox was used extensively to create the full trail of models, of both the aMp and dMp. The interactive, simplified scripts were used also in measuring the degree of model complexity. Most objects, files and folders, were processed and renamed according to the embedded metadata concept introduced in the theoretical section, further guiding and automating the process (Fegraus et al. 2005; Leinfelder et al. 2010).

Simpler models will contain fewer parameters than more complex ones. However, the appropriate level of complexity can only be determined in relation to modelling performance and modelling purpose.

Modelling Purpose

In the context of this thesis two different modelling purposes are considered. Halvorsen 2012 distinguishes between Ecological Response Modelling (ERM) as “*distribution modelling with the main purpose of modelling the relationship between a target, typically the performance of a species, and a set of explanatory variables, to find and understand general patterns in the overall ecological response of the modelled target to the supplied explanatory variables*”, and Spatial Prediction Modelling (SPM), whose main purpose is to optimise “*the fit between model predictions and the true distribution of the modelled target’s performance in the study area in the time interval data were collected.*” (Halvorsen 2012: p. 39)

Thus ERM models should help us obtain the best representation of the overall ecological response in the environmental space, whereas the aim of SPM models is to obtain the best predictive performance of in geographical space (Halvorsen 2012).

In line with ecological theory, and given simpler models are less susceptible to the idiosyncrasies or bias of the data used to train the model, ERM models should be less complex than SPM models, irrespective of final model performance.

Overall model performance assessment and evaluation

Though embedded in this last aim, the model assessment and evaluation methods described here have been central to achieving all empirical objectives of this thesis, particularly in the way they relate to model specification and selection.

Model performance assessment methods are highly dependent on the modelling purpose and the properties of the model/data and include quantitative and qualitative measures. Note that in this thesis, these definitions are collectively referring to evaluation methods being used throughout the modelling process, rather than just the final model evaluation process (step 11 in the 12-step DM process of Halvorsen (2012), also in Figure 2).

In SPM models, model evaluation should, ideally, be based upon independently collected presence/absence data (Hirzel et al. 2006; Edvardsen, Bakkestuen & Halvorsen 2011b; Halvorsen 2012; Radosavljevic & Anderson 2014). Determining which model is best from the ERM perspective varies with specific ERM purpose; ERM is not only purpose but a suite of specific purposes. Common to all of these is, however, that the focus is on the ecological response curves and that these need to be realistic, calling for simple models. Accordingly, performance from an SPM perspective was evaluated by use of independent evaluation data when such data were available (PAPERS 5 and 6). When such data were not available, we cautiously used training AUC as an indicator of model performance (PAPER 3) or a randomisation test for comparison with random predictions (PAPER 4). No single best ERM model was searched for in each case; performance of candidate ERM models was instead discussed by inspection of the modelled FoP curves (see previous section), also taking model complexity into account.

AUC (the Area Under the Receiver Operating Curve) was used throughout this thesis as a relative measure of model performance. AUC has been criticised for several reasons (Lobo, Jimenez-Valverde & Real 2008; Jiménez-Valverde 2012; Li & Guo 2013), notably the lack of information on the prevalence of the modelled target in PO models. However, none of the reasons for not using AUC listed by Lobo et al. (2008) are considered relevant for the purpose of our comparisons (Pontius & Parmentier 2014) [see PAPER 5 and Halvorsen (2013) for discussion]. Both training and test AUC were calculated for all models, however, the training AUC was only reported (in conjunction with other model characteristics) in PAPERS 4-5. Whenever AUC was calculated for presence-only data, it was adjusted in accordance with Halvorsen (2013). All MaxEnt models in PAPERS 5 and 6 were evaluated by use of independently collected Presence/Absence test data.

In PAPER 3 AUC and model complexity (in terms of number of derived variables), were tested for differences among model selection methods (2), background datasets (2) and species (9), by generalized linear models (GLM, Crawley 2012).

In PAPER 4, randomisation tests were devised to test all models for non-randomness and to test for non-similarity of models for intraspecific lineages of the rare lichen *Staurolemma omphalarioides*. This allowed us to evaluate whether the final models were significantly different from random models.

The resulting 30 MaxEnt models of swamp forests (PAPER 5) were assessed by use of AUC and number of DV used in the final model. Differences between training and test AUC values were also assessed.

In PAPER 6, several types of statistical analysis are carried out to evaluate the final MaxEnt models, and determine overall optimal model complexity. As in the previous studies, the number of EVs and DVs used in the final models and the Test AUC values were calculated for each of the final 184 MaxEnt models. Test AUCs were all obtained with the use of independently collected field data. This time, these statistics were used in a secondary set of statistical analysis to determine the source of variation in overall model performance. This is accomplished in a series of gLM analysis (a subfamily of generalised linear models, GLM, McCullagh and Nelder (1989)), where the resulting test AUC values are treated as the response variables and four specific model parameters (D, S, M, and I -- **D**ata properties, **M**odel selection method, **S**trictness in model performance, and variable complexity of **I**nteraction variables) are used as the independent variables. Sequential *F*-ratio, and a significance level of $\alpha=.05$ were used to assess each variables' contribution to explaining the additional variation against the null models. The 184 observations of test AUC values made up an unreplicated but fully balanced factorial design for combinations of D, M, S and I, with the restriction that M, S and I were nested within D.

Furthermore, a separate gLM analysis was performed on the “relative performance” of 23 comparable model pairs: aMp and dMp models with a relaxed level of strictness and with no interaction variables considered (MxS1I1 models). Relative performances (ΔAUC) of each of the 23 datasets are calculated by subtracting the test-AUC of aMp from the corresponding dMp model. The statistical analysis was performed by use of linear models parameterised with ΔAUC as response variable, and the number of derived variables (*model complexity*) as the independent variables.

Finally, all final MaxEnt models raw values are multiplied by the number of background points (N) to obtain the probability-ratio output (PRO). Map representations of these were prepared for PAPERS 3 and 4. Though the range will vary for each model result, they all have a mean of approximately one, indicating “average” predicted suitability.

The last two modules of the MIAT toolbox enabled semi-automated, guided assessment of model performance for all papers.

Results

The study object and basic dataset properties (aim 4)

PAPER 6's gLM analysis of the 184 MaxEnt models' showed that the properties of the datasets accounted for 85.7% of the variation (PAPER 6: Table 3, Model 1a). This table presents the gLM models parameterised with the log of the test AUC as the modelled target (RV), and D (23 classes of Dataset properties) and the three binary M, S and I treated as the independent variables. Because of the balanced factorial design, that only left less than 1.2% of the total variation to be explained by M, S and I (Models 2a-2c). Of these remaining three models, the variation explained by the Modelling practice was, with aMp models explaining a fraction more of the variation than dMp models (average of .16 units, gLM Model 2a). The results for the variation accounted for by the M, S and I parameters are covered in later sections.

The resulting effects of the properties of the data from PAPER 3 are also presented in the next section, since they relate directly to the detection and mitigation of potential sampling bias.

In PAPER 4, the ability of MaxEnt to provide DMs for data sets that are extremely sparse was explored. The ERM models from PAPER 4 showed that the lichen *S. omphalarioides* contained genetic variation that correlates with geographical distance, although with a few shared haplotypes across disjunct ranges. Distributions of the species as well as the two genetic lineages were non-random. Detailed model specifications can be found in PAPER 4's supplementary material. See Figure 9 for map representations of the three selected models' PRO values.

Figure 9: Map representations of predictions from MaxEnt distribution models of the rare lichen species, *Staurolemma omphalarioides*, obtained for the species as such, using all known localities (a), the Northern lineage (b), and the Mediterranean lineage (c). Black squares = known localities. The scale used for model output is the probability-ratio output format (Halvorsen, 2013), which attributes a value of 1 to a grid cell with average suitability. Source PAPER 4: Figure 2.

All the results from PAPER 5 on the other hand, indicate the least amount of effect of the data model on model performance. There was very little difference between the Test AUC of any of the parameterized models, including in the choice of response variable selected. These results are presented in more details later, when addressing the potential effects of spatial autocorrelation.

Detection and mitigation of potential sampling bias (aim 5)

As the **bias detection** framework suggests, a set of presence observations contains sampling bias if the FOP curve deviates from the *true frequency of presence* (TFP) curve. The results from applying this framework on the 36 models are presented in details in PAPER 3 (see figures 17-52). The next set of figures, (figs. 53–88) present the results of employing the **bias effects** framework, showing the comparison between the *frequency of observed presence curve* (FoP) and the *predicted relative probability of presence* (PRFP) curves for the nine insect species. PRFP curves from the final MaxEnt models were, in general, similar to the corresponding observed FoP curves. This result underscores the value of being able to distinguish potential shortcomings resulting from the data or the modelling method itself. Among-curve differences were, in most cases, interpreted as indications of sampling bias. More sampling bias was detected for species that tend to be widely distributed (generalists) than those more restrictedly distributed. Indications of more sampling bias were also detected for Background Target Group (BTG) models (compare the corresponding figures in figs 17–52 and 53–88, also in PAPER 3), which also had lower AUC values compared to the Uninformed Background (UB) models.

The value of these frameworks, including the important role of the properties of the data, can be seen by focusing on the three species where the observed patterns deviated strongly from the expected. FoP curves obtained with BTG data against either environmental gradients, for two core species (*Rhagonycha limbata* and *Rhagium mordax*), and the urban species *Parnassius apollo*, deviated most strongly from the expected unimodal response. This pattern was not observed for those obtained with UB for the same species. Because of the nature of the BTG data itself, the probabilities of the model target will be different than those of the other species being sampled along those particular gradient. This leads to a more or less constant ratio between the focal and the BTG species, and a more flattened FoP curve.

Finally, most BTG curves tended to be more irregular than corresponding ones generated by UB data. Considering the much smaller sample size of the former (between 211 to 1541 observations, as compared with more than 10,000 for the latter), this is perhaps not surprising, and calls for caution in use of BTG with such low number.

Effects of spatial autocorrelation in the response variable (aim 6)

With regards to potential sampling bias due to spatial autocorrelation our results do not show a strong effect. There were no consistent overall difference in the test AUCs between the fifteen pairs of models built for RV1 and RV2 (Wilcoxon-Mann Whitney paired test: $p=0.712$, $n=30$) for either model selection method (aMp and dMp). The majority of the RV1 dMp models had higher test AUC than the equivalent built for RV2, but the opposite pattern was encountered for aMp models (Table 5), where eight of the nine RV2 models outperformed RV1 models (Wilcoxon-Mann-Whitney paired test: $p = 0.017$, $n = 18$). The highest test AUC values for RV1 and RV2 were:

0.8675 ($\lambda = 4$) and 0.8633 ($\lambda = 32$) for dMp models. Similar values were obtained for the best aMp models: 0.8627 ($\alpha = 1 \cdot 10^{-5}$) and 0.8618 (for $\alpha = 1 \cdot 10^{-11}$) respectively. The number of parameters in the final dMp models decreased consistently with increasing regularisation values, for both response variables. No similar pattern was found for aMp models.

The eleven EV, all derived from the same LiDAR source, can be divided in three groups of strongly correlated variables, with 1, 7 and 3 variables, respectively (PAPER 5: Fig. 2): (1) Slope, which had no correlation coefficient $|\tau| > 0.2$ with any other variable. (2) Topographic position, consisting of the three TPI indices, the three curvature indices and Flow accumulation. TPI3, which was strongly correlated ($|\tau| > 0.4$) with all other variables of the group, was the 'core variable' of this group. (3) Terrain ruggedness, consisting of the three VRM indices.

Model selection: options and settings (aim 7)

The results of this aims are presented together with the next one, but discussed separately.

Relating model complexity, performance and modelling purpose (aim 8)

Most of the studies reported here show that simpler models obtained by MaxEnt models with 1–5 explanatory variables, obtained by the aMp method, tend to have similar or even better predictive performance than the standard dMp MaxEnt, irrespective of model selection methods, strictness or variable complexity.

The number of derived variables in Spatial Prediction Models (parameterised by the dMp) is significantly higher than in Ecological Response Models (parameterised by the aMp), while AUC values show no significant difference (see PAPER 3: Appendix 1) and, do not necessarily perform significantly better than simpler ecological response models, even in terms of training AUC.

Results from PAPER 5 also predictably showed that all dMp models were far more complex than their comparable (that is, considering the same response variable and with similar complexity parameters) aMp models, with the former having up to 242 derived variables from the original 11 EV, compared to a maximum number of 14 DV in the latter (Table 5). No significant difference in performance between dMp and aMp models was found, and even where dMp performed better than aMp models, the differences were minimal, and overshadowed by the level of model complexity. dMp models using the centroid as response variable (RV1) performed significantly better than those with a more complex response variable (RV2), whereas the opposite trend was detected for the aMp models. Increasing the regularisation values in the dMp models led to a consistent decrease in the number of EVs and DVs in the final models for both response variables. Values for RV1 ranged from 11EVs and 101DV for $\lambda = 1$ to 7EVs and 9DV for $\lambda = 32$. Numbers for RV2 ranged from 11 EVs and 240 DVs for $\lambda = 1$ to 9 EVs and 28 DVs for $\lambda = 32$. The final aMp models contained significantly higher numbers of interaction variables than dMp models (Wilcoxon-Mann-Whitney unpaired test: $p = 0.0008$, $n = 30$). No relationship between the number of DVs in models and test AUC was found (Kendall's rank correlation coefficient: $\tau = -0.0986$, $p = 0.468$, $n = 30$).

Table 5. MaxEnt models for swamp forests in Østmarka, using two different response variables: RV1 with centroids for the 121 swamp forests recorded in the study area, and RV2 that includes all 7175 5 × 5 m grid cells in which presence of swamp forest was observed. Models were built using two model selection procedures: The default, or standard, MaxEnt procedure (dMp) refers to models obtained by using the ℓ_1 -regularisation method for model selection. The alternative MaxEnt procedure (aMp) refers to models obtained by stepwise forward selection of environmental variables (EVs), each represented by a set of transformed, or derived, variables (DVs). Strictness refers to the value used for the internal model performance assessment criterion, i.e., the values for the regularization multiplier λ in dMp and the significance level α in the F-ratio test in aMp. Training and test AUC values refer to AUC values obtained for the training and independent test data sets, respectively. EV = number of explanatory variables represented by DVs in the final model; DV = number of variables derived from EVs; P = product (interaction) variables. Rank = model rank according to test AUC. Properties of Model No. 13, which is used for studying properties of models of increasing complexity created in the forward selection process, are shown in bold face. Source PAPER 5: Table 2.

Mod. No.	Model procedure	Presence Observations	Lambda (λ) Alpha (α)	Training AUC	Test AUC	AUC difference	EV	DV	P	Rank
1	dMp	RV1	$\lambda = 1$	0.9668	0.8616	0.1050	11	101	1	8
2	dMp	RV1	$\lambda = 2$	0.9268	0.8675	0.0591	10	51	3	2
3	dMp	RV1	$\lambda = 4$	0.9572	0.8684	0.0886	9	25	4	1
4	dMp	RV1	$\lambda = 8$	0.9252	0.8666	0.0584	8	13	4	3
5	dMp	RV1	$\lambda = 16$	0.9514	0.8639	0.0873	9	13	5	4
6	dMp	RV1	$\lambda = 32$	0.9233	0.8569	0.0662	7	9	3	30
7	aMp	RV1	$\alpha = 0.25$	0.9458	0.8584	0.0872	7	14	6	22
8	aMp	RV1	$\alpha = 0.1$	0.9458	0.8584	0.0872	7	14	6	23
9	aMp	RV1	$\alpha = 0.05$	0.9458	0.8584	0.0872	7	14	6	24
10	aMp	RV1	$\alpha = 0.01$	0.9458	0.8584	0.0872	7	14	6	25
11	aMp	RV1	$\alpha = 0.001$	0.9458	0.8584	0.0872	7	14	6	26
12	aMp	RV1	$\alpha = 0.0001$	0.9458	0.8584	0.0872	7	14	6	27
13	aMp	RV1	$\alpha = 0.00001$	0.9431	0.8627	0.0802	6	9	5	6
14	aMp	RV1	$\alpha = 0.0000001$	0.9431	0.8589	0.0840	6	9	4	19
15	aMp	RV1	$\alpha = 0.0000000001$	0.9429	0.8596	0.0831	6	9	3	14
16	dMp	RV2	$\lambda = 1$	0.9606	0.8584	0.0881	11	240	2	28
17	dMp	RV2	$\lambda = 2$	0.9349	0.8587	0.0629	11	211	2	21
18	dMp	RV2	$\lambda = 4$	0.9527	0.8592	0.0796	11	122	3	18
19	dMp	RV2	$\lambda = 8$	0.9332	0.8573	0.0626	8	100	3	29
20	dMp	RV2	$\lambda = 16$	0.9475	0.8608	0.0730	7	46	4	12
21	dMp	RV2	$\lambda = 32$	0.9319	0.8633	0.0553	9	28	6	5
22	aMp	RV2	$\alpha = 0.25$	0.9299	0.8595	0.0572	7	12	5	15
23	aMp	RV2	$\alpha = 0.1$	0.9299	0.8595	0.0572	7	12	5	16
24	aMp	RV2	$\alpha = 0.05$	0.9299	0.8595	0.0572	7	12	5	17
25	aMp	RV2	$\alpha = 0.01$	0.9297	0.8608	0.0557	8	11	6	13
26	aMp	RV2	$\alpha = 0.001$	0.9298	0.8618	0.0548	7	10	6	7
27	aMp	RV2	$\alpha = 0.0001$	0.9287	0.8612	0.0543	8	11	5	10
28	aMp	RV2	$\alpha = 0.00001$	0.9287	0.8612	0.0543	8	10	5	11
29	aMp	RV2	$\alpha = 0.0000001$	0.9280	0.8589	0.0560	8	11	4	20
30	aMp	RV2	$\alpha = 0.0000000001$	0.9278	0.8615	0.0532	7	12	4	9

PAPER 6 also shows quite clearly that aMp models are invariably less complex (contained fewer parameters) than comparable dMp models. The largest difference in complexity between the two methods was found in those with the relaxed strictness criterion (S1). For comparable S1 aMp and dMp models, the average number of EVs was 6.78 and 10.02 respectively, and 71.74 and 21.20 for the DVs. Though not statistically significant (t-test: $t = -0.6769$, one-sided $P = 0.2527$, $n = 23$), the difference in the test AUC was slightly in favour of aMp models (PAPER 6: Table 4, showing the $\Delta AUC > 0$ for 12 out of 23 data sets). Furthermore, aMp models tended to perform relatively better

than dMp models when many explanatory variables were available for modelling (PAPER 6: Fig. 2).

Unsurprisingly, applying the more strict performance assessment criterion reduced the complexity of both the dMp and aMp models, most strongly for dMp models (PAPER 6: Table 2). However, models obtained by a more relaxed setting for improvement criteria performed overall slightly better than their more strict counterpart in both model selection approaches (PAPER 6: gLM Model 4a in Table 3). Increasing model complexity by adding interaction terms also did not significantly improve models. A t-test of test-AUC differences (Δ AUC) between the 23 comparable pairs of dMp and aMp models (PAPER 6: Fig. 1, Tables 4 and 5) failed to support results of gLM Model 2a in Table 3. The Δ AUC = 0.008 indicated that aMp models are overall slightly better than dMp models, but the difference was not significant (t-test: $t = -0.6769$, one-sided $P = 0.2527$, $n = 23$). The relative performance of dMp and aMp models (Δ AUC) was neither significantly related to test AUC nor to the number of presence observations for the modelled target. Test-AUC values for the 184 MaxEnt models of PAPER 6 ranged between 0.500 and 0.945 (mean=0.732).

PAPER 3's results directly relating to bias detection are presented in the earlier section. Figures 89–142, show the geographical distribution of the PRO values from the spatial prediction models (SPM) and ecological response models (ERM) respectively, for all nine species, with uninformed background and background target group. Presence observations are shown as black dots in the uninformed background (UB) maps. Similarly, Figure 2, PAPER 4 shows the PRO maps of the predicted distribution of the lichen *S. omphalarioides* at the species level, as well as the two genetic lineages.

Discussion

The added theoretical and practical control opened up by the MIAT approach has enabled a systematic empirical exploration of the importance of the different components of DM by MaxEnt. The results of the five empirical aims are discussed first in terms of the relative role each of these components play in determining model performance. This is followed by a discussion relating model complexity, performance and modelling purpose for overall model assessment, in the broader context of modelling practice. Finally, recommendations are made on how to explore these further.

The study object and basic dataset properties (aim 4)

As ecologists we intuitively know that the “nature of the object itself” must play a central role in determining its distribution, including how it will respond to its environment. As such, it is perhaps not surprising that in most of the studies presented here, the properties of the study object and basic dataset properties had the most significant effect on the predictive performance of the final models.

This is seen most strongly in the results of PAPER 6, where the D variable (23 classes of Dataset properties) accounted for 85.7% of the variation (PAPER6: Table 3, Model 1a). The fact that relative model performance of the two modelling approaches varied consistently among modelled targets and response variables, is further evidence of the data properties' strong influence. Additionally, the best performing models, those with the highest Test AUC values, were aMp models with the greatest number of EVs (and to a smaller degree, number of DVs). The number of

presence observations for the modelled target itself had no significant effect on model performance. This calls for more attention to be paid in the variable selection process, at all stages of the modelling, especially in terms of fine-tuning the model selection parameter choices to the data (Anderson & Gonzalez 2011; Aguirre-Gutiérrez *et al.* 2013; Van Looy *et al.* 2014).

PAPERS 3 and 5 also confirm the need to better understand the ecological requirements of the model target and dataset properties, and the specific issues of potential bias held the dataset properties are discussed in the subsequent two sections.

Because the original purpose of this study was to focus on model complexity and model performance related to overall model complexity and identifying an optimal model parameterisation approach, different types of Dataset properties in PAPER 6 were lumped into one categorical 'Dataset' variable. Several researchers are also urging practitioners to pay closer attention to the properties of the data being used in their studies, and call for a more detailed understanding of how these affect final model performance and assessment (Radosavljevic & Anderson 2014; García-Callejas & Araújo 2015; Moreno-Amat *et al.* 2015). Following these recommendations, a follow up to our study could be designed to better distinguish between the idiosyncrasies of the data itself from the specific dataset properties resulted from the study design itself (Barve *et al.* 2011). This would also make it possible to evaluate the appropriateness of the individual datasets and modelling method, in relation to the model complexity and modelling purpose (Stoklosa *et al.* 2015).

Finally, indirect evidence that the data model plays a far greater role in predicting model performance is evidenced by the fact that the sets of comparable models performed equally well, when independent test data were used to measure their relative performance.

How these findings relate to overall model complexity and modelling purpose shall be discussed later, when model selection methods are also considered in more details. However, it is important to note that these findings point to asking further questions regarding the extent to which the idiosyncratic properties of the data overshadow generally valid patterns and relationships. The fact that simpler (aMp) models performed similarly well (even better at times) than more complex (dMp), models means that the former are capable of capturing both the theoretical (ecological) as well as the realised (geographical) requirements of the modelled target. As ecologists we're interested not only in good model performance, but also in ensuring that the data we use adequately captures the underlying process. This in turn contributes further to our ecological understanding and guides future research. Exploring this specific aspect of data properties is particularly important with regards to issues of model transferability (Verbruggen *et al.* 2013; Keith, Elith & Simpson 2014; Schibalski, Lehtonen & Schröder 2014). Valuable insight on this could be gained by varying specific properties of some of the datasets that produced the best or worst performing models in PAPER 6, starting with the spatial grain size of the variables (Sandman *et al.* 2013; Song *et al.* 2013; Suárez-Seoane *et al.* 2014).

Finally, PAPER 4 shows the value of using DM alongside current phylogenetic methods to address ecological and historical phylogeographic questions, particularly when dealing with relatively small sample sizes and rare species. These results are quite promising, maximising the value of museum collections data for both purposes. Furthermore, additional collaborations in these fields will contribute much needed understanding for practitioners using different but complementary modelling approaches (Kalkvik *et al.* 2012; Wilson & Pitts 2012)

Detection and mitigation of potential sampling bias (aim 5)

PAPER 3 shows that addressing sampling bias in presence-only data in the context of species distribution modelling is an important issue. The results of PAPER 3 also indicate that choosing the appropriate level of model complexity is quite important.

We demonstrate how balancing theoretical and applied knowledge can be used to develop important tools to tackle issues of potential sampling bias (Stokland, Halvorsen & Støa 2011; Václavík & Meentemeyer 2012; Vale & Jenkins 2012). Drawing upon a gradient analytic perspective, we present and test comparative frameworks that use Frequency of Presence (FoP) curves to detect bias in both the data and the MaxEnt model results. By using a diverse range of presence-only data, we show that their observed FoP curves deviate quite strongly from the expected unimodal (or truncated unimodal) smooth distribution. The amount and the ways in which these deviated, varied according to the ecological properties of the species being modelled. This indicates different types of bias in the data, ranging from potential under- or over-sampling of the ecological gradient (as seen in widespread species) to random, stochastic effects (for narrowly distributed species). This further underscores the importance of taking the properties of the data into account, as noted above.

By extending this comparison of theoretical and observed curves, we can also more distinctly separate the effects that are due to the properties of the data, and those due to the modelling methods (Ottaviani, Lasinio & Boitani 2004; Gontier, Mörtberg & Balfors 2010; Newbold 2010; Gonzalez, Soto-Centeno & Reed 2011). Our results also suggest a more critical use of the BTG approach, as it does not seem to be able to differentiate between bias due to the training and the background data (Phillips et al. 2009; Bystriakova et al. 2012; Brown & Yoder 2015). This is because by definition, the background data is selected based on the “potentially biased” sampling effort of the focal group the modelled target belongs to. Furthermore, the total number of background points in BTG datasets tends to be much smaller than the uninformed background datasets (211-1541 data points for BTG compared to more than 10,000 for UB), adding another element of variability in the results that could result in decreased model performance (Phillips & Dudík 2008).

Effects of spatial autocorrelation in the response variable (aim 6)

PAPER 5 on the other hand, does not conclusively show that spatial autocorrelation in the response variable has a significant impact on the predictive performance of distribution models, something also reported by Thibaud et al. (2014). In this study, we show that representing each swamp-forest locality by numerous presence observations, i.e., by a response variable with strong spatial autocorrelation, neither reduces nor improves model performance. Most likely, this is because the additional presence observations do not bring new information about the modelled target's relationship to the environment, which is structured on much broader spatial scales than the grid-cell size used in the study. MaxEnt's strong predictive abilities, especially with high quality and high resolution data (Dormann et al. 2008) as were used in PAPER 5, may also be a contributing factor to our result.

When potential bias in a dataset is not easily detectable, especially when projecting into temporally or geographically different space, choosing models built from a forward selection

approach (aMp) is a recommended practice (Peres-Neto & Legendre 2010; Rodríguez-Rey, Jiménez-Valverde & Acevedo 2013).

Model selection: options and settings (aim 7)

In this thesis, I compare a wide range of model selection methods, including statistical and practical options and settings, and corresponding methods for variable selection, and systematically ask what effect these options and settings have on model performance.

Having addressed in the first three aims, the theoretical and practical versatility needed to “fine-tune” these options and settings (Vaughan & Ormerod 2003a; Barve *et al.* 2011; Dormann *et al.* 2012; Shcheglovitova & Anderson 2013), to the very important properties of both the model target and the dataset, the effects of each of these model selection choices are discussed first. Modelling purpose (ERM or SPM), which is fundamental in guiding these choices, will be addressed later, as part of the discussion on how to reach the overarching aim relating all these for overall model performance assessment and evaluation.

The empirical results demonstrate that, whilst still important, the choice of model selection method in itself does not play as big a role in model performance as the number of studies on this subject (including our own) would suggest. This does not mean that every settings or options will produce equally good models, but rather that more accessible and informed choices need to be taken in exploring these options. In the first section of this thesis, we have shown how opening up the MaxEnt method to a simpler and more traceable approach, made exploring the full modelling process more integrated and relatable to ecological and informatics theory.

In general, the results from PAPERS 3-6 indicate that the proposed new options resulting from a Maximum Likelihood interpretation of MaxEnt, which make up the alternative MaxEnt practice (aMp), in many cases improve MaxEnt models, even for the SPM purpose of simply providing good predictions of presence of the modelled target. Furthermore, the trail of MaxEnt models of decreasing complexity resulting from stepwise variable selection provides a good starting point for picking the appropriate model for the specific purpose of each study. This is particularly emphasised in PAPER 5.

It also seems that, irrespective of choice modelling method, the level of strictness in the model selection criteria itself has little or no significant effect on final model performance alone, confirming other findings stating that the strictness of the model performance assessment criterion has little effect on the models’ predictive performance (Merow, Smith & Silander 2013; Halvorsen *et al.* 2015). This is seen most clearly in the results of PAPERS 5 and 6, which show that when independently collected test data are used, little difference is detected in the model performance of both strict and more relaxed aMp and dMp models (PAPER 5 minimum AUC= 0.8569, maximum AUC= 0.8684). In view of the significantly larger role both the data properties and the modelling method play, this is also perhaps not surprising. However, especially in view of the importance of the data properties, further research into the effects of model strictness should be carried out by choosing only those datasets where there is evidence that the ecological gradient represented by the selected variables has been appropriately captured by the data used (Austin 2007; Zimmermann *et al.* 2007; Mateo *et al.* 2012; Millar & Blouin-Demers 2012).

Defaults options are meant to be starting points, selected for several good reasons, including making the process more accessible and easier to understand at first. These benefits may come

with the cost of reduced control and the risk of users unconsciously adopting these options, without properly exploring their appropriateness. The inflexibility of the black box approach of MaxEnt modelling by default options and settings (dMp practice), has on many occasions hindered a full understanding of the options offered by both the method and the software (Newbold 2010; Elith *et al.* 2011; Halvorsen 2013; Merow, Smith & Silander 2013; Yackulic *et al.* 2013). For many users this has meant a lack of intuitive understanding and practical control over the full model parameterisation process and interpretation of the resulting outputs. Whether produced by a series of mouse clicks, or coding of commands via the operating system shell or other tools such as R, given the complex and diverse range of outputs and the limited control over the full parameterisation of the final models (Halvorsen 2013), it is not hard to understand why maxent.jar lambda output files are rarely accessed. I maintain that more control and better access is necessary, in order to fully understand the properties of the final model, rather than the model originally specified.

The computational shortcut taken by MaxEnt 'black box' dMp comes at the expense of a loss of trail of models and full control of all options and settings. The more open and intuitive statistical and informatics approach worked through the MIAT toolbox has enabled a more informed exploration of MaxEnt models. The forward selection process suggested and the object oriented implementation are both traceable and flexible. The complexity of the results can be accessed and assessed and thus, also documented into a trail of models, in a more explicit and transparent way. The explicit use of object orientation and workflow processes, along with the more ecologically simpler, forward selection approach makes the process accessible and replicable in a computationally effective way.

Relating model complexity, performance and modelling purpose (aim 8)

The large number of both strongly and weakly performing MaxEnt models presented here, regardless of choice of model selection methods (dMp or aMp, with a broad range of model settings and options) indicates that a generally optimal strategy for MaxEnt modelling does not exist. This calls for an open approach to practical distribution modelling by which several models are compared and evaluated with respect to performance and suitability for the modelling purpose (Douma *et al.* 2012; Dray *et al.* 2012; van Son & Halvorsen 2014).

PAPERS 3 and 5 clearly show that simpler models are generally preferable to more complex models, whether or not these models' predictive performance is evaluated in geographical space (Hortal, Lobo & Jimenez-Valverde 2012; Merow *et al.* 2014b). PAPER 3 shows that bias in the data is more important for the quality of distribution models than the modelling method, supporting earlier findings by Lobo (2008), and more recent findings (Beck *et al.* 2014; Boria *et al.* 2014; Fourcade *et al.* 2014; Fithian *et al.* 2015).

PAPER 6 makes full use of the traceable automation and improved user control of the MIAT toolbox for MaxEnt modelling to tackle effectively the complexity involved in this exhaustive exploration of model selection methods. What this study shows, is that the model's predictive performance is mainly dictated by the properties of the modelled target's itself, rather than those of the modelling method alone. When the pairs of comparable MaxEnt models are evaluated on independent test data alone, aMp performed best in 12 and dMp in 11 cases, often with little difference between the two sets of AUCs. However, when we also consider model simplicity as an

important criterion for good model performance, the best aMp models were invariably less complex than the best dMp models.

Our worked examples show the importance of choosing a level of model complexity that is appropriate for the modelling purpose, be it based on modelling methods (Jimenez-Valverde 2014; Merow & Silander 2014), or variable selection and type, such as interactions (Araújo & Rozenfeld 2014; de Araújo, Marcondes-Machado & Costa 2014; Silva et al. 2014). This is important, given the strong focus on choice of method for SDM in recent literature (Elith et al. 2006; Mateo et al. 2010; Fitzpatrick, Gotelli & Ellison 2013).

Whilst the strictness criterion only had secondary effect on final model performance, the issue warrants further investigation. Although aMp and dMp models with comparable levels of strictness performed similarly, varying the regularization multiplier λ had a stronger effect (dMp models), than changing the significance level α in the F-ratio test (aMp models). We argue that although complexity is reduced by choosing stricter selection criteria for both methods, the fact that this difference is stronger for dMp models, makes this model selection method more susceptible to variable setting choices (Warren & Seifert 2011; Radosavljevic & Anderson 2014).

Indeed, all of the work presented in this thesis consistently shows that all aMp models, tended to have similar or even better predictive performance than the comparable (with regards to options and settings) dMp models. Furthermore, in terms of number of parameters of the final models, aMp models were significantly simpler than the corresponding dMp models. Being able to balance theoretical and practical consideration when distinguishing between simple and more complex models, i.e. between models that are too simplistic and models that over fit the data to an extent that its idiosyncratic properties overshadow generally valid patterns and relationships (Jiménez-Valverde, Lobo & Hortal 2008; Halvorsen 2012; Merow et al. 2014b) is critical and better guidelines and control is key. In line with ecological theory, I propose that this also indicates that the balance point often may lie closer to the simple than the complex end of this complexity gradient. Furthermore, I argue that the flexible modelling practice proposed here has several favourable properties, such as full user control over model complexity and ability to produce simpler, easily interpretable models that facilitate ecological understanding.

Rather than considering modelling purpose only at the project proposal stage, and choosing a fixed methodology based on that, it is important to review both when assessing final model complexity and overall model performance. Recent papers (Aguirre-Gutiérrez et al. 2013; Radosavljevic & Anderson 2014; García-Callejas & Araújo 2015; Guillera-Aroita et al. 2015) support our findings that no single optimal strategy for model selection and parameterisation exists, and that closer attention should be paid to matching the properties of the data, the modelling methods, and modelling purpose.

Concluding remarks and recommendations

Having demonstrated the key role of the properties of the data model, gives further support to claims that final model performance should be evaluated wherever possible with independently collected datasets (Veloz 2009; Edvardsen, Bakkestuen & Halvorsen 2011b; Halvorsen 2012) which, amongst other things, enables additional decoupling of potential effects resulting from the properties of the data itself, as evidenced in PAPERS 5 and 6. More explicit awareness of how the properties of the data model, the modelling method and the tools, relate to one another and to modelling purpose (Dormann et al. 2012), should also be given a higher priority.

Whilst addressing quality and uncertainties of the properties of the data itself (Maxwell *et al.* 2015; Naimi 2015) was beyond the remit of this thesis, future studies should look into this aspect more closely, including the knowledge that practitioners have in addressing these potential shortcomings of their data and tools used in processing them. The results of PAPER 3 and 4, where the test AUC is considerably lower than the training AUC, are important reminders of the critical role data properties have in model performance and assessment, and of the value in using independently collected test data in determining model performance. Additional investigations on the identity and quality of the data used by those models are also recommended. PAPER 5 shows that training and test AUCs do not rank corresponding models similarly. A detailed investigation as to why the evaluated distribution models had different success in predicting the modelled target is also warranted.

Once the properties of the data are adequately understood and accounted for, a full exploration of the range of options and settings is needed. The degree of complexity of these parameters can be varied accordingly, be it guided by the consideration of modelling purpose, the properties of the data, the modelling tools, or the research question to be explored. By employing the flexible and traceable MIAT approach we have put forward, these elements can all be explored systematically. Improving our ability to trace and interpret our modelling results both in theoretical and practical terms, benefits the evaluation of the model itself (including facilitating reproducibility), and contributes to maintaining a dynamic link between theory and practice. Computational methods will continue to improve, which in turn will improve our ability to produce and process more data and/or models (Rahn et al. 2013; Visser et al. 2015). This may in turn require even more statistically sophisticated approaches and computational tools to handle them. Thus, a constant monitor on how practitioners understand these tools and how to implement them is necessary (Mackenzie 2013; Merow et al. 2014a; Stasch et al. 2014). Therefore, extending this flexibility and traceability to the discipline as a whole, would further ensure more responsive and implementable solutions, better guiding of choices between simplicity and complexity of the process (Kissling et al. 2012; Merow et al. 2014b; Visser et al. 2015).

Finally, we need to keep building expert knowledge (Niamir 2014; van Zonneveld et al. 2014) from an ecological (Stewart-Koster et al. 2013; Warton 2015), a geo-statistical (Parviainen et al. 2013; Rocchini 2013; Nazeri et al. 2015) and an informatics (Walls et al. 2014; Holst & Belete 2015) perspective. This is essential in maintaining a dynamic link in how new findings are implemented and understood in practice, so as to more responsibly hide some of the (theoretical or computational) complexity but improve accessibility and final model performance (Engler, Hordijk & Guisan 2012). The flexible MIAT toolbox presented here may not suit all practitioners,

and the simplicity and openness of the code has come at the cost of a rather complex and rudimentary implementation.

Now that the flexible implementation of the MIA framework and toolbox has been worked out, the code of the MIAT toolbox should be substantially improved, including making better use of higher level functions and the flexible structure of R's list objects. Key elements of these novel options, including more accessible guided user control via modular toolboxes, could also be embedded into existing modelling parameterisation and analysis packages such as DISMO, BIOMOD, ENMTools (Thuiller et al. 2009; Warren, Glor & Turelli 2010; Hijmans & Elith 2013). These have been very effective in pushing existing theoretical understandings of how different elements in DM come together, but have often required a more sophisticated level of coding expertise, and have thus far only relied on the existing fixed options offered by `maxent.jar`.

The MIAT toolbox has been developed in great detail to explore the more exhaustive and thorough comparison of alternative options and variants of the MaxEnt method, the properties of the data and the modelling purpose, and coded in a way that would not require access to more complex tools accessible to most current users of the *default* approach.

Fully incorporating additional informatics development and authoring tools, such as Knitr with R Markdown, Sweave (Leisch 2002), LaTeX (Lamport 1986) and collaborating platforms such as GitHub, `sdm` (Naimi & Araújo 2016), *zoonproject*, *Kepler* (Barseghian et al. 2010), will greatly improve MIAT's flexibility and reproducibility. It will also make extending this Modular Integrated Approach to other methods and tools easier, thus contributing to making the distribution modelling practice be more flexibly adaptable to the complexity of the research and/or policy question they are being applied to. To this end, efforts into translating the core functionalities of the toolbox into a full-fledge novel R-package have also been underway, with close attention being paid to the level of complexity they entail.

Let us not forget that modelling itself is just a tool, a tool that approximates reality, and it is up to all modellers to understand how to balance the simplicity and complexity of these tools.

Acknowledgements

In theory, the scope and contents of a doctoral thesis should be kept tightly bound within a very fixed temporal and topical box. In practice, when the original text advertising the post is titled: *“Prediction modelling as a tool for mapmaking, monitoring and management of biodiversity”*, chances are, that box is wide open.

Paradoxically, this thesis ended up having a very tight scope by metaphorically opening up the *“black box approach”* of one of the most popular distribution modelling tools for ecologists, Maxent.jar. The emphasis has been on exposing the core theoretical and practical elements contained herein, and in turn, through various iterative rounds, reintegrate it and rewrap the process into a more flexible and practical approach and toolbox. This flexibility and integration hopefully contributes added ecological interpretability and traceability of both the statistical method and the tools.

The nature of the dynamic link between the theory and practice is also reflected in the work carried out throughout this thesis, which has evolved from the iterative exploration of this subject not simply through the literature, but also, via active discussions and working collaborations with fellow PhD students, NHM-UiO staff and IMN/GEco members, family, friends and researchers from diverse range of environments and literature. The process itself has been a metaphorical decoupling of several multidisciplinary components, spanning both temporal and geographical spaces, finding novel ways of communicating, trade-offs between what is theoretically possible and what is applicable in practice, simplicity and complexity. I wish to extend a broad scale thank you to everyone who has contributed specifically to this work, both personally and professionally.

A deep rooted, wide reaching personal appreciation for the enriching, diverse, loving (albeit challenging at times) but always supportive environment that every single one of my family, friends, and colleagues/mentors have provided me across the three continents and five decades of my journey: without all of this, and the inextinguishable energy and hope I somehow possess, I would not have managed to get this far. Despite the much longer gestation period, higher number of delivered and aborted projects, and after having lost more I could have ever imagined, including a *beautiful soul*, I’m still grateful to have been given the opportunity and support to carry this PhD journey to full term.

Please see next page for more detailed ringraziamenti.

A sincere appreciation to the institutional gatekeepers, for being willing to look past their own pre conceptions, and let this age and gender non-conforming outsider embark on this privileged, and rewarding (albeit perilous) journey.

Extended **thank you** to my 3 very different supervisors. Jogeir Stokland: for playing a crucial role in getting both our nascent research group off the ground, and being there for me, its first PhD student, at the start. Vegar Bakkestuen, for being so predictably yourself: reliably late yet dependably impeccable in your contribution, be that a “*beer-full*” of local knowledge and key thesis insight, or more importantly, teaching me how to extract a simpler and smoother story out of the entangled layers of complexity of my work.

Rune Halvorsen: for being in many ways, the exact opposite, including allowing me to extract simpler and smoother messages out of your less tangled but equally complex and lengthy sentences. Thank you, also for teaching me the patience to learn from things I didn’t think were worth learning from, and showing incredible patience in teaching me things I was too impatient to recall I had previously learned. Finally thank you for seeing past the abstract complexity of my language, and trust we’d eventually understand each other (time and again). I think we both did. And despite the challenging delays, whether due to personal or workplace related overload of responsibilities and commitments, you always made time to follow my work, including when you didn’t have time.

Thank you to everyone who has collaborated with me, for being open (and patient) to the alternative but often complementary approaches I bring to the table, including the lunch and language table. A special **takk** to all IMN/GEco members, for all the inspiring seminars, enriching personal and professional conversations; and for being generally curious, especially by the diversity of my matpakke. **Hjertelig takk** Trine Bekkby, for the professional and personal ways in which you’ve been keeping the group going, especially lifting my spirit and always being willing to actually taste the contents of my curious lunches! Thijs Christiaan van Son and Knut Anders Hovstad *deseRve* an individual mention for *shaRing theiR* knowledge, and Bente Støa and John Wirkola Dirksen for being willing to test mine.

Takk til alle mine kjære kolleger på NHM, fra alle avdelinger: takk for å ha skapt, gjennom og til tross forskjellig forhold, et vakkert sted å være, inne og ute, og alltid hilste meg med en smil. Takk til alle på Dagny Tande Lids Hus, som har delt masse historier og lært meg mye om norske vaner, språk, mat, sopp, og god humor! I særlig grad, til Siri Rui og Anders Wollan, min egen ‘**guardian angel**’! (Anders, du kunne ha vært litt misunnelig over at jeg ble valgt over deg til doktorgrad-stillingen da den først ble ulyst, men din sjenerøsitet er alt du har vist meg. Det var kanskje bedre for deg å ikke være den først stipendiaten i gruppa, og fikk du en GIS skytsengel etterhvert!)

Thanks to my mentor Wenche Dramstad: for being there helping me “*plan the changes*” but also being ok with me constantly “*changing the plans*”.

Thanks to all the children in my life (and to their family for sharing them with me): thank you for your never ending colourful laughter and tears!

Pakka þér fyrir to all my dear neighbours and friends for all the shared picnics, barbeques, or impromptu dinner parties, and always checking in on me.

A special **bedankt** to my fellow toast master of the queer version of the traditional *Herrenes og Damen Juletale* and Skeiv Verden's board member: for being supportive in so many ways, including making sure I would not be the only 40+ year old student at NHM.

Gracias a todos mis colegas y amigos con quien he hablado español, de la Coruña a la Catalonia, del Argentina/Uruguay al Mexico: emos compartido mucho, comida, casa y mucho más. **Asante** to all members of Afro-Alp, big and small, old and new, original and extended ones.

Merci aussi à les nouveaux arrivées, ils m'ont aidé a pas perdre l'espoir, même s'ils ont risqué' la ouvrais perdu sois meme. Tack alla, gamla och nya, for sharing pasiunea mea pentru pranzuri internaționale, including crossing icy rivers and reaching high peaks, a aby se moje sdílení mezinárodní oběd vašeň do nových výšin.

Excharisto mikrós pringipos, takk for reminding me not to fear the pink error code. Πράσινο: για τις άνοιξη που θα είναι.

Finally, despite the difficulties and the tough choices that at times felt beyond my control, I am grateful for having finally achieved my "modelling purpose": delivering this PhD. My deepest and most humble gratitude for this goes to my *Italian* and *Kiwi* families.

Grazie a tutti and Kia Ora.

She is tossed by the waves,

but does not sink.

<https://www.facebook.com/artsbymcm/>

*To laugh often and much;
To win the respect of intelligent people and the affection of children;
To earn the appreciation of honest critics and endure the betrayal of false friends;
To appreciate beauty, to find the best in others;
To leave the world a bit better,
whether by a healthy child, a garden patch or a redeemed social condition;
To know even one life has breathed easier because you have lived.
This is to have succeeded.*

Ralph Waldo Emerson (1803 - 1882)

Note: although this quote has always been attributed to Emerson, it has never been located within any of his work.

References

- Aarvik, L., Hansen, L.O. & Kononenko, V. (2009) *Norges sommerfugler. Håndbok over Norges dagsommerfugler og nattsvermere*. Norsk entomologisk forening og NHM, UiO, Oslo.
- Acevedo, P., Jiménez-Valverde, A., Lobo, J.M. & Real, R. (2012) Delimiting the geographical background in species distribution modelling. *Journal of Biogeography*, **39**, 1383-1390.
- Adler, J. (2010) *R in a nutshell: A desktop quick reference*. O'Reilly Media, Sebastopol, California. USA.
- 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.
- Ahti, T., Hämet-Ahti, L. & Jalas, J. (1968) Vegetation zones and their sections in northwestern Europe. *Annales Botanici Fennici*, pp. 169-211.
- Anderson, R.P. & Gonzalez, I. (2011) Species-specific tuning increases robustness to sampling bias in models of species distributions: An implementation with Maxent. *Ecological Modelling*, **222**, 2796-2811.
- Anonymous (2008) ArcGIS Desktop and Spatial Analyst Extension, v. 9.3. *Environmental Systems Research Institute, Redlands, California*.
- Anonymous (2012) ArcGIS Desktop and Spatial Analyst Extension. v. 10.1. *Environmental Systems Research Institute, Redlands, California*.
- Araújo, M.B. & Rozenfeld, A. (2014) The geographic scaling of biotic interactions. *Ecography* **37**, 406-415.
- Auestad, I., Halvorsen, R., Bakkestuen, V. & Erikstad, L. (2012) Videreføring av prediksjonsmodellering av invaderende fremmede arter. *DN Rapport*, **6**, 1-38.
- Austin, M. (2007) Species distribution models and ecological theory: a critical assessment and some possible new approaches. *Ecological Modelling*, **200**, 1-19.
- Austin, M.P. (2002) Spatial prediction of species distribution: an interface between ecological theory and statistical modelling. *Ecological Modelling*, **157**, 101-118.
- Bakkestuen, V., Erikstad, L. & Halvorsen, R. (2008) Step-less models for regional environmental variation in Norway. *Journal of Biogeography*, **35**, 1906-1922.
- Bakkestuen, V., Stabbetorp, O.E., Molia, A., Evju, M., Erikstad, L., Bekkby, T. & Halvorsen, R. (2014) Hotspot åpen grunnlendt kalkmark i Oslofjordområdet. Beskrivelse av habitatet og forslag til overvåkingsopplegg fra ARKO-prosjektet. Effekten av skala og kvalitet i prediksjon av bunntyper. *Vann*, pp. 46. Norsk institutt for naturforskning, Norsk Vannforening, Oslo, Norway.
- 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.
- Bartsch, H., Binkiewicz, E., Rådén, A. & Nasibov, E. (2009) *Nationalnyckeln till Sveriges flora och fauna. Tvåvingar: Blomflugor: Syrphinae. Diptera: Syrphidae: Syrphinae*. ArtDatabanken, Uppsala.
- Barve, N., Barve, V., Jiménez-Valverde, A., Lira-Noriega, A., Maher, S.P., Peterson, A.T., Soberón, J. & Villalobos, F. (2011) The crucial role of the accessible area in ecological niche modeling and species distribution modeling. *Ecological Modelling*, **222**, 1810-1819.
- Beck, J., Böller, M., Erhardt, A. & Schwanghart, W. (2014) Spatial bias in the GBIF database and its effect on modeling species' geographic distributions. *Ecological Informatics*, **19**, 10-15.
- Beck, K. (2000) *Extreme programming explained: embrace change*. Addison-Wesley Professional, University of Michigan, USA.

- Bekkby, T. & Moy, F. (2011) Developing spatial models of sugar kelp (*Saccharina latissima*) potential distribution under natural conditions and areas of its disappearance in Skagerrak. *Estuarine, Coastal and Shelf Science*, **95**, 477-483.
- 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.
- Boria, R.A., Olson, L.E., Goodman, S.M. & Anderson, R.P. (2014) Spatial filtering to reduce sampling bias can improve the performance of ecological niche models. *Ecological Modelling*, **275**, 73-77.
- Breidenbach, J. & Astrup, R. (2012) Small area estimation of forest attributes in the Norwegian National Forest Inventory. *European Journal of Forest Research*, **131**, 1255-1267.
- Brown, J.L. & Yoder, A.D. (2015) Shifting ranges and conservation challenges for lemurs in the face of climate change. *Ecology and Evolution*, **5**, 1131-1142.
- Buckland, S., Cole, N.C., Aguirre-Gutiérrez, J., Gallagher, L.E., Henshaw, S.M., Besnard, A., Tucker, R.M., Bachraz, V., Ruhomaun, K. & Harris, S. (2014) Ecological Effects of the Invasive Giant Madagascar Day Gecko on Endemic Mauritian Geckos: Applications of Binomial-Mixture and Species Distribution Models. *PLoS ONE*, **9**, e88798.
- Burger, A. & Page, R. (2007) The need for biological realism in habitat modeling: a reinterpretation of Zharikov et al. (2006). *Landscape Ecology*, **22**, 1273-1281.
- Bystriakova, N., Peregrym, M., Erkens, R.H., Bezsmertna, O. & Schneider, H. (2012) Sampling bias in geographic and environmental space and its effect on the predictive power of species distribution models. *Systematics and Biodiversity*, **10**, 305-315.
- Chala, D., Brochmann, C., Psomas, A., Dorothee, E., Gizaw, A., Masao, C., Bakkestuen, V. & Zimmermann, N. (Submitted) Goodbye to tropical alpine plant giants under warmer climates? Loss of range and genetic diversity in *Lobelia rhynchopetalum*. *Global Ecology and Biogeography*.
- Collins, S.L., Glenn, S.M. & Roberts, D.W. (1993) The hierarchical continuum concept. *Journal of Vegetation Science*, **4**, 149-156.
- Crall, A.W., Jarnevich, C.S., Panke, B., Young, N., Renz, M. & Morisette, J. (2013) Using habitat suitability models to target invasive plant species surveys. *Ecological Applications*, **23**, 60-72.
- Crawley, M.J. (2012) Generalized Linear Models. *The R Book*, pp. 557-578. John Wiley & Sons, Ltd, Chichester, England.
- 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.
- de Araújo, C.B., Marcondes-Machado, L.O. & Costa, G.C. (2014) The importance of biotic interactions in species distribution models: a test of the Eltonian noise hypothesis using parrots. *Journal of Biogeography*, **41**, 513-523.
- del Barrio, G., Harrison, P.A., Berry, P.M., Butt, N., Sanjuan, M.E., Pearson, R.G. & Dawson, T. (2006) Integrating multiple modelling approaches to predict the potential impacts of climate change on species' distributions in contrasting regions: comparison and implications for policy. *Environmental Science & Policy*, **9**, 129-147.
- Della Pietra, S., Della Pietra, V. & Lafferty, J. (1997) Inducing features of random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **19**, 380-393.
- Dirksen, J.W. (2013) Modelling presence of swamp forest and forest dwelling birds in a boreal forest reserve using airborne laser scanning. Philosophiae doctor thesis, Norwegian University of Life Sciences.
- 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.

- Dormann, C.F., Purschke, O., Márquez, J.R.G., Lautenbach, S. & Schröder, B. (2008) Components of uncertainty in species distribution analysis: a case study of the great grey shrike. *Ecology*, **89**, 3371-3386.
- Dormann, C.F., Schymanski, S.J., Cabral, J., Chuine, I., Graham, C., Hartig, F., Kearney, M., Morin, X., Römermann, C., Schröder, B. & Singer, A. (2012) Correlation and process in species distribution models: bridging a dichotomy. *Journal of Biogeography*, **39**, 2119-2131.
- Douma, J.C., Witte, J.-P.M., Aerts, R., Bartholomeus, R.P., Ordoñez, J.C., Olde Venterink, H., Wassen, M.J. & van Bodegom, P.M. (2012) Towards a functional basis for predicting vegetation patterns; incorporating plant traits in habitat distribution models. *Ecography*, **35**, 294-305.
- Dray, S., Péliissier, R., Couteron, P., Fortin, M.J., Legendre, P., Peres-Neto, P.R., Bellier, E., Bivand, R., Blanchet, F.G., De Cáceres, M., Dufour, A.B., Heegaard, E., Jombart, T., Munoz, F., Oksanen, J., Thioulouse, J. & Wagner, H.H. (2012) Community ecology in the age of multivariate multiscale spatial analysis. *Ecological Monographs*, **82**, 257-275.
- Dudík, M., Phillips, S.J. & Schapire, R.E. (2007) Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. *Journal of Machine Learning Research*, **8**, 1217-1260.
- Edvardsen, A., Bakkestuen, V. & Halvorsen, R. (2011a) A fine-grained spatial prediction model for the red-listed vascular plant *Scorzonera humilis*. *Nordic Journal of Botany*, **29**, 495-504.
- Edvardsen, A., Bakkestuen, V. & Halvorsen, R. (2011b) A fine-grained spatial prediction model for the red-listed vascular plant *Scorzonera humilis*. *Nordic Journal of Botany*, **29**, 495-504.
- Edwards, T.C.J., Cutler, D.R., Zimmermann, N.E., Geiser, L. & Alegria, J. (2005) Model-based stratifications for enhancing the detection of rare ecological events. *Ecology*, **86**, 1081-1090.
- Ehnström, B. & Holmer, M. (2007) *Nationalnyckeln till Sveriges flora och fauna. Skalbaggar: Långhorningar. Coleoptera: Cerambycidae*. ArtDatabanken, Uppsala.
- Elith, J. & Graham, C.H. (2009) Do they? How do they? WHY do they differ? On finding reasons for differing performances of species distribution models. *Ecography*, **32**, 66-77.
- Elith, J., Graham, C.H., Anderson, R.P., Dudík, M., Ferrier, S., Guisan, A., Hijmans, R.J., Huettmann, F., Leathwick, J.R., Lehmann, A., Li, J., Lohmann, L.G., Loiselle, B.A., Manion, G., Moritz, C., Nakamura, M., Nakazawa, Y., Overton, J.M.M., Peterson, A.T., Phillips, S.J., Richardson, K., Scachetti-Pereira, R., Schapire, R.E., Soberón, J., Williams, S., Wisz, M.S. & Zimmermann, N.E. (2006) Novel methods improve prediction of species' distributions from occurrence data. *Ecography*, **29**, 129-151.
- Elith, J. & Leathwick, J.R. (2009) Species Distribution Models: Ecological Explanation and Prediction Across Space and Time. *Annual Review of Ecology, Evolution, and Systematics*, **40**, 677-697.
- Elith, J., Phillips, S.J., Hastie, T., Dudík, M., Chee, Y.E. & Yates, C.J. (2011) A statistical explanation of MaxEnt for ecologists. *Diversity and Distributions*, **17**, 43-57.
- Endresen, D.T.F., Street, K., Mackay, M., Bari, A. & De Pauw, E. (2011) Predictive association between biotic stress traits and ecogeographic data for wheat and barley landraces. *Crop science*, **51**, 2036-2055.
- 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.
- Fegraus, E.H., Andelman, S., Jones, M.B. & Schildhauer, M. (2005) Maximizing the value of ecological data with structured metadata: an introduction to Ecological Metadata Language (EML) and principles for metadata creation. *Bulletin of the Ecological Society of America*, **86**, 158-168.
- Fithian, W., Elith, J., Hastie, T. & Keith, D.A. (2015) Bias correction in species distribution models: pooling survey and collection data for multiple species. *Methods in Ecology and Evolution*, **6**, 424-438.
- Fitzpatrick, M.C., Gotelli, N.J. & Ellison, A.M. (2013) MaxEnt versus MaxLike: empirical comparisons with ant species distributions. *Ecosphere*, **4**, art55.

- Fourcade, Y., Engler, J.O., Rödder, D. & Secondi, J. (2014) Mapping Species Distributions with MAXENT Using a Geographically Biased Sample of Presence Data: A Performance Assessment of Methods for Correcting Sampling Bias. *PLoS ONE*, **9**, e97122.
- Franklin, J. (2009) *Mapping species distributions: spatial inference and prediction*. Cambridge University Press, Cambridge.
- García-Callejas, D. & Araújo, M.B. (2015) The effects of model and data complexity on predictions from species distributions models. *Ecological Modelling*.
- Gibson, L., Barrett, B. & Burbridge, A. (2007) Dealing with uncertain absences in habitat modelling: a case study of a rare ground-dwelling parrot. *Diversity and Distributions*, **13**, 704-713.
- Gontier, M., Mörtberg, U. & Balfors, B. (2010) Comparing GIS-based habitat models for applications in EIA and SEA. *Environmental Impact Assessment Review*, **30**, 8-18.
- Gonzalez, S., Soto-Centeno, J.A. & Reed, D. (2011) Population distribution models: species distributions are better modeled using biologically relevant data partitions. *BMC Ecology*, **11**, 20.
- Gould, S.F., Beeton, N.J., Harris, R.M.B., Hutchinson, M.F., Lechner, A.M., Porfirio, L.L. & Mackey, B.G. (2014) A tool for simulating and communicating uncertainty when modelling species distributions under future climates. *Ecology and Evolution*, **4**, 4798-4811.
- Graham, C.H., Ferrier, S., Huettman, F., Moritz, C. & Peterson, A.T. (2004) New developments in museum-based informatics and applications in biodiversity analysis. *Trends in Ecology & Evolution*, **19**, 497-503.
- Guisan, A., Lehmann, A., Ferrier, S., Austin, M., Overton, J.M.C., Aspinall, R. & Hastie, T. (2006) Making better biogeographical predictions of species' distributions. *Journal of Applied Ecology*, **43**, 386-392.
- Guisan, A., Tingley, R., Baumgartner, J.B., Naujokaitis-Lewis, I., Sutcliffe, P.R., Tulloch, A.I.T., Regan, T.J., Brotons, L., McDonald-Madden, E., Mantyka-Pringle, C., Martin, T.G., Rhodes, J.R., Maggini, R., Setterfield, S.A., Elith, J., Schwartz, M.W., Wintle, B.A., Broennimann, O., Austin, M., Ferrier, S., Kearney, M.R., Possingham, H.P. & Buckley, Y.M. (2013) Predicting species distributions for conservation decisions. *Ecology Letters*, **16**, 1424-1435.
- Guisan, A. & Zimmermann, N.E. (2000) Predictive habitat distribution models in ecology. *Ecological Modelling*, **135**, 147-186.
- Gundersen, H., Bekkby, T., Norling, K., Rygg, B., Norderhaug, K.M., Christie, H.C., Andersen, G.S. & Bekkby, T. (2012) Using GAM models to predict soft sediment diversity - modelling reference condition along the Norwegian coast. Does the diversity of kelp forest macrofauna increase with wave exposure? *MAREANO conference*, pp. 36-42. Nederlands Instituut voor Onderzoek der Zee, Trondheim.
- 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*, **36**, 1-132.
- 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.
- Hansen, V. & Larsson, S. (1973) Biller X. Blødvinger, klannere m. m.: (Malacodermata, Fossipedes, Marcrodactylia og Brachymera). *Danmarks Fauna*, **44**, 41-42.
- Hastie, T. & Fithian, W. (2013) Inference from presence-only data; the ongoing controversy. *Ecography*, **36**, 864-867.
- Hastie, T., Tibshirani, R. & Friedman, J. (2009) *The elements of statistical learning*, ed. 2. Springer, New York.
- Heibl, C. & Renner, S.S. (2012) Distribution models and a dated phylogeny for Chilean *Oxalis* species reveal occupation of new habitats by different lineages, not rapid adaptive radiation. *Systematic Biology*, **61**, 823-834.

- Heller, N.E. & Zavaleta, E.S. (2009) Biodiversity management in the face of climate change: A review of 22 years of recommendations. *Biological Conservation*, **142**, 14-32.
- Hernandez, P.A., Graham, C.H., Master, L.L. & Albert, D.L. (2006) The effect of sample size and species characteristics on performance of different species distribution modeling methods. *Ecography*, **29**, 773-785.
- Hijmans, R., Cameron, S., Parra, J., Jones, P. & Jarvis, A. (2005) Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology*, **25**, 1965-1978.
- 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., Helfer, V., Randin, C. & Guisan, A. (2006) Evaluating the ability of habitat suitability models to predict species presences. *Ecological Modelling*, **199**, 142-152.
- Holst, N. & Belete, G.F. (2015) Domain-specific languages for ecological modelling. *Ecological Informatics*, **27**, 26-38.
- Hortal, J., Lobo, J.M. & Jimenez-Valverde, A. (2012) Basic Questions in Biogeography and the (Lack of) Simplicity of Species Distributions: Putting Species Distribution Models in the Right Place. *Natureza & Conservacao*, **10**, 108-118.
- Hovstad, K., Grenne, S. & Warda, M. (2012) Habitat modelling with high resolution in an agricultural landscape. *Grassland-a European resource? Proceedings of the 24th General Meeting of the European Grassland Federation*, pp. 682-684. Polskie Towarzystwo Łakarskie (Polish Grassland Society), Lublin, Poland.
- Hörnberg, G., Zackrisson, O., Segerström, U., Svensson, B.W., Ohlson, M. & Bradshaw, R.H. (1998) Boreal swamp forests. *BioScience*, 795-802.
- Jarvis, A., Reuter, H., Nelson, A. & Guevara, E. (2008) Hole-filled seamless SRTM data V4, International Centre for tropical Agriculture (CIAT).
- Jaynes, E.T. (1957) Information theory and statistical mechanics I-II. *Physical Review*, **106**, **108**, 620-630, 171-190.
- Jimenez-Valverde, A. (2014) Threshold-dependence as a desirable attribute for discrimination assessment: implications for the evaluation of species distribution models. *Biodiversity and Conservation*, **23**, 369-385.
- Jiménez-Valverde, A. (2012) Insights into the area under the receiver operating characteristic curve (AUC) as a discrimination measure in species distribution modelling. *Global Ecology and Biogeography*, **21**, 498-507.
- Jiménez-Valverde, A., Lobo, J. & Hortal, J. (2008) Not as good as they seem: the importance of concepts in species distribution modelling. *Diversity and Distributions*, **14**, 885-890.
- Jørgensen, S.E. (1993) Model-Theoretical aspects of ecological modelling. *Ecological Modelling*, **68**, viii.
- Kadmon, R. & Allouche, O. (2007) Integrating the effects of area, isolation, and habitat heterogeneity on species diversity: a unification of island biogeography and niche theory. *American Naturalist*, **170**, 443-454.
- Kalkvik, H.M., Stout, I.J., Doonan, T.J. & Parkinson, C.L. (2012) Investigating niche and lineage diversification in widely distributed taxa: phylogeography and ecological niche modeling of the *Peromyscus maniculatus* species group. *Ecography*, **35**, 54-64.
- Keith, D.A., Elith, J. & Simpson, C.C. (2014) Predicting distribution changes of a mire ecosystem under future climates. *Diversity and Distributions*, **20**, 440-454.
- Kendall, M.G. (1938) A new measure of rank correlation. *Biometrika*, 81-93.
- Kissling, W.D., Dormann, C.F., Groeneveld, J., Hickler, T., Kühn, I., McInerny, G.J., Montoya, J.M., Römermann, C., Schiffrers, K., Schurr, F.M., Singer, A., Svenning, J.-C., Zimmermann, N.E. & O'Hara, R.B. (2012)

Towards novel approaches to modelling biotic interactions in multispecies assemblages at large spatial extents. *Journal of Biogeography*, **39**, 2163-2178.

Lamport, L. (1986) *LaTeX: User's Guide & Reference Manual*.

Legendre, P. & Legendre, L. (2012) *Numerical ecology*. Elsevier, Amsterdam.

Leinfelder, B., Tao, J., Costa, D., Jones, M.B., Servilla, M., O'Brien, M. & Burt, C. (2010) A metadata-driven approach to loading and querying heterogeneous scientific data. *Ecological Informatics*, **5**, 3-8.

Leisch, F. (2002) Sweave: Dynamic generation of statistical reports using literate data analysis. *Compstat*, pp. 575-580. Springer.

Li, W. & Guo, Q. (2013) How to assess the prediction accuracy of species presence-absence models without absence data? *Ecography*, **36**, 788-799.

Lobo, J.M., Jimenez-Valverde, A. & Real, R. (2008) AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, **17**, 145-151.

Mackenzie, A. (2013) Programming subjects in the regime of anticipation: Software studies and subjectivity. *Subjectivity*, **6**, 391-405.

Mateo, R.G., Croat, T.B., Felicísimo, Á.M. & Muñoz, J. (2010) Profile or group discriminative techniques? Generating reliable species distribution models using pseudo-absences and target-group absences from natural history collections. *Diversity and Distributions*, **16**, 84-94.

Mateo, R.G., Felicísimo, A.M. & Muñoz, J. (2010) Effects of the number of presences on reliability and stability of MARS species distribution models: the importance of regional niche variation and ecological heterogeneity. *Journal of Vegetation Science*, **21**, 908-922.

Mateo, R.G., Felicísimo, Á.M., Pottier, J., Guisan, A. & Muñoz, J. (2012) Do Stacked Species Distribution Models Reflect Altitudinal Diversity Patterns? *PLoS ONE*, **7**, e32586.

Matloff, N. (2011) *The art of R programming: A tour of statistical software design*. No Starch Press, San Francisco, CA.

Maxwell, S.L., Rhodes, J.R., Runge, M.C., Possingham, H.P., Ng, C.F. & McDonald-Madden, E. (2015) How much is new information worth? Evaluating the financial benefit of resolving management uncertainty. *Journal of Applied Ecology*, **52**, 12-20.

Mazzoni, S., Halvorsen, R., Bakkestuen, V. & Bratli, H. (2011) Romlig prediksjonsmodellering av kulturmarkseng i Oppdal kommune (Sør-Trøndelag). *Faglig grunnlag for naturtypeovervåking i Norge – grunnlagsundersøkelser*, pp. 221. Universitetet i Oslo, Naturhistorisk museum, Oslo, Norway.

McCullagh, P. & Nelder, J.A. (1989) *Generalized linear models*. Chapman and Hall/CRC, Boca Raton, Florida.

Merckx, B., Steyaert, M., Vanreusel, A., Vincx, M. & Vanaverbeke, J. (2011) Null models reveal preferential sampling, spatial autocorrelation and overfitting in habitat suitability modelling. *Ecological Modelling*, **222**, 588-597.

Merow, C., Dahlgren, J.P., Metcalf, C.J.E., Childs, D.Z., Evans, M.E.K., Jongejans, E., Record, S., Rees, M., Salguero-Gómez, R. & McMahon, S.M. (2014a) Advancing population ecology with integral projection models: a practical guide. *Methods in Ecology and Evolution*, **5**, 99-110.

Merow, C. & Silander, J.A. (2014) A comparison of Maxlike and Maxent for modelling species distributions. *Methods in Ecology and Evolution*, **5**, 215-225.

Merow, C., Smith, M.J., Edwards, T.C., Guisan, A., McMahon, S.M., Normand, S., Thuiller, W., Wüest, R.O., Zimmermann, N.E. & Elith, J. (2014b) What do we gain from simplicity versus complexity in species distribution models? *Ecography*, **37**, 1267-1281.

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.

- Millar, C.S. & Blouin-Demers, G. (2012) Habitat suitability modelling for species at risk is sensitive to algorithm and scale: A case study of Blanding's turtle, *Emydoidea blandingii*, in Ontario, Canada. *Journal for Nature Conservation*, **20**, 18-29.
- Moen, A. (1999) National atlas of Norway: vegetation. Norwegian Mapping Authority, Hønefoss, Norway.
- Moreno-Amat, E., Mateo, R.G., Nieto-Lugilde, D., Morueta-Holme, N., Svenning, J.-C. & García-Amorena, I. (2015) Impact of model complexity on cross-temporal transferability in Maxent species distribution models: An assessment using paleobotanical data. *Ecological Modelling*, **312**, 308-317.
- Mörtberg, U.M., Balfors, B. & Knol, W.C. (2007) Landscape ecological assessment: A tool for integrating biodiversity issues in strategic environmental assessment and planning. *Journal of Environmental Management*, **82**, 457-470.
- Naimi, B. (2015) On uncertainty in species distribution modelling. PhD, University of Twente.
- Naimi, B. & Araújo, M.B. (2016) sdm: a reproducible and extensible R platform for species distribution modelling. *Ecography*, n/a-n/a.
- Nazeri, M., Madani, N., Kumar, L., Salman Mahiny, A. & Kiabi, B.H. (2015) A geo-statistical approach to model Asiatic cheetah, onager, gazelle and wild sheep shared niche and distribution in Turan biosphere reserve-Iran. *Ecological Informatics*, **29**, Part 1, 25-32.
- Newbold, T. (2010) Applications and limitations of museum data for conservation and ecology, with particular attention to species distribution models. *Progress in Physical Geography*, **34**, 3-22.
- Niamir, A. (2014) Knowledge Oriented Species Distribution Modelling. Doctoral dissertation, University of Twente, ITC.
- Norderhaug, A., Halvorsen, R., Johansen, L., Mazzoni, S., Bratli, H., Svalheim, E., Jordal, J.B. & Pedersen, O. (2012) Kulturmarkseng i Naturindeks – utvikling av kunnskapsgrunnlaget for overvåking og forvaltning. pp. 173. Bioforsk.
- Ohlson, M., Söderström, L., Hörnberg, G., Zackrisson, O. & Hermansson, J. (1997) Habitat qualities versus long-term continuity as determinants of biodiversity in boreal old-growth swamp forests. *Biological Conservation*, **81**, 221-231.
- Ottaviani, D., Lasinio, G.J. & Boitani, L. (2004) Two statistical methods to validate habitat suitability models using presence-only data. *Ecological Modelling*, **179**, 417-443.
- Parviainen, M., Zimmermann, N.E., Heikkinen, R.K. & Luoto, M. (2013) Using unclassified continuous remote sensing data to improve distribution models of red-listed plant species. *Biodiversity and Conservation*, **22**, 1731-1754.
- Pawitan, Y. (2001) *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press, Oxford, UK.
- Peres-Neto, P.R. (2006) A unified strategy for estimating and controlling spatial, temporal and phylogenetic autocorrelation in ecological models. *Oecologia Brasiliensis*, **10**, 7.
- Peres-Neto, P.R. & Legendre, P. (2010) Estimating and controlling for spatial structure in the study of ecological communities. *Global Ecology and Biogeography*, **19**, 174-184.
- 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.
- Petzoldt, T. & Rinke, K. (2007) simecol: An Object-Oriented Framework for Ecological Modeling in R. *Journal of Statistical Software*, **22**, 1-311.
- Phillips, S.J. (2010) A brief tutorial on Maxent. American Museum of Natural History, New York., New York.
- Phillips, S.J., Anderson, R.P. & Schapire, R.E. (2006) Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, **190**, 231-259.

- 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., Elith, J., Graham, C.H., Lehmann, A., Leathwich, J.R. & Ferrier, S. (2009) Sample selection bias and presence-only distribution models: implications for background and pseudo-absence data. *Ecological Applications*, **19**, 181-197.
- Phillips, S.J., Dudík, M. & Schapire, R. (2004) A maximum entropy approach to species distribution modeling. *Proceedings of the 21st international conference on machine learning* (ed. Anonymous), pp. 655-662. ACM Press, New York.
- Plant, R.E. (2012) *Spatial data analysis in ecology and agriculture using R*. CRC Press Taylor & Francis Group, Boca Raton, Florida.
- Polce, C., Termansen, M., Aguirre-Gutiérrez, J., Boatman, N.D., Budge, G.E., Crowe, A., Garratt, M.P., Pietravalle, S., Potts, S.G., Ramirez, J.A., Somerwill, K.E. & Biesmeijer, J.C. (2013) Species Distribution Models for Crop Pollination: A Modelling Framework Applied to Great Britain. *PLoS ONE*, **8**, e76308.
- Pontius, R.G. & Parmentier, B. (2014) Recommendations for using the relative operating characteristic (ROC). *Landscape Ecology*, **29**, 367-382.
- R Core Team (2014) R: A Language and Environment for Statistical Computing.
- Radosavljevic, A. & Anderson, R.P. (2014) Making better Maxent models of species distributions: complexity, overfitting and evaluation. *Journal of Biogeography*, **41**, 629-643.
- Rahn, K.-H., Klatt, S., Haas, E. & Butterbach-Bahl, K. (2013) Improving efficiency of a statistical analysis of complex ecological models, when using the statistical software R by parallelising tasks with Rmpi. *Ecological Informatics*, **15**, 53-57.
- Reineking, B. & Schröder, B. (2006) Constrain to perform: Regularization of habitat models. *Ecological Modelling*, **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.
- Rinde, E., Tjomsland, T., Hjermann, D., Kempa, M., Norling, P. & Kolluru, V.S. (2015) Spreading of the invasive Pacific oyster (*Crassostrea gigas*) in the Oslofjord - increased risk due to warmer climate? *Marine and Freshwater Research*.
- Rocchini, D. (2013) Seeing the unseen by remote sensing: satellite imagery applied to species distribution modelling. *Journal of Vegetation Science*, **24**, 209-210.
- Rodríguez-Rey, M., Jiménez-Valverde, A. & Acevedo, P. (2013) Species distribution models predict range expansion better than chance but not better than a simple dispersal model. *Ecological Modelling*, **256**, 1-5.
- Sandman, A.N., Wikström, S.A.A., Blomqvist, M., Kautsky, H. & Isaeus, M. (2013) Scale-dependent influence of environmental variables on species distribution: a case study on five coastal benthic species in the Baltic Sea. *Ecography*, **36**, 354-363.
- Santika, T. & Hutchinson, M.F. (2009) The effect of species response form on species distribution model prediction and inference. *Ecological Modelling*, **220**, 2365-2379.
- Schibalski, A., Lehtonen, A. & Schröder, B. (2014) Climate change shifts environmental space and limits transferability of treeline models. *Ecography*, **37**, 321-335.
- Shcheglovitova, M. & Anderson, R.P. (2013) Estimating optimal complexity for ecological niche models: A jackknife approach for species with small sample sizes. *Ecological Modelling*, **269**, 9-17.
- Silva, D.P., Gonzalez, V.H., Melo, G.A.R., Lucia, M., Alvarez, L.J. & de Marco Jr, P. (2014) Seeking the flowers for the bees: integrating biotic interactions into niche models to assess the distribution of the exotic bee species *Lithurgus huberi* in South America. *Ecological Modelling*, **273**, 200-209.
- Silvert, W. (1993) Object-oriented ecosystem modelling. *Ecological Modelling*, **68**, 91-118.

- Sokal, R. & Rohlf, F. (2012) *Biometry (4th edn)*. WH Freeman and Company, New York.
- Song, W., Kim, E., Lee, D., Lee, M. & Jeon, S.-W. (2013) The sensitivity of species distribution modeling to scale differences. *Ecological Modelling*, **248**, 113-118.
- Stasch, C., Scheider, S., Pebesma, E. & Kuhn, W. (2014) Meaningful spatial prediction and aggregation. *Environmental Modelling & Software*, **51**, 149-165.
- Stewart-Koster, B., Boone, E.L., Kennard, M.J., Sheldon, F., Bunn, S.E. & Olden, J.D. (2013) Incorporating ecological principles into statistical models for the prediction of species' distribution and abundance. *Ecography*, **36**, 342-353.
- Stokland, J.N., Bakkestuen, V., Bekkby, T., Rinde, E., Skarpaas, O., Sverdrup-Thygeson, A., Yoccoz, N.G. & Halvorsen, R. (2008) Prediksjonsmodellering av arters og naturtypers utbredelse og forekomst: utfordringer og potensiell bruksverdi i Norge. pp. 1-72. Natural History Museum, University of Oslo, Oslo, Norway.
- Stokland, J.N., Halvorsen, R. & Støa, B. (2011) Species distribution modelling—Effect of design and sample size of pseudo-absence observations. *Ecological Modelling*, **222**, 1800-1809.
- Stoklosa, J., Daly, C., Foster, S.D., Ashcroft, M.B. & Warton, D.I. (2015) A climate of uncertainty: accounting for error in climate variables for species distribution models. *Methods in Ecology and Evolution*, **6**, 412-423.
- Suárez-Seoane, S., Virgós, E., Terroba, O., Pardavila, X. & Barea-Azcón, J.M. (2014) Scaling of species distribution models across spatial resolutions and extents along a biogeographic gradient. The case of the Iberian mole *Talpa occidentalis*. *Ecography*, **37**, 279-292.
- Swab, R.M., Regan, H.M., Keith, D.A., Regan, T.J. & Ooi, M.K.J. (2012) Niche models tell half the story: spatial context and life-history traits influence species responses to global change. *Journal of Biogeography*, no-no.
- Teetor, P. (2011) *R Cookbook*. O'Reilly Media, Sebastopol, California.
- Telenius, A. (2011) Biodiversity information goes public: GBIF at your service. *Nordic Journal of Botany*, **29**, 378-381.
- Ter Braak, C.J. & Prentice, I.C. (1988) A theory of gradient analysis. *Advances in Ecological Research*, **18**, 271-317.
- Thibaud, E., Petitpierre, B., Broennimann, O., Davison, A.C. & Guisan, A. (2014) Measuring the relative effect of factors affecting species distribution model predictions. *Methods in Ecology and Evolution*, **5**, 947-955.
- Thuiller, W., Albert, C., Araújo, M.B., Berry, P.M., Cabeza, M., Guisan, A., Hickler, T., Midgely, G.F., Paterson, J., Schurr, F.M., Sykes, M.T. & Zimmermann, N.E. (2008) Predicting global change impacts on plant species' distributions: future challenges. *Perspectives in Plant Ecology, Evolution and Systematics*, **9**, 137-152.
- 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. (1996) Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, pp. 267-288.
- Tognelli, M.F., Roig-Juñent, S.A., Marvaldi, A.E., Flores, G.E. & Lobo, J.M. (2009) An evaluation of methods for modelling distribution of Patagonian insects. *Revista Chilena Historia Natural*, **82**, 347-360.
- Václavík, T. & Meentemeyer, R.K. (2012) Equilibrium or not? Modelling potential distribution of invasive species in different stages of invasion. *Diversity and Distributions*, **18**, 73-83.
- Vale, M.M. & Jenkins, C.N. (2012) Across-taxa incongruence in patterns of collecting bias. *Journal of Biogeography*, **39**, 1744-1748.
- Van der Loo, M.P. (2012) *Learning RStudio for R statistical computing*. Packt Publishing Ltd, Dallas, TX, USA.

- Van Looy, K., Piffady, J., Cavillon, C., Tormos, T., Landry, P. & Souchon, Y. (2014) Integrated modelling of functional and structural connectivity of river corridors for European otter recovery. *Ecological Modelling*, **273**, 228-235.
- van Son, T.C. & Halvorsen, R. (2014) Multiple Parallel Ordinations: the Importance of Choice of Ordination Method and Weighting of Species Abundance Data. *Sommerfeltia*, pp. 1.
- van Zonneveld, M., Castañeda, N., Scheldeman, X., van Etten, J. & van Damme, P. (2014) Application of consensus theory to formalize expert evaluations of plant species distribution models. *Applied Vegetation Science*, **17**, 528-542.
- Vaughan, I.P. & Ormerod, S.J. (2003a) Improving the Quality of Distribution Models for Conservation by Addressing Shortcomings in the Field Collection of Training Data. *Conservation Biology*, **17**, 1601-1611.
- Vaughan, I.P. & Ormerod, S.J. (2003b) Improving the Quality of Distribution Models for Conservation by Addressing Shortcomings in the Field Collection of Training Data
- Mejorando la Calidad de los Modelos de Distribución para la Conservación Enfocando en las Limitaciones en la Recolección a Campo de Datos de Capacitación. *Conservation Biology*, **17**, 1601-1611.
- Veloz, S.D. (2009) Spatially autocorrelated sampling falsely inflates measures of accuracy for presence-only niche models. *Journal of Biogeography*, **36**, 2290-2299.
- Verbruggen, H., Tyberghein, L., Belton, G.S., Mineur, F., Jueterbock, A., Hoarau, G., Gurgel, C.F.D. & De Clerck, O. (2013) Improving Transferability of Introduced Species' Distribution Models: New Tools to Forecast the Spread of a Highly Invasive Seaweed. *PLoS ONE*, **8**, e68337.
- Visser, M.D., McMahon, S.M., Merow, C., Dixon, P.M., Record, S. & Jongejans, E. (2015) Speeding Up Ecological and Evolutionary Computations in R; Essentials of High Performance Computing for Biologists. *PLoS Computational Biology*, **11**, e1004140.
- Walls, R.L., Deck, J., Guralnick, R., Baskauf, S., Beaman, R., Blum, S., Bowers, S., Buttigieg, P.L., Davis, N., Endresen, D.T.F., Gandolfo, M.A., Hanner, R., Janning, A., Krishtalka, L., Matsunaga, A., Midford, P., Morrison, N., Ó Tuama, É., Schildhauer, M., Smith, B., Stucky, B.J., Thomer, A., Wiczorek, J., Whitacre, J. & Wooley, J. (2014) Semantics in Support of Biodiversity Knowledge Discovery: An Introduction to the Biological Collections Ontology and Related Ontologies. *PLoS ONE*, **9**, 1-13.
- Warren, D.L., Glor, R.E. & Turelli, M. (2010) ENMTools: a toolbox for comparative studies of environmental niche models. *Ecography*, **33**, 607-611.
- Warren, D.L. & Seifert, S.N. (2011) Ecological niche modeling in Maxent: the importance of model complexity and the performance of model selection criteria. *Ecological Applications*, **21**, 335-342.
- Warton, David I. (2015) New opportunities at the interface between ecology and statistics. *Methods in Ecology and Evolution*, **6**, 363-365.
- Whittaker, R.H. (1967) Gradient Analysis of Vegetation. *Biological reviews*, **42**, 207-264.
- Wilson, J.S. & Pitts, J.P. (2012) Identifying Pleistocene refugia in North American cold deserts using phylogeographic analyses and ecological niche modelling. *Diversity and Distributions*, **18**, 1139-1152.
- Wisn, M.S., Hijmans, R.J., Li, J., Peterson, A.T., Graham, C.H., Guisan, A. & NCEAS Predicting Species Distributions Working Group (2008) Effects of sample size on the performance of species distribution models. *Diversity and Distributions*, **14**, 763-773.
- 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 in Ecology and Evolution*, **4**.
- Zimmermann, N.E., Edwards, T.C., Moisen, G.G., Frescino, T.S. & Blackard, J.A. (2007) Remote sensing-based predictors improve distribution models of rare, early successional and broadleaf tree species in Utah. *Journal of Applied Ecology*, **44**, 1057-1067.
- Økland, R. (2007) Wise use of statistical tools in ecological field studies. *Folia Geobotanica*, **42**, 123-140.

Supplementary Materials

S1. MIAT Toolbox Code

```
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
10 #####
11 #####
12 #####
13 #####
14 #####
15 #####
16 #####
17 #####
18 #####
19 #####
20 #####
21 #####
22 #####
23 #####
24 #####
25 #####
26 #####
27 #####
28 #####
29 #####
30 #####
31 #####
32 #####
33 #####
34 #####
35 #####
36 #####
37 #####
38 #####
39 #####
40 #####
41 #####
42 #####
43 #####
44 #####
45 #####
46 #####
47 #####
48 #####
49 #####
50 #####
51 #####
52 #####
```

MIAT

Modular Integrated Approach Toolbox

Modular R-wrappers to Distribution Modelling Practice

by

Sabrina Mazzoni
sabgs@yahoo.com
2013-2015

GECO - Geo Ecology group
<http://www.nhm.uio.no/english/research/groups/geco/>
Natural History Museum - University of Oslo

Module 1: Data and model definition

Specification of the data model, statistical model and modelling tools settings or parameters

(M1_a_SpecPar.R)

M1a requires users to run each section of executable code (one or more line) manually to input the key parameters.

Users are urged to run through the scripts section by section to also understand the process better, at least initially.

1. Make sure you have a access to installing libraries and a copy of the executable MAXENT software file, Maxent.jar, into the working directory.

2. The toolbox is currently coded to run on WINDOWS machines, as it uses WINDOWS batch (bat) files and uses backslashes (\) to denote directories.

3. Make sure you have a current version of JAVA, and that the system's environment variables (Path and PATH-EXT) include the necessary information to access it.

4. Open and save the script file contained below using R or any text editor; start an R session, select all lines of the script and source (run) it.

You will see one pop-up window asking for the location of the working directory, i.e. the directory that contains the data and files.

The pop-up window may be hiding behind the current R window.

5. Activate it (by clicking on the icon), and navigate (or cut and paste the path) to the working directory as instructed above. After that, just sit back and watch!

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) and are either at the back of the code or on top of a section

Throughout the scripts, you will also find Section-specific 'commented-out inspection code' preceded by (#_InsCode)

This is simply a series of code such as str(object name)' or 'head(object name)' that enable the user to inspect the process along the way.

They have a hash symbol (#_InsCode) in front and are, accordingly, not run unless the hash symbol is removed. They are provided both for convenience and as guidance. Users are urged to temporarily remove the hash symbol to explore the process and the data along the way.

Additionally, you will also find optional code (#_OptCode), as well as alternative lines of code that can be used to test/develop tools further (#_AltCode)

```
53 #####
54 #####
55 #####
56 #####
57 #####
58 #####
59 #####
60 #####
61 #####
62 #####
63 #####
64 #####
65 #####
66 #####
67 #####
68 #####
69 #####
70 #####
71 #####
72 #####
73 #####
74 #####
75 #####
76 #####
77 #####
78 #####
79 #####
80 #####
81 #####
82 #####
83 #####
84 #####
85 #####
86 #####
87 #####
88 #####
89 #####
90 #####
91 #####
92 #####
93 #####
94 #####
95 #####
96 #####
97 #####
98 #####
99 #####
100 #####
101 #####
102 #####
103 #####
104 #####
105 #####
106 #####
107 #####
108 #####
109 #####
110 #####
```

NOTE on terms and code abbreviations

Some of the code abbreviations refer to old terminology, however, the descriptive comments use, as much as possible, the latest terms.

Thus, abbreviations like P, stood for "predictor", but are now simply referred to as "variable", "derived variable" or "significant variable"

This is a set of exploratory data and modelling tools:MIAT. \nMlt will allow you to load raw explanatory variables (either in raster or vector format) and response variables; explore their distribution, apply rescaling and several transformations for use as future input to other tools (MaxEnt and or GIS)

... oh, and I guess do just a few other things! ... like building by variable manual forward selection (without the manual part)"

. fill=TRUE.sep=""

You will be asked a few questions at the beginning, place the cursor on the console then type. The rest the tools will take care of it!", fill=TRUE.sep=""

Please place your input data in one folder (which you will be asked to identify), along with the latest copy of the maxent.jar file.

Within this folder, place explanatory variables in 2 separate folders: 1 for continuous and one for categorical data (if the latter is present)!", fill=TRUE.sep=""

Several options for loading input the data for these tools have been scripted. This script is for loading Maxent software standard practice data, in ASCII and comma separated text files. Please refer contact developers for additional data input format scripts', fill=TRUE.sep=""

ScriptNameCode <- "M1a" # script name code used to identify where you are in the process

M1a: Data and model definition

Defining key starting parameters

and grouping them into one key object

MIAPar

Several types of parameters:

- Parameters of the Tools/Scripts
- Parameters of the Transformations/Features (mathematical model)
- Parameterisation of the statistical model
- Parameters of the "Maxent runs" and results(from each new model)

MIAParNames <- c("dir.Rlib", # R library location to load packages
"dir.Rlib.num", # R library location choice (a numeric value the user chooses after all libraries are listed)
"DAC", # Dataset Abbreviation Code, will be used as a prefix to results, (handy if running tools on multiple datasets)
"dir.in", # Starting directory, where to locate the Input Data, and subsequently place all results files and folders
"inType", # This version works with ASCII and csv files as starting Input, other options are possible
user is asked to split the Explanatory Variables in Continuous and Categorical and identify folder
"dir.ev.cnt", # Continuous Environmental Data Folder Name/location
"dir.ev.cat", # Categorical Environmental Data Folder Name/location
"EVCntNames", # Names of CONTINUOUS Variables Only
"EVCatNames", # Names of CATEGORICAL Variables Only
"EVAInNames", # Name of all Environmental Variables
"TrainFile", # user is asked for input training file to be chosen by list
"TestFile", # user is asked for input test data file if present, otherwise 0
"RVNames", # Response Variable Names
"SubSampleSize", # used in the subsampling of data if records more than 500,000. User can change default
"BTGFile", # Background Target Group files
"Mx8kgNum", # Maximum number of background points to be selected to build MaxEnt model with - 10,000
maxent.jar default
"qn.meth", # Quantile classes to build deviation predictors - 10, 20, 100 etc
"DevExp", # deviation exponents: default: c(5,1,2)
"num.smoothing", # number of smoothing iterations (3 default)
"Knots", # num knots (iterations) for Spline objects - readline input the number of knots to inspect
"KValues", # KValue at knot position i # a dataframe with both the n.iterations (the knot), and the value at the knot ()
"AlphaValues", # internal model assessment/improvement criteria: significance value used in F-test of model


```
improvement
"RegulValues" # value for the regularization parameter, used in shrinkage (lasso penalty) model selection
)
M1aPar <- as.list(!:length(M1aParNames))
names(M1aPar) <- M1aParNames
rm(M1aParNames) # users can decide if to keep or not

M1aPar$SubSampleSize <- 500000 #500000 # used in the subsampling of data if too many records or ask user
M1aPar$MxBkgNum<-10000 # Default as in maxent.jar or ask user
#_AltCode M1aPar$AlphaValues <- as.vector(c(0.050,0.000001)) #_AltCode
as.vector(c(0.250,0.100,0.050,0.010,0.001,0.00001,0.0000001,0.0000000000001))
M1aPar$AlphaValues <- as.vector(c(0.100,0.050,0.010,0.001,0.0001,0.00001,0.000001,0.0000000000001)) #
The model selection/assessment criteria, (p value for nested comparison forward selection)
#_AltCode M1aPar$RegulValues <- as.vector(c(0.1,8,16)) # rz will be the counter for regularization value #MORE COMPLETE
M1aPar$RegulValues <- as.vector(c(0.5,1.2,4,8,16)) # rz will be the counter for regularization value
M1aPar$gn.meth <- 100 # the number of quantile frequency classes: Default setting, or ask user, other (20)
M1aPar$num.smoothing <- 3 # number of smoothings, Default: 5 or ask user how many to do
M1aPar$DevExp <- c(5,1,2) # deviation exponents default or ask user to do so
M1aPar$Knots <- 20 # num knots (iterations) for $line objects - readline input the number of knots to inspect
#####
##### Knots and KValues for Splines ###
#####
#####
i=1 # i for iteration
for (i in 1:M1aPar$Knots)
{
  M1aPar$KValues[i]=as.vector(c(2*1-1)/(2*M1aPar$Knots)) # KValue at knot position i # a dataframe with both the
  n.iterations (the knot), and the value at the knot ( )
}
M1aPar$KValues
M1aPar$Knots
rm(i) # tidying
#####
#####
##### INTERACTIVE USER INPUT #####
##### Setting working directories and loading data ###
#####
#####
Rlib <- as.data.frame(libPaths()) # finds out which Rlibrary paths exist
Rlib
# shows it to user
cat("\nPlease place the cursor on the console, and type a number from the list shown above, identifying which library path do
you want use for the required packages to be installed?")
###
Space to indicate Next User Input
cat("\nWhich library path do you want use for the required packages to be installed? ")
dir.Rlib.num <- as.numeric(readline("\nWhich library path do you want use for the required packages to be installed? "))
this is useful if you have more than 1 path
dir.Rlib.num # stick this if they miss this line ... get the default 1
if(is.na(dir.Rlib.num)) == TRUE{dir.Rlib.num <- 1} # Default if not read properly
dir.Rlib.num
dir.Rlib <- as.character(Rlib[dir.Rlib.num,1])

###
Space to indicate Next User Input
cat("Next you'll choose the working directory -- all outputs will be created inside this folder.\n Make sure you also place the
maxent.jar file here too\n Pop up Window may hide behind others")
dir.in <- choose.dir("Choose the working directory -- all outputs will be created inside this folder.\n Make sure you also place
the maxent.jar file here too\n Pop up Window may hide behind others")
dir.in <- paste(dir.in, "\", sep="") # adding \\ to all directory objects consistently
setwd(dir.in) # necessary or it will use your default
getwd() # visually confirming it

###
Space to indicate Next User Input
```

```
cat("\n Please place the cursor on the console, and type a Data Abbreviation Code,(ie A01_).\nThis will be placed in front of
many objects created, useful if running multiple datasets or just to group your results. \n\nHit return if not relevant")
DAC <- readline("\nPlace the cursor on the console \" and type the Data Abbreviation Code here if you have one - (ie A01_).
\nsimply hit return if you don't have one\n (after placing cursor on console as other times) \n ")

DAC # visually confirming it

###
Space to indicate Next User Input
cat("\nNow you'll choose directory where your raw explanatory CONTINUOUS variables are located. \nHit cancel if not
relevant")
dir.ev.ent <- choose.dir(default=dir.in,caption="Pick CONTINUOUS explanatory data directory")

###
Space to indicate Next User Input
cat("\nNext you'll choose directory where your raw explanatory CATEGORICAL variables are located. \nHit cancel if not
relevant")
dir.ev.cat <- choose.dir(default=dir.in,"Pick CATEGORICAL explanatory data directory, hit cancel if not relevant")

###
Background Target Group ###
Space to indicate Next User Input
cat("\n Loading the Background Target Group Data: \nThis can be either a single *.csv file\n with one or more response
variables, or several individual response files.\n Note, format the same as for Maxent.jar: a three columns, Comma
Separated Values table. \nHit cancel if not relevant")
BTGFile <- choose.files(default = "",csv", caption= "Choose the Background Target Group Datafile")

###
Space to indicate Next User Input
cat("\n Loading the Training Data: This can be either a single *.csv file\n with one or more response variables, or several
individual response files.\n Note, format the same as for Maxent: Comma Separated Value, with three columns")
TrainFile <- choose.files(default = "",csv", caption= "Choose the Training Datafile")

###
Space to indicate Next User Input
cat("\n Finally pick the Test Data if you have a separate file. \n Hit cancel if not relevant")
TestFile <- choose.files(default = "",csv",caption= "Choose the Test Datafile, hit Cancel if none available")

###
next lines below are options for users to enter their own values

cat("\n Example of a simple way to additionally change some of the settings by simply removing the #_AltCode from the next
few lines")
cat("\n Number of Quantiles Classes and Smoothing iterations\n for use in Frequency of Response Curve plot")
#_AltCode M1aPar$gn.meth <- as.numeric(readline("Type 10 for deciles or 100 for percentile")) # interactive or ask to
accept default
#_AltCode M1aPar$num.smoothing <- as.numeric(readline("Type the number of smoothing iterations desired (i.e. 3) ")) #
interactive or ask to accept default
#_AltCode MxBkgNum <- as.numeric(readline("Type number of random background points you want Maxent to use"))
#10000 is default=Length(Train.SDP[[rv]]) # Decide what is the number of "sample" background points you want Maxent
to use

#####
##### Saving User Settings #####
#####
This leaves open for you to keep all in 1 list or loose ###
#####
dir.ev.ent <- paste(dir.ev.ent, "\", sep="") # helps so you don't have to remember ... for all directories
if (is.na(dir.ev.cat) == FALSE) {dir.ev.cat <- paste(dir.ev.cat, "\", sep="")} # helps so you don't have to remember ... for all
directories

getwd()
M1aPar$dir.Rlib <- dir.Rlib
M1aPar$DAC <- DAC # Save this onto your "parameters object list"
M1aPar$dir.Rlib.num <- dir.Rlib.num
M1aPar$dir.in <- dir.in # Save this onto your "parameters object list"
M1aPar$dir.ev.ent <- dir.ev.ent
M1aPar$dir.ev.cat <- dir.ev.cat
M1aPar$BTGFile <- BTGFile
M1aPar$TrainFile <- TrainFile
M1aPar$TestFile <- TestFile
names(M1aPar)
M1aPar # Visually check everything is as it should be

###
Assign this BTG Flag
```



```
226
227 if (length(MIAPar$BTGFile)==0) {BTGFlag<-FALSE}
228 if (length(MIAPar$BTGFile)>0) {BTGFlag<-TRUE} #TRUE
229
230 ## improve option of "if BTGFlag, then pick accordingly
231
232 #####
233 ###
234 ##
235 ##      Setting the maxent.jar "settings", parameters, flags, options
236 #####
237
238 ## Breaking down into a "series of simple text objects" that will be "pasted together"
239 ## to make up the "command file" that will define the maxent.jar settings
240 ## Simple set up for now is ability to choose certain objects or not
241 ## Users should consult the maxent.jar help (or the alphabetically sorted table provided here)
242 ## for more info on these maxent.jar flags
243 ## Depending on the script, they are selected and pasted together (with or without a space)
244 ## Make sure there's a space (and no more than 1) at the appropriate place
245
246 ## As we have moved all the "set up specs" checks in the first module, include this here, so you can run m2g, m3 etc. all
    without having to check
247
248 JarFileFlag <- paste(dir.in, "maxent.jar", sep="")
249 ##> checking if maxent can be run (normally not part of your script)
250 if (file.exists(JarFileFlag)==FALSE){cat("!!!! WARNING !!!!\nPlease place a copy of maxent.jar in your dir.in directory\n!!!!\n
    WARNING !!!!)} else {"All's good, carry on, carry on"}
251
252
253 JarFileFlag
254
255 ##> this is now its own Parameter Table/object for Maxent.jar Flags, but may simplify this when functions are made
256 ##
257
258 JarPar <- list()
259
260 ##> to integrate this if desired later: M2g_MaxSetting <- "maxent.jar threads=6 pictures=FALSE
    writebackgroundpredictions=FALSE autofeature=FALSE beta_categorical=0 beta_hinge=0 beta_lqp=0 betamultiplier=0
    beta_threshold=0 dodamp=FALSE extrapolate=FALSE hinge=FALSE jackknife=FALSE nowarnings outputformat=raw
    outputgrids=FALSE perspectives=TRUE product=FALSE quadratic=FALSE randomtestpoints=0 redofixits=TRUE
    removeduplicates=TRUE responsecurves=FALSE responsecurveexponent=FALSE skipfixits=FALSE threshold=FALSE
    visible=FALSE writelotdata=FALSE samplefile="
261
262 JarPar$JarFlag_AM1 <- " autofeature=FALSE beta_categorical=0 beta_hinge=0 beta_lqp=0 betamultiplier=0
    beta_threshold=0 linear=TRUE hinge=FALSE product=FALSE quadratic=FALSE threshold=FALSE" # This is starting point
    for F55 approach, (no shrinkage)removing Lregularisation and "autofeature" approach altogether
263 JarPar$JarFlag_AM2 <- " addallsamplestobackground=FALSE addsamplestobackground=FALSE
    maximumbackground=200000" # has to be false as 1 includ n in N already
264 JarPar$JarFlag3 <- " dodamp=FALSE extrapolate=FALSE jackknife=TRUE responsecurves=FALSE perspectives=TRUE
    randomtestpoints=0 responsecurveexponent=FALSE"
265 JarPar$JarFlag4 <- " adjustsamplerradius=-1 outputformat=raw outputgrids=TRUE writebackgroundpredictions=TRUE
    writelotdata=TRUE"
266 JarPar$JarFlag5 <- " nowarnings redofixits=TRUE removeduplicates=TRUE skipfixits=FALSE threads=6" # Stuck visible at
    the end, easier to fin and edit after visibles=FALSE" # These are general "housekeeping bits", choose accordingly
267
268 JarPar$JarFlag_SM1a <- " addallsamplestobackground=TRUE addsamplestobackground=TRUE maximumbackground=10000"
    ## Ensures N is Backgrnd+n, when n is not necessarily in Backgrnd
269 JarPar$JarFlag_SM1b <- " addallsamplestobackground=FALSE addsamplestobackground=TRUE maximumbackground=10000"
    ## Ensures N is no greater than Backgrnd+n, when n is by definition part of Backgrnd as in Background Target Group or
    Total Background points same as extent/N
270 ## Currently the rest is set in the 'very flexible' M3_SM module
271 ##> Like this ...
272 JarPar$FTCode <- c("NI", "W1") # Feature Types chosen ... set ... NI for No Interactions, W1 for With Interactions, set these 2
    for now by hand, can add other parameters too
273 JarPar$MaxSet_SM_FT <- c(" autofeature=FALSE product=FALSE", autofeature=TRUE)" ## just the bit that specifies the
    feature type, matches the vector FT, and don't forget the space before! rudimentary again yes but it works!
274
275 ## users can choose which ones to put together
276 ## very rudimentary example, more options to be developed
277
```

```
278 JarPar$M3A_JarFlags <- paste(JarPar$JarFlag_AM1,JarPar$JarFlag_AM2,JarPar$JarFlag3,JarPar$JarFlag4,JarPar$JarFlag5, sep
    =")
279 JarPar$M3B_JarFlags <- paste(JarPar$JarFlag_AM1,JarPar$JarFlag_AM2,JarPar$JarFlag3,JarPar$JarFlag4,JarPar$JarFlag5, sep=
    =")
280 JarPar$M3C_JarFlags <- paste(JarPar$JarFlag_AM1,JarPar$JarFlag_AM2,JarPar$JarFlag3,JarPar$JarFlag4,JarPar$JarFlag5, sep
    =")
281 JarPar$M3SM_JarFlags <- paste(JarPar$JarFlag_SM1a,JarPar$JarFlag3,JarPar$JarFlag4,JarPar$JarFlag5, sep=(""))
282
283 JarPar # visual chec
284
285 MaxRunBat <-paste("java -Xmx512m -jar ",
286 JarFileFlag,
287 JarPar$JarFlag_SM1a,
288 #_OptCode " visible=FALSE autorun",
289 sep="")
290
291 cat("if maxent.jar windows pops up, all is good carry on (after closing jar window), otherwise \nresolve issues with OS, java in
    order for the maxent.jar section of the scripts to work properly, \nMay be issues of memory, system variables and
    paths.between Java and OS")
292
293 system(paste(MaxRunBat),wait=TRUE) ## if maxent.jar pops up, all is good and your os, java and the scripts are working
    together
294
295 #####
296 ###
297 ###> creating location names and directories ###
298 ##> Keeping them separate for now, could group them in a directory parameter list ##
299 ##> Some are remmed out as the are not currently used, marked with # AltCode in front, but could be used again ###
300 #####
301
302 # AltCode dir.scripts <- paste(dir.in,"Scripts\\",sep="") # scripts here
303 dir.rdata <- paste(dir.in,"RData\\",sep="") # location of R's datafiles
304 dir.ascdi.out <-paste(dir.in,"R_OUT\\ASCDI\\", sep="")
305 dir.r.out<-paste(dir.in,"R_OUT\\", sep="")
306 dir.swd<-paste(dir.in,"R_OUT\\SWDPlots\\", sep="") # After RV has been overlayed and cleaned up of dups, and only those
    within the Data Exent
307 dir.plot.out<-paste(dir.in,"R_OUT\\PLOTS\\", sep="")
308 dir.fct<-paste(dir.r.out,"FBin\\", sep="") #
309 dir.knots<-paste(dir.r.out,"KNOTS\\", sep="")
310 # AltCode dir.hr<-paste(dir.r.out,"KNOTS\\HRF\\", sep="") # useful if you want to spit out individual csv for all splines --
    user to choose
311 # AltCode dir.hr<-paste(dir.r.out,"KNOTS\\HRN\\", sep="")
312 # AltCode dir.th<-paste(dir.r.out,"KNOTS\\TH\\", sep="")
313 dir.max<-paste(dir.in,"AM\\", sep="") # AM stands for Alternative Maxent, user decides, but this is good alphabetically
314
315 ##> Create these directories in the next script, so that you don't have to specify parameters again
316
317 MIAVect <- objects(pattern="MIAPar")
318 DirVect <- objects(pattern="dir") # currently all scripts run on indiv dir object, but could upgrade to use this
319 CSVVect1<- objects(pattern="Train") # Choosing numbers rather than Train, otherwise this get saved too
320 CSVVect2 <- objects(pattern="Test")
321
322 Mia_ParVect <- c(DirVect,MIAVect, #MIAPar", # Adding MIAPar to DirVect in order to save all parameters specified thus far,
    also this are the "initial" ones
    CSVVect1.CSVVect2,"DAC") # and location of input training and test data
323
324 #####
325 ##### Tidy and Saving Objects to RData workspaces #####
326 #####
327 m(CSVVect1.CSVVect2) # Some objects not needed to keep all the times
328 save(list=Mia_ParVect, # Saving MIAParameters and Directory objects as "RData" workspaces
    file=paste(dir.in,
    "Obj_", # distinguish single objects from entire Images
    ScriptNameCode, # which script? which step, part in the process etc
    "MIAParAll", # object name
    "RData",sep=""))
329
330 save.image(file=paste(dir.in,
    if(DAC>0)DAC,
    ScriptNameCode,
331
332
333
334
335
336
337
338
```

```
339 "end_"
340 "AllObj_KeyOnes", # object name
341 "RData",sep="")
342 #####
343 #####
344 ## next script ###
345
346
```



```
M1b_LoadData_MIA72015_V2.R
1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
10 #####
11 #####
12 #####
13 #####
14 #####
15 #####
16 #####
17 #####
18 ### NOTE on terms and abbreviations ###
19 ### Some of the code abbreviations refer to old terminology, however,
20 ### the descriptive comments use, as much as possible, the latest terms.
21 ### Thus, abbreviations like P, stood for "predicator", but are now simply referred to as
22 ### "variable", "derived variable" or "significant variable"
23 #####
24
25 # start with end of previous module
26 # AltCode load("A01_M1a_end_AltObj_KeyOnes.RData") # users need to edit this
27
28 # AltCode load(file=paste(dir.in,
29 # AltCode if(DAC>0){DAC},
30 # AltCode ScriptNameCode,
31 # AltCode end,"
32 # AltCode "_AltObj_KeyOnes", # object name
33 # AltCode .RData'.sep="") # Saving RData File
34
35
36 memory.limit(6000000) # set memory based on requirements
37 SaveTempFlag <- FALSE # Set to TRUE or FALSE For Saving temporary or portions of RData files from time to time. Useful
   when testing, or dealing with large files
38 # AltCode SaveTempFlag <- TRUE # Set to TRUE or FALSE For Saving temporary or portions of RData files from time to
   time, Useful when testing, or dealing with large files
39
40 setwd(dir.in) # necessary or it will use your default
41 getwd() # visually confirming it
42
43 #####
44 #####
45 #####
46 #####
47 #####
48 #####
49 ### Choose which packages to load and decide if to embed them in individual scripts or just 1
50
51 # AltCode MIAPar$dir.Rlib <- "C:/Program Files/R/R-3.0.0/library"
52 # options for installing packages if not already there ... hiding it in an IF loop now
53
54 #####
55 #####
56 ## M1b1 - Just Libraries section ###
57 #####
58 #####
59
60 ScriptNameCode <- "M1b1" # this can be just a "short bit of ScriptName
```

```
M1b_LoadData_MIA72015_V2.R
61
62 InstallFlag <- FALSE #TRUE# FALSE # used to install the default packages if not already there.
63 # AltCode InstallFlag <- TRUE
64 if (InstallFlag==TRUE) # # only instll if this flag is set to FALSE
65 {
66   install.packages("lattice",MIAPar$dir.Rlib)
67   install.packages("raster",MIAPar$dir.Rlib) #
68   install.packages("sp",MIAPar$dir.Rlib)
69   install.packages("rgdal",MIAPar$dir.Rlib)
70   install.packages("dummies",MIAPar$dir.Rlib) #
71   install.packages("e1071",MIAPar$dir.Rlib)
72   # install.packages("maptools",MIAPar$dir.Rlib)
73   install.packages("MASS",MIAPar$dir.Rlib)
74   install.packages("plyr",MIAPar$dir.Rlib)
75   install.packages("PresenceAbsence",MIAPar$dir.Rlib)
76   # install.packages("rJava",MIAPar$dir.Rlib)
77   install.packages("scales",MIAPar$dir.Rlib)
78   # install.packages("stats",MIAPar$dir.Rlib)
79   # install.packages("ROCR",MIAPar$dir.Rlib)
80   # install.packages("vegan",MIAPar$dir.Rlib)
81   # install.packages("dsmo",MIAPar$dir.Rlib)
82 }
83
84 require(lattice,MIAPar$dir.Rlib) # put this first since others require it
85 require(raster,MIAPar$dir.Rlib) # put this first since others require it
86 require(sp,MIAPar$dir.Rlib)
87 require(rgdal,MIAPar$dir.Rlib)
88 require(dummies,MIAPar$dir.Rlib) # used by the FBin Script
89 require(e1071,MIAPar$dir.Rlib) # used by the zeroskewness script, also Smooth Dev
90 #require(maptools,MIAPar$dir.Rlib) # dont know if I use it actually
91 require(MASS,MIAPar$dir.Rlib)
92 require(plyr,MIAPar$dir.Rlib) # used by all the Joining portions of the scripts
93 require(PresenceAbsence,MIAPar$dir.Rlib) # Used in Mod 5
94 # require(rJava,MIAPar$dir.Rlib)
95 require(scales,MIAPar$dir.Rlib) # used by the Lin and other scripts Smooth and Dev
96 # require(dsmo,MIAPar$dir.Rlib)
97 # require(stats,MIAPar$dir.Rlib)
98 # require(vegan,MIAPar$dir.Rlib)
99 # require(ROCR,MIAPar$dir.Rlib)
100
101 #####
102 #####
103 ### Creating Starting directories ###
104 #####
105 #####
106 dir.create(dir.r.out)
107 dir.create(dir.swd)
108 # dir.create(dir.qn)
109 # dir.create(dir.scripts)
110 dir.create(dir.r.data)
111 dir.create(dir.ascli.out)
112 dir.create(dir.plot.out)
113 dir.create(dir.fct)
114 dir.create(dir.knots)
115 # dir.create(dir.hf)
116 # dir.create(dir.hr)
117 # dir.create(dir.th)
118 dir.create(dir.max) # keeping max for now, but folder is AM, consider changing
119
120 save.image(file=paste(dir.r.data,
121   if(DAC>0){DAC},
122   ScriptNameCode,
123   ".End", # object name
124   ".RData".sep="")) # Saving RData File
125
126 #####
127 #####
128 #####
129 #####
130 M1b LoadData - Part 2
131 Loading EV layers, Training and Test data,
132 #####
133 #####
134 #####
135 #####
136 #####
137 #####
138 #####
139 #####
140 #####
141 #####
142 #####
143 #####
144 #####
145 #####
146 #####
147 #####
148 #####
149 #####
150 #####
151 #####
152 #####
153 #####
154 #####
155 #####
156 #####
157 #####
158 #####
159 #####
160 #####
161 #####
162 #####
163 #####
164 #####
165 #####
166 #####
167 #####
168 #####
169 #####
170 #####
171 #####
172 #####
173 #####
174 #####
175 #####
176 #####
177 #####
178 #####
179 #####
180 #####
181 #####
182 #####
183 #####
184 #####
185 #####
186 #####
187 #####
188 #####
189 #####
190 #####
191 #####
192 #####
193 #####
194 #####
195 #####
196 #####
197 #####
198 #####
199 #####
200 #####
201 #####
202 #####
203 #####
204 #####
205 #####
206 #####
207 #####
208 #####
209 #####
210 #####
211 #####
212 #####
213 #####
214 #####
215 #####
216 #####
217 #####
218 #####
219 #####
220 #####
221 #####
222 #####
223 #####
224 #####
225 #####
226 #####
227 #####
228 #####
229 #####
230 #####
231 #####
232 #####
233 #####
234 #####
235 #####
236 #####
237 #####
238 #####
239 #####
240 #####
241 #####
242 #####
243 #####
244 #####
245 #####
246 #####
247 #####
248 #####
249 #####
250 #####
251 #####
252 #####
253 #####
254 #####
255 #####
256 #####
257 #####
258 #####
259 #####
260 #####
261 #####
262 #####
263 #####
264 #####
265 #####
266 #####
267 #####
268 #####
269 #####
270 #####
271 #####
272 #####
273 #####
274 #####
275 #####
276 #####
277 #####
278 #####
279 #####
280 #####
281 #####
282 #####
283 #####
284 #####
285 #####
286 #####
287 #####
288 #####
289 #####
290 #####
291 #####
292 #####
293 #####
294 #####
295 #####
296 #####
297 #####
298 #####
299 #####
300 #####
301 #####
302 #####
303 #####
304 #####
305 #####
306 #####
307 #####
308 #####
309 #####
310 #####
311 #####
312 #####
313 #####
314 #####
315 #####
316 #####
317 #####
318 #####
319 #####
320 #####
321 #####
322 #####
323 #####
324 #####
325 #####
326 #####
327 #####
328 #####
329 #####
330 #####
331 #####
332 #####
333 #####
334 #####
335 #####
336 #####
337 #####
338 #####
339 #####
340 #####
341 #####
342 #####
343 #####
344 #####
345 #####
346 #####
347 #####
348 #####
349 #####
350 #####
351 #####
352 #####
353 #####
354 #####
355 #####
356 #####
357 #####
358 #####
359 #####
360 #####
361 #####
362 #####
363 #####
364 #####
365 #####
366 #####
367 #####
368 #####
369 #####
370 #####
371 #####
372 #####
373 #####
374 #####
375 #####
376 #####
377 #####
378 #####
379 #####
380 #####
381 #####
382 #####
383 #####
384 #####
385 #####
386 #####
387 #####
388 #####
389 #####
390 #####
391 #####
392 #####
393 #####
394 #####
395 #####
396 #####
397 #####
398 #####
399 #####
400 #####
401 #####
402 #####
403 #####
404 #####
405 #####
406 #####
407 #####
408 #####
409 #####
410 #####
411 #####
412 #####
413 #####
414 #####
415 #####
416 #####
417 #####
418 #####
419 #####
420 #####
421 #####
422 #####
423 #####
424 #####
425 #####
426 #####
427 #####
428 #####
429 #####
430 #####
431 #####
432 #####
433 #####
434 #####
435 #####
436 #####
437 #####
438 #####
439 #####
440 #####
441 #####
442 #####
443 #####
444 #####
445 #####
446 #####
447 #####
448 #####
449 #####
450 #####
451 #####
452 #####
453 #####
454 #####
455 #####
456 #####
457 #####
458 #####
459 #####
460 #####
461 #####
462 #####
463 #####
464 #####
465 #####
466 #####
467 #####
468 #####
469 #####
470 #####
471 #####
472 #####
473 #####
474 #####
475 #####
476 #####
477 #####
478 #####
479 #####
480 #####
481 #####
482 #####
483 #####
484 #####
485 #####
486 #####
487 #####
488 #####
489 #####
490 #####
491 #####
492 #####
493 #####
494 #####
495 #####
496 #####
497 #####
498 #####
499 #####
500 #####
501 #####
502 #####
503 #####
504 #####
505 #####
506 #####
507 #####
508 #####
509 #####
510 #####
511 #####
512 #####
513 #####
514 #####
515 #####
516 #####
517 #####
518 #####
519 #####
520 #####
521 #####
522 #####
523 #####
524 #####
525 #####
526 #####
527 #####
528 #####
529 #####
530 #####
531 #####
532 #####
533 #####
534 #####
535 #####
536 #####
537 #####
538 #####
539 #####
540 #####
541 #####
542 #####
543 #####
544 #####
545 #####
546 #####
547 #####
548 #####
549 #####
550 #####
551 #####
552 #####
553 #####
554 #####
555 #####
556 #####
557 #####
558 #####
559 #####
560 #####
561 #####
562 #####
563 #####
564 #####
565 #####
566 #####
567 #####
568 #####
569 #####
570 #####
571 #####
572 #####
573 #####
574 #####
575 #####
576 #####
577 #####
578 #####
579 #####
580 #####
581 #####
582 #####
583 #####
584 #####
585 #####
586 #####
587 #####
588 #####
589 #####
590 #####
591 #####
592 #####
593 #####
594 #####
595 #####
596 #####
597 #####
598 #####
599 #####
600 #####
601 #####
602 #####
603 #####
604 #####
605 #####
606 #####
607 #####
608 #####
609 #####
610 #####
611 #####
612 #####
613 #####
614 #####
615 #####
616 #####
617 #####
618 #####
619 #####
620 #####
621 #####
622 #####
623 #####
624 #####
625 #####
626 #####
627 #####
628 #####
629 #####
630 #####
631 #####
632 #####
633 #####
634 #####
635 #####
636 #####
637 #####
638 #####
639 #####
640 #####
641 #####
642 #####
643 #####
644 #####
645 #####
646 #####
647 #####
648 #####
649 #####
650 #####
651 #####
652 #####
653 #####
654 #####
655 #####
656 #####
657 #####
658 #####
659 #####
660 #####
661 #####
662 #####
663 #####
664 #####
665 #####
666 #####
667 #####
668 #####
669 #####
670 #####
671 #####
672 #####
673 #####
674 #####
675 #####
676 #####
677 #####
678 #####
679 #####
680 #####
681 #####
682 #####
683 #####
684 #####
685 #####
686 #####
687 #####
688 #####
689 #####
690 #####
691 #####
692 #####
693 #####
694 #####
695 #####
696 #####
697 #####
698 #####
699 #####
700 #####
701 #####
702 #####
703 #####
704 #####
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
741 #####
742 #####
743 #####
744 #####
745 #####
746 #####
747 #####
748 #####
749 #####
750 #####
751 #####
752 #####
753 #####
754 #####
755 #####
756 #####
757 #####
758 #####
759 #####
760 #####
761 #####
762 #####
763 #####
764 #####
765 #####
766 #####
767 #####
768 #####
769 #####
770 #####
771 #####
772 #####
773 #####
774 #####
775 #####
776 #####
777 #####
778 #####
779 #####
780 #####
781 #####
782 #####
783 #####
784 #####
785 #####
786 #####
787 #####
788 #####
789 #####
790 #####
791 #####
792 #####
793 #####
794 #####
795 #####
796 #####
797 #####
798 #####
799 #####
800 #####
801 #####
802 #####
803 #####
804 #####
805 #####
806 #####
807 #####
808 #####
809 #####
810 #####
811 #####
812 #####
813 #####
814 #####
815 #####
816 #####
817 #####
818 #####
819 #####
820 #####
821 #####
822 #####
823 #####
824 #####
825 #####
826 #####
827 #####
828 #####
829 #####
830 #####
831 #####
832 #####
833 #####
834 #####
835 #####
836 #####
837 #####
838 #####
839 #####
840 #####
841 #####
842 #####
843 #####
844 #####
845 #####
846 #####
847 #####
848 #####
849 #####
850 #####
851 #####
852 #####
853 #####
854 #####
855 #####
856 #####
857 #####
858 #####
859 #####
860 #####
861 #####
862 #####
863 #####
864 #####
865 #####
866 #####
867 #####
868 #####
869 #####
870 #####
871 #####
872 #####
873 #####
874 #####
875 #####
876 #####
877 #####
878 #####
879 #####
880 #####
881 #####
882 #####
883 #####
884 #####
885 #####
886 #####
887 #####
888 #####
889 #####
890 #####
891 #####
892 #####
893 #####
894 #####
895 #####
896 #####
897 #####
898 #####
899 #####
900 #####
901 #####
902 #####
903 #####
904 #####
905 #####
906 #####
907 #####
908 #####
909 #####
910 #####
911 #####
912 #####
913 #####
914 #####
915 #####
916 #####
917 #####
918 #####
919 #####
920 #####
921 #####
922 #####
923 #####
924 #####
925 #####
926 #####
927 #####
928 #####
929 #####
930 #####
931 #####
932 #####
933 #####
934 #####
935 #####
936 #####
937 #####
938 #####
939 #####
940 #####
941 #####
942 #####
943 #####
944 #####
945 #####
946 #####
947 #####
948 #####
949 #####
950 #####
951 #####
952 #####
953 #####
954 #####
955 #####
956 #####
957 #####
958 #####
959 #####
960 #####
961 #####
962 #####
963 #####
964 #####
965 #####
966 #####
967 #####
968 #####
969 #####
970 #####
971 #####
972 #####
973 #####
974 #####
975 #####
976 #####
977 #####
978 #####
979 #####
980 #####
981 #####
982 #####
983 #####
984 #####
985 #####
986 #####
987 #####
988 #####
989 #####
990 #####
991 #####
992 #####
993 #####
994 #####
995 #####
996 #####
997 #####
998 #####
999 #####
1000 #####
```

```
130 ##### Point-Raster overlay
131 ##### Extracting the SWD vectors, the full thing or a subsample of it
132 ##### ScriptNameCode<- "M1b2"
133 #####
134 ##### ##### ##### ##### ##### ##### ##### ##### ##### #####
135 #####
136 ScriptNameCode<- "M1b2"
137 #####
138 #####
139 #####
140 #####
141 ##### Processing Environmental Layers -- ASCII as per maxent.jar #####
142 ##### LOAD ASCII Files and convert into Vector Data #####
143 #####
144 #####
145 #####
146 #####
147 InType <- 1 # This option enables users to consider other input type, formats other than ASCII or csv for example. Not
    considered further in this version yet but leaving it for backcompatibility reasons.
148 MIAPar$InType <- InType
149
150 #_AltCode if (InType==1) { # use this if additional input types are considered. see above
151
152 cat ("Processing ASCII Data")
153 ev.cnt<-as.vector Sys.glob(paste(dir.ev.cnt, "%_asc", sep="")) # only pick ascii # could add other (txt or others) option later
154 EVCntNum <-length(ev.cnt) # number of environmental variables at the onset
155 EVCntNames <- casefold(sub("","_asc","",basename(ev.cnt)),upper=T) #
156
157 ev.cat <- dir.ev.cat # dummy for now. Need the NAs
158 EVCatNames<- ev.cat # dummy for now. Need the NAs
159 if(is.na(dir.ev.cat))==FALSE{ # kinda convoluted I know but it works!
160 ev.cat<-as.vector(Sys.glob(paste(dir.ev.cat, #
    "%_asc", sep="")))
161
162 EVCatNum <-length(ev.cat) # number of categorical variables at the onset
163 EVCatNames <- casefold(sub("","_asc","",basename(ev.cat)),upper=T) # good to have these caps
164 } # ev.cat processing if
165 ev.asc <- na.omit(ev.cnt.ev.cat)) # all ascii files remove NA if no categorical data is present
166 EVAINames <- na.omit(c(EVCntNames,EVCatNames)) # remove NA if no categorical data is present
167 # example of individual rasters if you want:
168 # _AltCode RAS1 <- raster(ev.asc[[1])
169 # _AltCode head(RAS1) # _AltCode RAS1@data # _AltCode RAS1@data # "SpatialGridDataFrame"
170
171
172 ev.stack<- stack(ev.asc) # Jun 2013 ... stacking all EV layers
173 object.size(ev.stack)
174 for(as in 1:length(ev.asc))
175 {RASext <- raster(ev.asc[ras])
176 print(ras) # raster(ev.asc[ras])
177 print(RASext@extent)
178 }
179 rm(RASext)
180
181 names(ev.stack) <- EVAINames
182 list(ev.stack) # _InxCode
183 str(ev.stack)
184 class(ev.stack)
185 dim(ev.stack) #number of rows, columns of each raster files, x.n.layers (so technically don't need my own counter)#
186 DimStack <-dim(ev.stack)
187 ExtSize <- DimStack[1]*DimStack[2] # use this to then "clip off" or limit if it's too large
188
189 #####
190 #####
191 ##### Extract to Dataframe #####
192 ##### the full thing or a subsample of it #####
193 ##### only consider up to 500,000 records or you choose
194 #####
195 ##### Bkgnd_Dftrm (only background, no training data) #####
196 #####
197 #####
```

```
198 # Could also change this by hand MIAPar$SubSampleSize <- 500000
199
200 if(ExtSize>MIAPar$SubSampleSize) {SubSample2Extract<-MIAPar$SubSampleSize }else{SubSample2Extract <- ExtSize} #
    make sample size the same as the Full extent or subsample if larger a "reasonable subsampleSize of what you've decided in
    M1. half million records in this case
201
202 SubSample2Extract
203 MIAPar$SubSampleSize # or could update this in the MiaParamter ...
204
205
206 cat("Extracting all or subsample size of", SubSample2Extract, "background pits data from all raster files:")## let user know what's
    going on
207 ExtSize
208 set.seed(232)
209 Extr.EV <- as.data.frame(sampleRandom(ev.stack,
    SubSample2Extract,
210
211 xy = TRUE,
212 cell=TRUE,
213
214 na.rm = TRUE)) # only consider all EV data, could choose otherwise
215 SWD <- as.character(rep("Ug",nrow(Extr.EV))) # Uninformed "background", will rbind this to other data
216 Bkgnd_Dftrm <- unique(cbind(Extr.EV[,1:SWD.Extr.EV[-1]]) # Remove duplicates: if picked at random
217 str(Bkgnd_Dftrm) ##### Bkgnd_Dftrm (only background, no training data)
218 names(Bkgnd_Dftrm)
219 names(Bkgnd_Dftrm)[1:4] <- c("Bkgnd_ID","SWD","XCoord","YCoord") # making all the same to be able to bring together as
    DWS ## default also the xy names, as it might be handy to have it just as xy
220 Bkgnd_Dftrm[[2]] <- as.character(Bkgnd_Dftrm[[2]])
221 Bkgnd_Dftrm[[3]] <- as.numeric(Bkgnd_Dftrm[[3]])
222 Bkgnd_Dftrm[[4]] <- as.numeric(Bkgnd_Dftrm[[4]])
223
224 # _InxCode str(Bkgnd_Dftrm)
225 rm(SWD.Extr.EV)
226
227 ## Could make this also a SpatialPoints or not, useful later
228 # _OptCode Bkgnd.SP <- Bkgnd_Dftrm
229 # _OptCode coordinates(Bkgnd.SP) <- Bkgnd_Dftrm[c(3,4)] # this defines your csv as a spatial datapoint!
230 str(Bkgnd.SP)
231
232 save.image(file=paste(dir.rdata,
    if(DAC>0){DAC},
233 ScriptNameCode,
234 " Bkgnd", # object name
235 ".RData", sep="")) # Saving RData File
236
237 #####
238 #####
239 ##### M1b2 Interactive visualisation for QC #####
240 ##### Check visually that the data looks right
241 #####
242 #####
243 #####
244
245 plot(ev.stack[[1]])
246
247 if(PrintDraftFlag==TRUE){ #dev.off() #dev.copy(file) #grDevices::jpeg
    dev.print(vin.metafile,filename=paste(dir,plot.out,
248 "DraftPlot",
249 "ev.stack",
250 "ScriptNameCode,
251 "ev.off", sep="")) # lower resolution this is just a draft
252
253 } # end of PrintDraftFlag
254
255 ## decide which column to plot
256 names(Bkgnd_Dftrm)
257 col<-4
258
259 plot(Bkgnd_Dftrm$XCoord, # [[3]], # not quite an SP but get the picture
260 Bkgnd_Dftrm$YCoord, # [[col+1]],
261 cex=4,col="gray")
262 #####
```

```

263 # OptCode points(Bkgnd_Dftrm[[6]], cex=4,col="red")
264
265
266 ### Option to save these as snapshots ###
267 ## Option to save these as snapshots
268 ## Just change the flag to TRUE if you want to ... or use other simpler snapshots tools
269
270 PrintDraftFlag <- TRUE # TRUE #FALSE
271
272 if(PrintDraftFlag==TRUE){ #dev.off() #dev.copy(file) #grDevices::jpeg
273   dev.print(vwin,metafile.filename=paste(dfr,plot.out,
274     "DraftPlot",
275     ScriptNameCode,
276     "Bkgnd_Dftrm",
277     names(Bkgnd_Dftrm[col]),
278     #_OptCode #_obj", obj,PageNum, "of",NPages,
279     ".wmf", sep=""), res=80) # lower resolution this is just a draft
280
281 dev.off() # need this after "printing it out!
282 } # end of PrintDraftFlag
283
284 #_OptCode densityplot(Bkgnd_Dftrm[[6]]) # explore several options
285 #_OptCode plot(Bkgnd_Dftrm[[6]],
286 #_OptCode type="h", # or h but not very interesting
287 #_OptCode cex=.4,col="blue")
288
289
290 #_OptCode plot(Bkgnd_SPl[[6]])
291
292 names(Bkgnd_Dftrm)
293 plot(Bkgnd_Dftrm[,5]) # add 1 more the ID, not just the SWD and XY
294
295 #####
296 #####
297 ##### Process the Background Target Group if present #####
298 #####
299 #####
300 #####
301 ##### Preparing the BTG Data #####
302 ##### Either 1 file with 1 or more BTG's
303 ##### or Multiple Files with 1 BTG in each
304 #####
305 ## M1b Part 2 BTG part 1
306 MIAPar$BTGFile
307
308 ## If BTG exists
309 if(length(na.omit(MIAPar$BTGFile))>0){cat("Processing Background Target Group")
310
311 #####
312 ##### Read BTG Data files directly #####
313 ##### Either 1 file with 1 or more RV's
314 ##### or Multiple Files with 1 RV in each
315
316 #####
317 #####
318 MIAPar$BTGFile
319 MIAPar$BTGFile <- sort(MIAPar$BTGFile) # ensures "alphanumerical order" also makes consistent
320 # InpsCode MIAPar$BTGFile
321 # InpsCode MIAPar$BTGFile
322 BTGFile.Data <- data.frame0 # this is only temporary
323 BTG.Data <- list0
324 BTG_RVNames <- vector0 # like RVNames but BTG, related but different
325 length(MIAPar$BTGFile)
326
327 if (length(MIAPar$BTGFile)==1) {
328   fl<-1 ## keep it so can keep code the same
329   BTGFile.Data <- na.omit(as.data.frame(read.csv(MIAPar$BTGFile[fl]))) # just opening the file and
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

326 loading the data .. could have 1 or more testdata columns
327 BTGFile.Data[[1] <- as.character(BTGFile.Data[[1]]) # make sure it is read as a character not factor
328 BTG_RVNames <- sort(unique(BTGFile.Data[[1]])) # Unique Occurrences if the file holds more
329   than 1 species - Sort to better keep track of RV
330 # InpsCode BTG_RVNames
331 for (rv in seq(length(BTG_RVNames)))
332 {
333   #_OptCode BTG_RVNames[rv] <- paste("RV",fl,"_",names(BTG.Data)[[fl]],sep="") # Could add
334   the RV in front if you want to group/order by RV.
335   pos <- which(BTGFile.Data[[1]]== BTG_RVNames[rv])
336   #_OptCode optional names(BTG.Data[[pos]] <-
337     paste("RV",fl,"_",names(BTG.Data[[pos]]),sep="") # add this to keep sorting and order ok
338   #_OptCode names(BTG.Data[[rv]][1] <- paste("RV",rv,"_",BTG_RVNames[rv],sep="")
339   names(BTG.Data)[rv] <- paste("BTG_RV",rv,"_",BTG_RVNames[rv],sep="") # not sure if we should
340   stick also the name or keep it generic
341   #_InsCode str(BTG.Data[[rv]])
342 }
343 rm(BTGFile.Data)
344 } # end of BTG File = 1 if statement
345
346 if (length(MIAPar$BTGFile)>1) { # more than 1 separate training file
347   for (fl in seq(length(MIAPar$BTGFile))) # the number of TestFiles ...
348   {
349     cat("\nProcessing Background Target Group Data File",fl)
350     BTG.Data[[fl]] <- na.omit(as.data.frame(read.csv(MIAPar$BTGFile[fl]))) # this time place
351     contents directly
352     #_InsCode str(BTG.Data[[fl]])
353     BTG.Data[[fl]][1] <- as.character(BTG.Data[[fl]][1]) # make sure it is read as a character not
354     factor
355     #_InsCode str(BTG.Data[[fl]])
356     BTG_RVNames <- c(BTG_RVNames,unique(BTG.Data[[fl]][1])) # 1 is column in this case adding
357     all RV names ... so then they can be "sorted", BTW, not using BTGFile.Data dfr here at all
358     actually, as opposed to above oh wle
359     names(BTG.Data)[fl] <- paste("BTG_RV",fl,"_",BTG_RVNames[fl],sep="") # not sure if we
360     should stick also the name or keep it generic
361     } # end of processing all Training Files
362     rm(BTGFile.Data)
363     } # end of BTG File greater than 1 if statement ends
364   }
365   } # end of loading the BTG Data, part 1
366
367 #####
368 ###
369 #####
370 ##### If BTGFlag==TRUE (or BTGFile actually greater 0) ###
371 ##### Final of first part: Reviewing BTG.Data ###
372 #####
373 #####
374 #####
375 #####
376 #####
377 #####
378 #####
379 #####
380 #####
381 #####
382 #####
383 #####
384 #####
385 #####
386 #####
387 #####
388 #####
389 #####
390 #####
391 #####
392 #####
393 #####
394 #####
395 #####
396 #####
397 #####
398 #####
399 #####
400 #####
401 #####
402 #####
403 #####
404 #####
405 #####
406 #####
407 #####
408 #####
409 #####
410 #####
411 #####
412 #####
413 #####
414 #####
415 #####
416 #####
417 #####
418 #####
419 #####
420 #####
421 #####
422 #####
423 #####
424 #####
425 #####
426 #####
427 #####
428 #####
429 #####
430 #####
431 #####
432 #####
433 #####
434 #####
435 #####
436 #####
437 #####
438 #####
439 #####
440 #####
441 #####
442 #####
443 #####
444 #####
445 #####
446 #####
447 #####
448 #####
449 #####
450 #####
451 #####
452 #####
453 #####
454 #####
455 #####
456 #####
457 #####
458 #####
459 #####
460 #####
461 #####
462 #####
463 #####
464 #####
465 #####
466 #####
467 #####
468 #####
469 #####
470 #####
471 #####
472 #####
473 #####
474 #####
475 #####
476 #####
477 #####
478 #####
479 #####
480 #####
481 #####
482 #####
483 #####
484 #####
485 #####
486 #####
487 #####
488 #####
489 #####
490 #####
491 #####
492 #####
493 #####
494 #####
495 #####
496 #####
497 #####
498 #####
499 #####
500 #####
501 #####
502 #####
503 #####
504 #####
505 #####
506 #####
507 #####
508 #####
509 #####
510 #####
511 #####
512 #####
513 #####
514 #####
515 #####
516 #####
517 #####
518 #####
519 #####
520 #####
521 #####
522 #####
523 #####
524 #####
525 #####
526 #####
527 #####
528 #####
529 #####
530 #####
531 #####
532 #####
533 #####
534 #####
535 #####
536 #####
537 #####
538 #####
539 #####
540 #####
541 #####
542 #####
543 #####
544 #####
545 #####
546 #####
547 #####
548 #####
549 #####
550 #####
551 #####
552 #####
553 #####
554 #####
555 #####
556 #####
557 #####
558 #####
559 #####
560 #####
561 #####
562 #####
563 #####
564 #####
565 #####
566 #####
567 #####
568 #####
569 #####
570 #####
571 #####
572 #####
573 #####
574 #####
575 #####
576 #####
577 #####
578 #####
579 #####
580 #####
581 #####
582 #####
583 #####
584 #####
585 #####
586 #####
587 #####
588 #####
589 #####
590 #####
591 #####
592 #####
593 #####
594 #####
595 #####
596 #####
597 #####
598 #####
599 #####
600 #####
601 #####
602 #####
603 #####
604 #####
605 #####
606 #####
607 #####
608 #####
609 #####
610 #####
611 #####
612 #####
613 #####
614 #####
615 #####
616 #####
617 #####
618 #####
619 #####
620 #####
621 #####
622 #####
623 #####
624 #####
625 #####
626 #####
627 #####
628 #####
629 #####
630 #####
631 #####
632 #####
633 #####
634 #####
635 #####
636 #####
637 #####
638 #####
639 #####
640 #####
641 #####
642 #####
643 #####
644 #####
645 #####
646 #####
647 #####
648 #####
649 #####
650 #####
651 #####
652 #####
653 #####
654 #####
655 #####
656 #####
657 #####
658 #####
659 #####
660 #####
661 #####
662 #####
663 #####
664 #####
665 #####
666 #####
667 #####
668 #####
669 #####
670 #####
671 #####
672 #####
673 #####
674 #####
675 #####
676 #####
677 #####
678 #####
679 #####
680 #####
681 #####
682 #####
683 #####
684 #####
685 #####
686 #####
687 #####
688 #####
689 #####
690 #####
691 #####
692 #####
693 #####
694 #####
695 #####
696 #####
697 #####
698 #####
699 #####
700 #####
701 #####
702 #####
703 #####
704 #####
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
741 #####
742 #####
743 #####
744 #####
745 #####
746 #####
747 #####
748 #####
749 #####
750 #####
751 #####
752 #####
753 #####
754 #####
755 #####
756 #####
757 #####
758 #####
759 #####
760 #####
761 #####
762 #####
763 #####
764 #####
765 #####
766 #####
767 #####
768 #####
769 #####
770 #####
771 #####
772 #####
773 #####
774 #####
775 #####
776 #####
777 #####
778 #####
779 #####
780 #####
781 #####
782 #####
783 #####
784 #####
785 #####
786 #####
787 #####
788 #####
789 #####
790 #####
791 #####
792 #####
793 #####
794 #####
795 #####
796 #####
797 #####
798 #####
799 #####
800 #####
801 #####
802 #####
803 #####
804 #####
805 #####
806 #####
807 #####
808 #####
809 #####
810 #####
811 #####
812 #####
813 #####
814 #####
815 #####
816 #####
817 #####
818 #####
819 #####
820 #####
821 #####
822 #####
823 #####
824 #####
825 #####
826 #####
827 #####
828 #####
829 #####
830 #####
831 #####
832 #####
833 #####
834 #####
835 #####
836 #####
837 #####
838 #####
839 #####
840 #####
841 #####
842 #####
843 #####
844 #####
845 #####
846 #####
847 #####
848 #####
849 #####
850 #####
851 #####
852 #####
853 #####
854 #####
855 #####
856 #####
857 #####
858 #####
859 #####
860 #####
861 #####
862 #####
863 #####
864 #####
865 #####
866 #####
867 #####
868 #####
869 #####
870 #####
871 #####
872 #####
873 #####
874 #####
875 #####
876 #####
877 #####
878 #####
879 #####
880 #####
881 #####
882 #####
883 #####
884 #####
885 #####
886 #####
887 #####
888 #####
889 #####
890 #####
891 #####
892 #####
893 #####
894 #####
895 #####
896 #####
897 #####
898 #####
899 #####
900 #####
901 #####
902 #####
903 #####
904 #####
905 #####
906 #####
907 #####
908 #####
909 #####
910 #####
911 #####
912 #####
913 #####
914 #####
915 #####
916 #####
917 #####
918 #####
919 #####
920 #####
921 #####
922 #####
923 #####
924 #####
925 #####
926 #####
927 #####
928 #####
929 #####
930 #####
931 #####
932 #####
933 #####
934 #####
935 #####
936 #####
937 #####
938 #####
939 #####
940 #####
941 #####
942 #####
943 #####
944 #####
945 #####
946 #####
947 #####
948 #####
949 #####
950 #####
951 #####
952 #####
953 #####
954 #####
955 #####
956 #####
957 #####
958 #####
959 #####
960 #####
961 #####
962 #####
963 #####
964 #####
965 #####
966 #####
967 #####
968 #####
969 #####
970 #####
971 #####
972 #####
973 #####
974 #####
975 #####
976 #####
977 #####
978 #####
979 #####
980 #####
981 #####
982 #####
983 #####
984 #####
985 #####
986 #####
987 #####
988 #####
989 #####
990 #####
991 #####
992 #####
993 #####
994 #####
995 #####
996 #####
997 #####
998 #####
999 #####
1000 #####

```



```
keep track of RV
RVNames
for (rv in seq(length(RVNames)))
{
  #_OptCode RVNames[rv] <- paste("RV",fil,"_",names(Train.Data)[fil],sep="") # Add the RV in front ...or maybe not?
  pos <- which(TrainFile.Data[[1]]== RVNames[rv])
  Train.Data[[rv]] <- TrainFile.Data[pos]
  #_OptCode names(Train.Data[[fil]])[1] <- paste("RV",fil,"_",names(Train.Data[[fil]]),sep="") # this option is useful for
  sorting order, but again, user dependent
  names(Train.Data[[rv]][1:3]) <- c("SWD","XCoord","YCoord") # making all the same to be able to bring together as SWD ##
  ## with SWD for first column which later will be used to identify if it's training or backgrnd in the SWD file
  #_OptCode names(Train.Data)[rv] <- paste("RV",rv,"_",RVNames[rv],sep="") # Keep Train Data Just that RV
  names(Train.Data)[rv] <- paste(if(DAC>0){DAC}, # Handy
    # "Train_",
    "RV",
    if(rv<10){},rv,
    sep="")
  #_InCode str(Train.Data[[rv]])
}
# end of Train File = 1 if statement ends
519 }
520 if (length(MIAPar$TrainFile)>1) { cat("Processing more than 1 training file") # more than 1 separate training file
521
522
523
524
525
526
527
528
529
530 names(Train.Data[[fil]][1:3]) <- c("SWD","XCoord","YCoord") # make sure it is read as a character not factor
531
532
533
534
535
536
537
538
539
540 } # end of processing all Training Files
541
542 # rm(SortRV)
543 } # Train File greater than 1 if statement ends
544
545 SortRV <- sort(names(Train.Data))
546 RVNames <- names(Train.Data)
547 Train.Data <- Train.Data[SortRV] ## it is koshier to use the same object name?
548
549 str(Train.Data)
550 dim(Train.Data[[fil]])
551 names(Train.Data[[rv]])
552 head(Train.Data[[rv]])
553 rm(TrainFile.Data)
554 #####
555 ### Load Test Data ###
556 ### This is for Presence Absence
557 ### or at least 1 file per Test RV
558 #####
559 TestFile <- MIAPar$TestFile # if not reloading from ParOnly
560 length(TestFile)
561 TestFile <- sort(TestFile) # ensures "alphanumerical order" also makes consistent with training RV
562 TestFile
563 Test.Data <- list()
564 rv<-1
565 fil<-1
```

```
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```

632 ### for point overlay with EVLayers      #####
633 ### Training and Test if present
634 ###
635 #####
636
637 str(Train.Data)
638 names(Train.Data)
639 Train.SWD <- list0 # Training Sample With Data, as a Spatial Data Point too
640 rv <- 1
641
642 {
643   Train.SWD[[rv]] <- as.data.frame(Train.Data[[rv]][,1]) # This will just get the Train Name, and the XY in the coordinate will
644   be the original, but in the data it will be 2 columns that are the same as the grain ev matrix
645   Extr.EV <- as.data.frame(extract(ev.stack, ## user can choose the number of layers to process ... default is all
646     Train.Data[[rv]][2:3], # Train.Data[[rv]][c(2,3)],
647     cellNumbers=TRUE, # yes, this cellnumbers here IS the same as cells in sampleRandom extract
648     #_OptCode sp= TRUE,
649     #_OptCode xy= TRUE,
650     na.rm=TRUE) # user choose ... Remove all NA's from any Layers or not
651   dim(Extr.EV[[1])
652   Extr.XYCoord <- xyFromCell(ev.stack.cell=Extr.EV[[1]])# these are the grain/grid's XY's of the ID, don't take unique till you've
653   chbided them to original then "squeeze doubles out"
654   # Insp dim(Extr.XYCoord)
655   ## Unique reminds users that it should only count presences at the right grain ... these are duplicates for that grain
656   ## only thing to decide is whether to keep carrying the original XY or the one extracted from the grid later
657   ## Well decision is taken ... keep only the "grid grain xy"s
658   Train.SWD[[rv]] <- unique(cbind(Extr.EV[[1],
659     Train.SWD[[rv]], # Don't use the @data here because we've not defined as SP yet, because we're
660     shrinking the size potentially
661     Extr.XYCoord,
662     Extr.EV[-1]))
663   #_InsCode str(Train.SWD)
664   ## Alternatively can change back to the 1:3 order, and put ID as the fourth element, rather than first Extr.EV[-1])) # assign
665   the info just as well - may not need them but may indeed, can remove later
666
667   Train.SWD[[rv]] <- na.omit(Train.SWD[[rv]])
668   names(Train.SWD[[rv]][,1]) <- c("Bgknd_ID", "SWD", "XCoord", "YCoord") # making all the same to be able to bring together
669   as DWS ## default also the xy names, as it might be handy to have it just as xy
670   coordinates(Train.SWD[[rv]] <- cbind(Train.SWD[[rv]]$XCoord, Train.SWD[[rv]]$YCoord) # now assign these coordinates to
671   the Spatial Dataframe
672   #_InsCode str(Train.SWD[[rv]]) # dim(Train.SWD[[rv]])
673   Train.SWD[[rv]]@data[[1]] <- as.character(Train.SWD[[rv]]@data[[2]])
674   Train.SWD[[rv]]@data[[1]] <- as.numeric(Train.SWD[[rv]]@data[[3]])
675   Train.SWD[[rv]]@data[[1]] <- as.numeric(Train.SWD[[rv]]@data[[4]])
676   names(Train.SWD[[rv]])
677   names(Train.SWD[[rv]] <- paste((IDAC>0)IDAC3, "Train_SWD_",
678     "RV",
679     if(rv<10){rv, # "_", # RVNames[rv],
680     sep=""}) # simpler to also rename the SDP
681   ##### Good practice to remember to tidy after yourself #####
682   rm(Extr.EV, Extr.XYCoord)
683   } # end of rv in Train.SWD
684
685 rv<-1
686 head(Train.SWD[[rv]])
687 str(Train.SWD[[rv]])
688 names(Train.SWD[[rv]])
689 # Train.SDP[[1]]$ExtID
690 dim(Train.SWD[[rv]])
691 class(Train.SWD[[1]])
692 # class(Train.SWD[[1]]@data[[2]])
693 # (Train.SWD[[rv]]@data[[2]])
694 #####
695 repeat if Test Data present ###
696   Sep 17 2013 --- Updated too ...
697 #####
698 #####
699 #####
700 #####
701 #####
702 #####
703 #####
704 #####
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
741 #####
742 #####
743 #####
744 #####
745 #####
746 #####
747 #####
748 #####
749 #####
750 #####
751 #####
752 #####
753 #####
754 #####
755 #####
756 #####
757 #####
758 #####

```

```

696
697 if(length(TestFile)!=0) # repeat same if test data present
698 {
699   #str(Test.Data)
700   names(Test.Data)
701   Test.SWD <- list0 # Testing Sample With Data, as a Spatial Data Point too
702   rv <- 1
703   for (rv in seq(length(Test.Data)))
704   {
705     Test.SWD[[rv]] <- as.data.frame(Test.Data[[rv]][,1]) # This will just get the Test Name, and the XY in the coordinate will be
706     the original, but in the data it will be 2 columns that are the same as the grain ev matrix
707     Extr.EV <- as.data.frame(extract(ev.stack, ## user can choose the number of layers to process ... default is all
708       Test.Data[[rv]][2:3], # Test.Data[[rv]][c(2,3)],
709       cellNumbers=TRUE, # yes, this cellnumbers here IS the same as cells in sampleRandom extract
710       #sp= TRUE,
711       #xy= TRUE,
712       na.rm=TRUE) # user choose ... Remove all NA's from any Layers or not
713     #head(Extr.EV) #dim(Extr.EV) #dim(Extr.EV[[1]])
714     Extr.XYCoord <- xyFromCell(ev.stack.cell=Extr.EV[[1]])# these are the grain/grid's XY's of the ID, don't take unique till
715     you've chbided them to original then "squeeze doubles out"
716     # dim(Extr.XYCoord)
717     ## Unique reminds users that it should only count presences at the right grain ... these are duplicates for that grain
718     ## only thing to decide is whether to keep carrying the original XY or the one extracted from the grid later
719     ## Well decision is taken ... keep only the "grid grain xy"s
720     Test.SWD[[rv]] <- unique(cbind(Extr.EV[[1],
721       Test.SWD[[rv]], # Don't use the @data here because we've not defined as SP yet, because we're
722       shrinking the size potentially
723       Extr.XYCoord, # Sep 17 2013 Still to make up your mind if you want to keep here or not ... keep for
724       now, it is in the SP coord
725       Extr.EV[-1]))
726     # change back to the 1:3 order, and put ID as the fourth element, rather than first Extr.EV[-1])) # assign the info just as
727     well - may not need them but may indeed, can remove later
728
729     Test.SWD[[rv]] <- na.omit(Test.SWD[[rv]])
730
731     names(Train.SWD[[rv]][,1]) <- c("Bgknd_ID", "SWD", "XCoord", "YCoord") # making all the same to be able to bring together
732     as DWS ## default also the xy names, as it might be handy to have it just as xy
733     coordinates(Train.SWD[[rv]] <- cbind(Train.SWD[[rv]]$XCoord, Train.SWD[[rv]]$YCoord) # now assign these coordinates to
734     the Spatial Dataframe
735     #_InsCode str(Train.SWD[[rv]])
736     Train.SWD[[rv]]@data[[1]] <- as.character(Train.SWD[[rv]]@data[[2]])
737     Train.SWD[[rv]]@data[[1]] <- as.numeric(Train.SWD[[rv]]@data[[3]])
738     Train.SWD[[rv]]@data[[1]] <- as.numeric(Train.SWD[[rv]]@data[[4]])
739     names(Train.SWD[[rv]] <- paste(IDAC, "Test_SWD_",
740       "RV",
741       if(rv<10){rv, # "_", # RVNames[rv],
742       sep=""}) # simpler to also rename the SDP
743     ##### Good practice to remember to tidy after yourself #####
744     rm(Extr.EV, Extr.XYCoord)
745     } # end of rv in Test.SWD
746
747 } # end of Test data present condition
748
749 if(length(TestFile)!=0){
750   head(Test.SWD)
751   names(Test.SWD)
752   names(Train.SWD[[1]])
753   head(Test.SWD[[1]])
754   } # check visually if test is present
755
756 unique(Test.SWD[[1]]$SWD)
757 str(Test.SWD[[rv]])# this is the new data after doing the overlay with EV's and dropping any NO DATA
758 str(Test.Data) # this is the original data before doing the overlay with EV's which may have no data
759 ##### Save here before visualisation so can skip visualisation if not needed ###
760 save.image(file=paste(dir.rdata,
761   if(IDAC>0)IDAC3,
762   ScriptNameCode,
763   "_KeyObj", # object name
764   ".R", # file extension
765   sep=""))
766

```

```
759 ".RData",sep="") # Saving RData File
760 #####
761 #####
762 ##### M1b2 Interactive visualisation bit
763 #####
764 #####
765 #####
766 load("RData/A01_M1b2_KeyObj.RData")
767 #####
768 #####
769 ## Part 2 - Train and test data
770 #####
771 #####
772 ##### Check visually that the data looks right #####
773 #####
774 #####
775 #####
776 class(Train.SWD[[1]])
777 rv<-1
778 plot(ev.stack[[1]])
779 points(Train.SWD[[rv]],col="red",
780 cex=.6)
781 points(Test.SWD[[1]],col="purple",cex=.4)
782 #####
783 ## Option to save these as snapshots ##
784 ## user should decide this and place name accordingly, here's a "skeleton"
785 ## Just change the flag to TRUE if you want to ... or use other simpler snapshots tools
786
787 PrintDraftFlag <- TRUE #FALSE
788 if(PrintDraftFlag==TRUE){ #dev.off() #dev.copy(file) #grDevices:jpeg
789 dev.print(vin.metafile,filename=paste(dir,plot.out,
790 "DraftPlot",
791 "Train_SWD_RV",rv,
792 names(Train.SWD[[rv])),
793 #obj", obj.PageNum, "of", NPages,
794 "wmf", sep=""), res=80) # lower resolution this is just a draft
795 dev.off() # need this after "printing it out!
796 } # end of PrintDraftFlag
797
798 #####
799 #####
800 ##### end of Interactive visualisation bit
801 #####
802 #####
803 #####
804 #####
805 #####
806 #####
807 ## Additional "data clean up" to clarify exact number of parameters ###
808 ##
809 ## Too often users are not aware there is a discrepancy between the "Starting n (number) of Training points"
810 ## their response variable .csv file has, and the number that is actually used in the MaxEnt modelling.
811 ## A reduction of this number often occurs, for several reasons, including:
812 ## 1. Simple duplication of records in the original .csv file,
813 ## 2. Two or more points falling on the same EV grid as a result spatial overlay with background grid
814 ## 3. Points falling outside the extent of EV grid as a result of spatial overlay
815
816 ## Therefore good to clean this up and print these out
817 ## Write object out to .cv
818 ## 2 versions for both train and test data -- one with and one without the EV Data columns
819 ## 1. Train.SWD -- meaning csv with EV data extracted
820 ## 2. Train_RV1 -- csv of Training points alone (without EV data)
821 ## Option to also include the first column (the ID)
822 ## Defaulting without as SWD, so it can be used by users outside R directly into maxent.jar
823 #####
824 #####
```

```
825 obj<-1
826 for (obj in seq(length(Train.SWD)))
827 {
828   write.csv(Train.SWD[[obj]]@data[-1], #_OptCode [-c(2.3)], ## Skip the xy or not
829   paste(dir.swd,
830   "R_", # just a small prefix to remind you where its come from (can choose to remove)
831   names(Train.SWD[obj]),
832   ".csv",
833   sep=""),row.names=FALSE) #
834 }
835
836
837 obj<-1
838 for (obj in seq(length(Train.SWD)))
839 {
840   write.csv(Train.SWD[[obj]]@data[2:-1], #_OptCode [-c(2.3)], ## Skip the xy or not
841   # OptCode "R_", # just a small prefix to remind you where its come from (can choose to remove)
842   paste(dir.swd,
843   # OptCode DAC,
844   "Train_",
845   "RV",
846   if(obj<10){obj, #_OptCode "_", # OptCode RVNames[rv],
847   # OptCode names(Train.SWD[obj]),
848   ".csv",
849   sep=""),row.names=FALSE) #
850 }
851
852 ### print the fixed Test data too the one after the overlay, with no na
853 obj<-1
854 for (obj in seq(length(Test.SWD)))
855 {
856   write.csv(Test.SWD[[obj]]@data[-1], # OptCode [-c(2.3)], ## Skip the xy or not
857   paste(dir.swd, #
858   "R_", # just a small prefix to remind you where its come from (can choose to remove)
859   names(Test.SWD[obj]),
860   ".csv",
861   sep=""),row.names=FALSE) #
862 }
863
864 obj<-1
865 for (obj in seq(length(Test.SWD)))
866 {
867   write.csv(Test.SWD[[obj]]@data[2:-1], #_OptCode [-c(2.3)], ## Skip the xy or not
868   paste(dir.swd,
869   # OptCode DAC,
870   "Test_",
871   "RV",
872   if(obj<10){obj, # "_", # RVNames[rv],
873   # OptCode names(Train.SWD[obj]),
874   ".csv",
875   sep=""),row.names=FALSE) #
876 }
877
878 save.image(file=paste(dir.rdata,
879 if(DAC>0){DAC},
880 ScriptNameCode, # which script? which step, part in the process etc
881 #_OptCode "_part2",
882 "ZAllObj", # object name
883 "RData",sep="")) #
884
885 #####
886 #####
887 #####
888 ##### M1b_LoadData Part 3 ###
889 ##### Creating Sample With Data which includes background and training pts: N (big N)
890 ##### Plating Training Data on top first, then "joining/adding" all the Background, but not if
891 ##### it already exists in the Training set of course
892 #####
893 ##### 1. Add training data to Bkgnd_Dftrm, make it a list, to fit one or more RV
```



```

894 #####
895 ##### 2. also option of BTG (Background Target Group)
896 #####
897 ### M1_N_SWD_RV -- Training + Subsample (or the full thing) Bkgnd Data, (N= n + (Bkgnd-n))
898 ##### In Module 2 SWD is either the entire Bkgnd or the max 500,000 subsample set
899 ##### In Module 3 SWD it will be the 10,000 + or whatever else you choose
900 ##### or the Bkgnd Target Group if you have one
901 #####
902 #####
903 #####
904 ScriptNameCode <- "M1b3" # script name code used to identify where you are in the process
905
906 length(Train.SWD)
907 Bkgnd_SWD <- list() # make this one back into simple dataframe, not sp
908 for (rv in seq(length(Train.SWD))) # need to make an SWD for each RV (option to use same bkgnd or diff of course, but BTG
  will all be different)
909 {
910   Bkgnd_SWD[[rv]] <- Bkgnd_Dfrfm # 2 in 1 ... if BTG exists ... this gets overwritten. Pros and cons of how many objects get
    created
911   names(Bkgnd_SWD)[rv] <- paste("Bkgnd_SWD RV",rv, sep="")
912   if (length(na.omit(MIAPar$BTGFile))>0){Bkgnd_SWD[[rv]] <- BTG.SWD[[rv]]@data
    Bkgnd_SWD[[rv]][?] <- "UB" # now it's ok to turn this into "UB"
913   names(Bkgnd_SWD)[rv] <- names(BTG.SWD)[rv]} # use Background Target instead
914
915 } # use Background Target instead
916
917 } # END OF RV
918
919 #_OptCode is: na(MIAPar$BTGFile[1]) == FALSE
920 #_OptCode if (length(na.omit(MIAPar$BTGFile))>0){Bkgnd_SWD <- BTG.SWD} # use Background Target instead
921
922 ## check that the names and dim are the same so we can rbind
923 str(Bkgnd_SWD)
924 str(Bkgnd_SWD[[rv]])
925 names(Bkgnd_SWD)
926 names(Bkgnd_SWD[[rv]])
927 class(Bkgnd_SWD[[rv]])
928 class(Bkgnd_SWD)
929 str(Train.SWD[[rv]])
930 ## Drop the Bkgnd_Dfrfm as it is now superflous. This can be changed by user.
931 rm(Bkgnd_Dfrfm)
932
933
934 #####
935 #####
936 ### Bring back the join
937 ## Join the Training to the Bkgnd, and then reclassify those overlapping
938 ## as Training vs UB
939 ##
940 #####
941 #####
942 ##
943 str(Bkgnd_SWD[[rv]])
944 str(Train.SWD[[rv]]@data)
945 str(Bkgnd_SWD)
946
947 require(plyr; MIAPar$dir.Rlib) # used by all the Joining portions of the scripts
948 rv<-1
949 M1_N_SWD_RV <-list() #
950 for (rv in seq(length(Train.SWD)))
951 {
952   xJoin <- Train.SWD[[rv]]@data # put RV first (as in Halvorsen literature) #Clarify statement, e.g. with reference to one of
    the papers or a place in it
953   yJoin <- Bkgnd_SWD[[rv]] # not a sp file for now @data
954
955   DfrfmJoin <- join(xJoin,yJoin,
    by=names(xJoin[1]),
    type="full",
    match="first" # as there are now no duplication in data, this is best, as it is a faster join option
956   )
957   M1_N_SWD_RV[[rv]] <- DfrfmJoin
958
959

```

```

960 names(M1_N_SWD_RV)[rv] <- paste("M1_N_SWD_",
    "RV",
961   if(rv<10){},rv,
    #_OptCode "_",RVNames[rv], # choice if you want to use the name or just abbreviation or both
962   sep="") #
963 ## decide if test data goes at the end or not
964
965 #_InsCode str(xJoin)
966 #_InsCode str(yJoin)
967 }
968
969 rm(xJoin,yJoin,DfrfmJoin)
970
971 str(M1_N_SWD_RV)
972 names(M1_N_SWD_RV)
973 names(M1_N_SWD_RV)
974 names(M1_N_SWD_RV[[rv]])
975 EVCatNames
976
977 head(M1_N_SWD_RV[[rv]])
978 tail(M1_N_SWD_RV[[rv]])
979 str(M1_N_SWD_RV[[rv]])
980 nrow(M1_N_SWD_RV[[rv]])
981 nrow(M1_N_SWD_RV[[rv]])
982 nrow(M1_N_SWD_RV[[rv]])
983 #####
984 #####
985 #####
986 #####
987 ### Add here Test data #####
988 #####
989 #####
990 str(M1_N_SWD_RV)
991 #_InsCode str(M1_N_SWD_RVNoTest)
992 M1_N_SWD_RVNoTest <- M1_N_SWD_RV # keep for now to deal with n and NswD
993 M1_N_SWD_RV <-M1_N_SWD_RVNoTest
994 rv<-1
995
996 head(Train.SWD)
997 Test_MIN_RV<-list() # Unique Test Data that has same order and location as M1_N_SWD_RV but can be used later for testing
998
999 for (rv in seq(length(Train.SWD)))
1000 {
1001   rvt <- rv # if more than 1 test data files, then rvt same as rv
1002   if (length(Train.SWD)==1){rvt <- } # else stick to 1
1003   xJoin <- M1_N_SWD_RV[[rv]]
1004   yJoin <- Test.SWD[[rv]]@data # NA OMIT!
1005   #str(yJoin)
1006
1007   names(yJoin)[?] <- "SWD_Test" # change so you don't loose in join
1008   # names(yJoin)
1009   DfrfmJoin <- join(xJoin,yJoin,
    by=names(xJoin[1]),
    type="full",
    match="all" # this does not notice dups or not ... but there are no dups here anyway (I did a unique before)
1010   )
1011   # str(DfrfmJoin)
1012   # unique(DfrfmJoin$SWD)
1013   # unique(DfrfmJoin$SWD_Test)
1014   # unique(cbind(DfrfmJoin$SWD,DfrfmJoin$SWD_Test))
1015   #count(DfrfmJoin$SWD)
1016   #str(DfrfmJoin$SWD[[1])
1017   # unique(DfrfmJoin$SWD[[1])
1018   #count(DfrfmJoin$SWD[[1])
1019   # unique(DfrfmJoin$SWD_Test)
1020   #count(DfrfmJoin$SWD_Test)
1021   #length(UBTest)
1022   #length(UBTest)
1023   DfrfmJoin$SWD_Test[UBTest] <- "UB"
1024   #unique(DfrfmJoin$SWD_Test)
1025
1026   UBPos <- which(is.na(DfrfmJoin$SWD==TRUE)) # New UB added from Test Data
1027   DfrfmJoin$SWD[UBPos] <- "UB" ## # New UB added from Test Data
1028   UBTest <-which(is.na(DfrfmJoin$SWD_Test==TRUE))
1029   #length(UBTest)
1030   DfrfmJoin$SWD_Test[UBTest] <- "UB"
1031   #unique(DfrfmJoin$SWD_Test)
1032

```

```
1030 # NAPos <- which(is.na(DtfmJoin[5])==TRUE)) # There are still NA's in the overlay with test data
1031
1032 M1_N_SWD_RV[r~v]<- na.omit(DtfmJoin) ### Adding the Test Data
1033 names(M1_N_SWD_RV[r~v]) <- paste("M1_N_SWD_",
1034 "RV",
1035 if(nv<10){0},nv,
1036 #_OptCode " ",RVNames[r~v], # choice if you want to use the name or just abbreviation or both
1037 sep="") #
1038
1039 str(M1_N_SWD_RV[r~v])$SWD_Test)
1040 # InsCode length(UBPos)
1041 rm(UBPos,UBTest)
1042 str(M1_N_SWD_RV[r~v])$SWD_Test)
1043 unique(M1_N_SWD_RV[r~v])$SWD_Test)
1044 names(M1_N_SWD_RV[r~v])
1045 names(M2_N_SWD_RV[r~v])
1046
1047 TestPos <- which(M1_N_SWD_RV[r~v])$SWD_Test=="UB") # New UB added from Test Data
1048 Test_MIN_RV[r~v]<- M1_N_SWD_RV[r~v][TestPos,] #length(M1_N_SWD_RV[r~v]),3,4)) # Unique Test Data that has
same order and location as M1_N_SWD_RV but can be used later for testing
1049 rm(TestPos)
1050 names(Test_MIN_RV[r~v]) <- paste("Test_MIN_RV", if (nv<10){0},nv, sep="")
1051
1052 ## decide if test data goes at the end or not
1053
1054 #_InsCode unique(M1_N_SWD_RV[r~v])$SWD_Test)
1055 #_InsCode str(xJoin)
1056 #_InsCode str(yJoin)
1057 } # adding test data to M1
1058 #rv<-2
1059 rm(xJoin,yJoin,DtfmJoin)
1060
1061 #_InsCode str(M1_N_SWD_RV)
1062 names(M1_N_SWD_RV)
1063 names(M1_N_SWD_RV[r~v])
1064 head(M1_N_SWD_RV[r~v])
1065 str(M1_N_SWD_RV[r~v])
1066 unique(M1_N_SWD_RV[r~v])$SWD)
1067 count(M1_N_SWD_RV[r~v])$SWD)
1068 count(M1_N_SWD_RV[r~v])$SWD_Test)
1069 nrow(M1_N_SWD_RV[r~v])
1070 nrow(M1_N_SWD_RV[r~v])
1071 str(Test_MIN_RV)
1072 str(M1_N_SWD_RV[r~v])
1073
1074 #####
1075 #####
1076 ## Part 3 M1b Interactive Visualisation of Data #####
1077 #####
1078 #####
1079 #####
1080
1081 ub<- which(M1_N_SWD_RV[r~v])$SWD=="UB") # just Uninformed Backgrnd rows
1082 pts<- which(M1_N_SWD_RV[r~v])$SWD!="UB") # everything else rows
1083
1084 plot(M1_N_SWD_RV[r~v][ub,3],M1_N_SWD_RV[r~v][ub,4],col="grey", cex=.6) # Background (not training, but could be
test data)
1085 points(M1_N_SWD_RV[r~v][pts,3],M1_N_SWD_RV[r~v][pts,4],col="red") # Training Data (or not Background but could be
test too)
1086 points(Test_MIN_RV[r~v][ub,3],Test_MIN_RV[r~v][ub,4],col="purple",cex=.8) # Test data
1087
1088 rm(ub,pts)
1089 ## Option to save there as snapshots ###
1090 ## user should decide this and place name accordingly, here's a "skeleton"
1091 ## Just change the flag to TRUE if you want to ... or use other simpler snapshots tools
1092
1093 # PrintDraftFlag <- FALSE # TRUE #FALSE
1094 if(PrintDraftFlag=="TRUE"){ #dev.off() #dev.copy(file) #grDevices::jpeg
1095 dev.print(vwin,metafile,filename=paste(dir,plotout,
```

```
1096 "DraftPlot",
1097 ScriptNameCode,
1098 "M1_N_SWD_RV",rv,
1099 "Bakgd_Train",
1100 #obj", obj,PageNum, "of",NPages,
1101 ",.wmf", sep=""),res=80) # lower resolution this is just a draft
1102 dev.off() # need this after "printing it out!
1103 } # end of PrintDraftFlag
1104
1105 #####
1106 #####
1107 #####
1108 #####
1109 ### This section may seem superflous, but there is often a lot of confusion between what
the modeller has on their starting traning spreadsheet and what happens when you overlay
1110 ## either with GIS tools or Maxent.jar due to grid differences etc.
1111 ## with often "Overlapping" n training points have been "identified"
1112 ## good to simply write out the new ones from here
1113 #####
1114 #####
1115 #####
1116 ## and extract the overlayed and unique Test_MIN_RV data to match these below
1117
1118 n_SWD_RV <- list() # stands for small n, training only
1119 BN_SWD_RV <- list() # stands for both BigN, as well as Backgrnd Full N (includes training, n)
1120 BNn_SWD_RV <- list() # stands for BigN minus n (the pseudo absences, or UB's )
1121
1122 length(M1_N_SWD_RV)
1123 names(M1_N_SWD_RV)
1124 length(M1_N_SWD_RV[r~v])
1125 names(M1_N_SWD_RV[r~v])
1126
1127
1128 for(rv in seq(length(M1_N_SWD_RV)))
1129 {
1130 n_SWD_RV[r~v] <- M1_N_SWD_RV[r~v][M1_N_SWD_RV[r~v][2]=="UB",-1] # 1 right now is the ID
BN_SWD_RV[r~v] <- M1_N_SWD_RV[r~v][1:-1] # I think I then have to make this UB into all B or N
BNn_SWD_RV[r~v] <- M1_N_SWD_RV[r~v][M1_N_SWD_RV[r~v][2]=="UB",-1]
1133
1134 names(n_SWD_RV[r~v]) <- paste("n_",SWD_RV",rv, sep="") #
1135 names(BN_SWD_RV[r~v]) <- paste("BN_",SWD_RV",rv, sep="") #
1136 names(BNn_SWD_RV[r~v]) <- paste("BNn_",SWD_RV",rv, sep="") #
1137 cat("\n RV ",rv,
" n=",nrow(n_SWD_RV[r~v]),
" N-n=",nrow(BN_SWD_RV[r~v]),
" N=",nrow(BN_SWD_RV[r~v]),
sep="")
1141 } # end of rv
1142
1143 rv<-1
1144 dim(BN_SWD_RV[r~v])
1145 head(BN_SWD_RV[r~v])
1146 tail(BN_SWD_RV[r~v])
1147
1148 dim(n_SWD_RV[r~v])
1149 head(n_SWD_RV[r~v])
1150 tail(n_SWD_RV[r~v])
1151
1152 dim(BNn_SWD_RV[r~v])
1153 head(BNn_SWD_RV[r~v])
1154 head(BNn_SWD_RV[r~v])
1155
1156 for(rv in seq(length(M1_N_SWD_RV)))
1157 {write.csv(n_SWD_RV[r~v]),
paste(dir.swd, #
names(n_SWD_RV[r~v]),
".csv",
sep=""),
row.names=FALSE) #
1162 write.csv(BNn_SWD_RV[r~v]),
paste(dir.swd, #
```



```
1165 names(BN_N_SWD_RV[r.v]).
1166 "csv",
1167 sep=",",
1168 row.names=FALSE) #
1169 write.csv(BN_SWD_RV[r.v], #
1170 paste(dir.swd, #
1171 #, OptCode="N",
1172 names(BN_SWD_RV[r.v]), # this is the full bgkgrnd including the training
1173 "csv",
1174 sep=",",
1175 row.names=FALSE) #
1176
1177 write.csv(Test_MIN_RV[r.v][F-J], # 1 right now is the ID don't need this now,
1178 paste(dir.swd, #
1179 names(Test_MIN_RV[r.v]),
1180 "csv",
1181 sep=",",
1182 row.names=FALSE) #
1183
1184 } # end of rv
1185
1186 #####
1187
1188
1189 save.image(file=paste(dir.rdata,
1190 if(DAC>0)DAC,
1191 # OptCode "Obj_",
1192 ScriptNameCode,
1193 "N_SWD_Start", # object name
1194 "RData",sep="")) # Saving RData File
1195
1196 #####
1197 #####
1198 ##### M1b_LoadData - Part 4 -- Setting up M2_CoreLists RV ###
1199 #####
1200 ##### Setting up M2_CoreLists RV ###
1201 ##### for all the datasets you will consider
1202 ##### Datasets here is the same as Transformations (features)
1203 ##### Each Transformation may have 1 or more "Iterations
1204 ##### so each object may be a n.dim object (vector/dataframe)
1205 ##### or an n.dim object in a list
1206 #####
1207 #####
1208
1209 ScriptNameCode <- "M1b4"
1210
1211 InspFlag<-FALSE ## this is a flag to inspect things manually or not
1212 InstallFlag<-FALSE
1213 SaveTempFlag <- FALSE
1214
1215 # InsCode str(M1_N_SWD_RV[[rv]])
1216
1217 #_InsCode str(M1_N_SWD_RV)
1218
1219 #####
1220 #####
1221 ##### Setting up the first "Data List Object" for all the datasets you will consider
1222 ##### Datasets here is the same as Transformations
1223 ##### Each Transformation may have 1 or more "Iterations
1224 ##### so each object may be a n.dim object (vector/dataframe)
1225 ##### or an n.dim object in a list
1226 #####
1227 #####
1228
1229 DataNamesVect<-as.vector(c("ev.stack", # All "raw" data loaded up at the start up, in one big Dataframe, overlaid
```

```
1230 geographically. Converting Raster Pixels to "GIS vector data"
1231 "NoNARaw", # Removes all Rows that have NA in any of the EV, thereby reducing the "extent" of the study
1232 "RV", # Response Variable's Presence Data, to be used in the Training of the models
1233 "XY", # Just the xy -- used to have this as RVXY, and also had Target Bkgnd ... not in this version
1234 "EV", # Environmental Variables -- All the original raw layers
1235 "NumRaw", # Vector of Independent Variables - Continuous Type
1236 "FeRaw", # Vector of Independent Variables - Categorical Type
1237 "NTys", # removing ties from the original "continuous EV scaled and ranged 0-1
1238 "QntCis", # this will hold the Quantile Classes for each EV ... as a list, depending if QntMeth is more than 1
1239 "QntCisMid", # this will hold the MidValue of the Quantile Classes for each EV .
1240 "RVFrqNTys", # The "original" (after removing ties) frequency of presence of the RV, over the individual
1241 EV's quantile classes
1242 "RVOpNTys", # The "optima information" (QntCisMid value of the corresponding MaxFrequency for this
1243 data) for each Response Variable of the "original (without ties) dataset (before smoothing)
1244 "RVFrqSmthAll", # Frequency of presence of the RV, over the individual EV's quantile classes after applying
1245 Smoothing Algorithm for all iterations of smoothing
1246 "RVFrqSmthLast", # Last Smoothing Value: Frequency of presence of the RV, over the individual EV's
1247 quantile classes after applying Smoothing Algorithm for only the last iteration
1248 "RVOpSmthLast", # The "optima information" (QntCisMid value of the corresponding MaxFrequency for
1249 this data) for last smoothed FreqPres
1250 #From here on down, will be the "transformations" that get regrouped
1251 "FBin", # Binary conversion of any Categorical Variable into a series of DV made up of as many classes as
1252 each categorical variable contains, and a value of 0 or 1 accordingly
1253 "Lin", # Linear Variable: Continuous EV scaled and ranged 0-1 (used to be called Rng)
1254 "ZSk", # Zero Skewness Variable -- Monotonous
1255 "Dev", # Dev Deviation Variable, up to 3 currently
1256 "HFK", # HF Spline 1: Forward Hinge Variable with up to 20 knots
1257 "HRK", # HR Spline 2: Reverse Hinge Variable with up to 20 knots
1258 "Thk", # Th Spline 3: Threshold Variable with up to 20 knots
1259 "SWD_Test" # Added April 2013 -- Sticking at the end 2 reasons ... 1 to distinguish from Training, 2 to not
1260 have to change everything else at this point
1261
1262 n.datasets<-length(DataNamesVect)
1263 M2_Core<-as.list(1:n.datasets)
1264 names(M2_Core)<-DataNamesVect
1265 str(M2_Core)
1266 MIAPar
1267
1268 # this allows one to really consider NoNARaw the "starting point" ... after you've decided how many variables to consider
1269 # could list the EVALNames and ask the user which ones to use ...
1270 # also save this info in the MIAPar object
1271
1272 MIAPar$EVCntNames <- EVCntNames
1273 MIAPar$EVCatNames <- EVCatNames
1274 MIAPar$EVALNames <- EVALNames
1275 MIAPar$AlphaValues
1276
1277 rv <- 1
1278 M2_CoreLists RV<- list()
1279 for (rv in seq(length(M1_N_SWD_RV)))
1280 {
1281 M1_N_SWD <- M1_N_SWD_RV[rv]] ## keep the ID after all
1282 # InsCode str(M1_N_SWD)
1283 M2_Core<-as.list(1:n.datasets)
1284 names(M2_Core)<-DataNamesVect
1285
1286 #####
1287 #####
1288 #####
1289 ##### Data Type and No Data Assignment #####
1290 ##### let's keep "smaller objects"
1291 #####
1292 #####
1293 #####
1294 #####
1295 #dir.ev.cat
```

```
1287 length(na.omit(dlr.ev.cat))
1288 if(length(na.omit(dlr.ev.cat))>0){
1289   for (fct in seq(length(EVCatNames)))
1290   {
1291
1292     fct.pos <- which(names(M1_N_SW D)==EVCatNames[fct])
1293     M1_N_SW D[fct.pos] <- as.factor(M1_N_SW D[fct.pos]) # coerce them into factor type
1294   }
1295   } # end of if dlr.ev.cat
1296
1297 #_InsCode str(M1_N_SW D)
1298
1299 ### Section below be superflous because of No NA option in ev.stack option, however this is useful to draw attention to this
1300 issue
1301 ### especially since "No Data" values have different values depending on the type, date and source of the data
1302
1303 M1_N_SW D[M1_N_SW D == -3.40282346639e+038] <- NA # assigning R NA to No Data values defaults from other
1304 systems
1305 M1_N_SW D[M1_N_SW D == -3.4e+38] <- NA
1306 M1_N_SW D[M1_N_SW D == -9999] <- NA # assigning R NA to No Data values defaults from other systems
1307 M1_N_SW D[M1_N_SW D == -999] <- NA # assigning R NA to No Data values defaults from other systems
1308
1309 M1_N_SW D <- na.omit(M1_N_SW D) ## NOW REMOVE ANY NA's
1310 #_InsCode str(M1_N_SW D)
1311 #_InsCode str(M1_N_SW D_ALL)
1312
1313 #_InsCode names(M1_N_SW D)
1314 IDXY_EVAINames <- names(M1_N_SW D) ### Names of Standard IDXY and EVAINames, without the Response Variable
1315 names (Training and Test)
1316 All_ColNames <- names(M1_N_SW D) ### Names of Standard IDXY and EVAINames and everything else
1317
1318 #####
1319 #####
1320
1321 M2_Core$ev.stack <- ev.stack # these are the loaded rasters
1322 M2_Core$NoNARaw <- M1_N_SW D
1323 M2_Core$EV <- na.omit(M1_N_SW D[,5:length(M1_N_SW D)]) ## Dataframe with Data for all EV with ID and xy
1324 if(length(TestFile)>0){ M2_Core$EV <- na.omit(M1_N_SW D[,5:(length(M1_N_SW D)-1)])} ## Dataframe with Data for all
1325 EV with ID and xy
1326 # if (InspFlag==TRUE){
1327   head(M2_Core$NoNARaw)
1328   dim(M2_Core$NoNARaw)
1329 }
1330 #_InsCode unique(M2_Core$NoNARaw[1:1])
1331 #_InsCode length(na.omit(M2_Core$NoNARaw$ExitID))
1332
1333 if (SaveTempFlag==TRUE) {
1334   save.image(file=paste(dlr.rdata,
1335     Temp="_",
1336     if(DAC>0)DAC),
1337     ScriptNameCode,
1338     "DataLoaded",
1339     "AllObj_Complete", # object name
1340     ".RData",sep=".") # Saving RData File
1341   } # end of SaveTempFlag
1342
1343 #_InsCode str(M2_Core)
1344 ## consider having also the ID' and perhaps the RV and Test
1345 #####
1346 #####
1347 #####
1348 #####
1349 #####
1350 #####
1351 #####
1352 #####
1353 #####
1354 #####
1355 #####
1356 #####
1357 #####
1358 #####
1359 #####
1360 #####
1361 #####
1362 #####
1363 #####
1364 #####
1365 #####
1366 #####
1367 #####
1368 #####
1369 #####
1370 #####
1371 #####
1372 #####
1373 #####
1374 #####
1375 #####
1376 #####
1377 #####
1378 #####
1379 #####
1380 #####
1381 #####
1382 #####
1383 #####
1384 #####
1385 #####
1386 #####
1387 #####
1388 #####
1389 #####
1390 #####
1391 #####
1392 #####
1393 #####
1394 #####
1395 #####
1396 #####
1397 #####
1398 #####
1399 #####
1400 #####
1401 #####
1402 #####
1403 #####
1404 #####
1405 #####
1406 #####
1407 #####
1408 #####
1409 #####
1410 #####
1411 #####
1412 #####
1413 #####
1414 #####
1415 #####
1416 #####
1417 #####
```

```
1351 M2_Core$NumRaw <- as.vector(!:length(M2_Core$EV[1:1])) # just dummy for now
1352 #_InsCode length(EVCatNames)
1353 if(length(EVCatNames)>0){ M2_Core$FctRaw <- as.vector(!:length(M2_Core$EV[1:1]))} # reverse iknow but it works! --
1354 only do this if not NA
1355
1356 str(M2_Core$FctRaw)
1357 for (ev in seq(length(M2_Core$EV))) ## skip the first as it is the ID, and the XY
1358 {
1359   if(class(M2_Core$EV[ev])=="numeric"){
1360     M2_Core$NumRaw <- cbind(M2_Core$NumRaw,M2_Core$EV[ev])
1361   }
1362   if(class(M2_Core$EV[ev])=="factor"){
1363     M2_Core$FctRaw <- cbind(M2_Core$FctRaw,M2_Core$EV[ev])
1364   }
1365   } # end of splitting EV's by type
1366
1367 M2_Core$NumRaw <- M2_Core$NumRaw[1:] # removing the first column which was temporary
1368 M2_Core$FctRaw <- M2_Core$FctRaw[1:] # removing the first column which was temporary
1369 #_InsCode str(M2_Core$NumRaw)
1370 #_InsCode str(M2_Core$FctRaw)
1371 #####
1372 #####
1373 #####
1374 #####
1375 #####
1376 #####
1377 #####
1378 #####
1379 #####
1380 #####
1381 unique(M2_Core$NoNARaw$SVD)
1382 #_OptCode NotNAFreq <- count(M2_Core$NoNARaw$SVD) # require plyr # could save this as object or just checking
1383
1384 #_InsCode str(M2_Core$RV)
1385 M2_Core$RV <- data.frame(SVD=M2_Core$NoNARaw$SVD,stringsAsFactors=FALSE) # the name of the second variable
1386 always or use RVNames ## simplify to a shorter "Look Up Table" approach, and just name things RV1...5 etc
1387 #_InsCode class(M2_Core$RV)
1388 #_InsCode str(M2_Core$RV)
1389 #_InsCode unique(M2_Core$RV)
1390 # Alternative could be this
1391 #_AltCode
1392 NotTrain <- which(M2_Core$NoNARaw$SVD == 0|M2_Core$NoNARaw$SVD == 1|M2_Core$NoNARaw$SVD == "UB")
1393 #_AltCode length(NotTrain)
1394 #_AltCode M2_Core$RV[!rv] <- "B" # Anything not test == Background
1395 #_AltCode Test <- which(M2_Core$RV[Test[!rv]] == 0|M2_Core$RV[Test[!rv]] == 1)
1396 #_InsCode head(M2_Core$RV[Test[!rv][1]])
1397 #_InsCode names(M2_Core$RV)[rv] <- RVNames[rv]#names(Train.SVD[!rv])
1398
1399 #_AltCode names(M2_Core$RV)[rv] <- paste("RV",rv,RVNames[rv],sep="") # or RVNames
1400 #_AltCode names(M2_Core$RV[!rv][1]) <- paste("RV",rv,RVNames[rv],sep="")
1401
1402 # Inspection lfs for loop
1403 if (InspFlag==TRUE){
1404   str(M2_Core$RV)
1405   length(M2_Core$RV)
1406   names(M2_Core$RV)
1407   head(M2_Core$RV[1:1])
1408   class(M2_Core$RV[1])
1409   class(M2_Core$RV[1:1])
1410
1411 #_InsCode (M2_Core$RV[1:1];1:1000) == 1)
1412 #_InsCode which(M2_Core$RV[1:1]) == RVNames)
1413 #_AltCode TrainAlone <- M2_Core$RV[1:1] == 1
1414 } # end of InspFlag
1415 #####
1416 #####
1417 #####
1418 #####
1419 #####
1420 #####
1421 #####
1422 #####
1423 #####
1424 #####
1425 #####
1426 #####
1427 #####
1428 #####
1429 #####
1430 #####
1431 #####
1432 #####
1433 #####
1434 #####
1435 #####
1436 #####
1437 #####
1438 #####
1439 #####
1440 #####
1441 #####
1442 #####
1443 #####
1444 #####
1445 #####
1446 #####
1447 #####
```

```
1418 if(length(TestFile)!=0) # repeat same if test data present
1419 {
1420   # InsCode length(Test.Data)
1421   M2_Core$SWD_Test <- data.frame(SWD_Test=M2_Core$SWD_Test,stringsAsFactors=FALSE)
1422   # InsCode head(M2_Core$SWD_Test)
1423   # AltCode names(M2_Core$RVText)[rvf] <- names(Test.SWD[rvf[1]])
1424   } # End of TestFile present
1425
1426   if (InspFlag==TRUE){
1427     # InsCode str(M2_Core$RVText)
1428     names(M2_Core$SWD_Test)
1429     dim(M2_Core$RV)
1430     str(M2_Core$RV)
1431     names(M2_Core$RV)
1432     names(M2_Core$RV[1])
1433   } # end of InspFlag
1434   #####
1435   ##### extract TrainXY ###
1436   #####
1437   #####
1438   #####
1439   # InsCode names(M2_Core)
1440   M2_Core$XY <- dbind(M2_Core$NonARaw[c(3:4)]) ## X and Y Coordinates .3:4 when ID is present
1441
1442   M2_CoreLists_RV[rvf] <- M2_Core # Sep 2013 -- do this as temporary measure instead of "rescripting everything"
1443   names(M2_CoreLists_RV)[rvf] <- paste("M2_Core_RV",rv, sep="")
1444   M1_N_SWD_RV[rvf] <- M1_N_SWD# return it now that you've added the test and defined the factors -- you'll use this later
1445   # May 2014
1446   # InsCode str(M1_N_SWD_RV[rvf])
1447   rm(M2_Core.M1_N_SWD) # now we have everything in M2_CoreLists_RV
1448   } # end of RV
1449
1449   # InsCode str(M2_CoreLists_RV[rvf][1:7])
1450   str(M2_CoreLists_RV[rvf])
1451   length(M2_CoreLists_RV)
1452   names(M2_CoreLists_RV[rvf])
1453
1454   str(M1_N_SWD_RV[rvf])
1455
1456   #####
1457   ##
1458   ##### Clean up things that are not necessary along the way or it gets "too big" ###
1459   ##### and save objects outside if necessary
1460   #####
1461   ## removing ascii and rasters and large unneeded dataframes if wanted to
1462   ## but not sure you should always. Could set up an if statement
1463   #####
1464
1465   ##### Another way to keep track of all parameters, and data so far
1466   MIAVect <- objects(pattern="MIAPar")
1467   DirVect <- object(pattern="dir") # currently all scripts run on indiv dir object, but could upgrade to use this
1468   ParVect1 <- objects(pattern="Train") # Choosing numbers rather than Train, otherwise this get saved too
1469   ParVect2 <- objects(pattern="Test")
1470   ParVect3 <- objects(pattern="Bgnd") ## UB or N or background
1471   ParVect4 <- objects(pattern="BTC") ## BTG
1472
1473   M1b_ParVect <- na.omit(c(DirVect,MIAVect, #"MIAPar", # Adding MIAPar to DirVect in order to save all parameters specified
1474   thus far, also this are the "initial" ones
1475   ParVect1,ParVect2,
1476   ParVect3,ParVect4,
1477   "DAC")) # and location of input training and test data
1478   str(M1b_ParVect)
1479   # save everything first
1480   save.image(file=paste(dir.rdata,
1481   if(DAC>0){DAC},
1482   ScriptNameCode,
1483   " end ",
1484   "AllObj_Complete", # object name
```

```
1485   "RData",sep="") # Saving RData File
1486
1487   save(list=M1b_ParVect, ## this can be decided not to be "saved", as I think it will be rather large, but could do!
1488   file=paste(dir.rdata,
1489   "Obj_", # distinguish single objects from entire Images
1490   ScriptNameCode, # which script? which step, part in the process etc
1491   " ParVect", # object name
1492   "RData",sep="") # Saving RData File
1493
1494   ## AltCode rm(NotNAFreq)
1495   rm(ev.cat,ev.cnt,ev.asc, ev.stack, IDXY_EVALNames, InType) # #_OptCode this is also optional ... depending on how much
1496   # users want to do with raster files
1497   rm(fct.fct,pos.ev,rvt.fl)
1498
1498   save.image(file=paste(dir.rdata,
1499   if(DAC>0){DAC},
1500   ScriptNameCode,
1501   " end ",
1502   "AllObj_KeyOnes", # object name
1503   "RData",sep="") # Saving RData File
1504
1505   1505
```



```

1 #####
2 #####
3 #####
4 #####
5 #####
6 #####
7 #####
8 #####
9 #####
10 #####
11 #####
12 #####
13 #####
14 #####
15 #####
16 #####
17 #####
18 #####
19 #####
20 #####
21 #####
22 #####
23 #####
24 #####
25 #####
26 #####
27 #####
28 #####
29 #####
30 ScriptNameCode <- "M2a" #
31 SaveTempFlag <- FALSE
32 # _OptCode install.packages("dummies") MIAPar$dir.Rlib # used by the FBin Script
33 require(dummies,MIAPar$dir.Rlib) # used by the FBin Script
34 require(rgdal,MIAPar$dir.Rlib) #
35
36
37
38
39 length(M2_CoreLists_RV)
40 str(M2_CoreLists_RV[[r.v]]) ## with RV flat ...
41
42
43 for (v in seq(length(M2_CoreLists_RV)))
44 {
45   M2_Core <- M2_CoreLists_RV[[r.v]]
46   #_InsCode names(M2_Core)
47   RVName <- paste("RV_",if(r.v<10){0}, r.v, sep="") # RVNames[r.v]
48
49   #_InsCode str(M2_Core[[71]])
50   #_InsCode head(M2_Core[[71]])
51   #_InsCode length(na.omit(dir.ev.cat))
52
53   if(length(na.omit(dir.ev.cat))>0){ # only do this if there is factor data to convert
54
55     #####
56     #####
57     #####
58     ## Create a data frame containing only factor variables
59     ## that will be "opened up" and converted
60     ## into binary variables, one for each factor level
61     #####
62     #####
63     DataName <- "FBin"
64     dat <- which(names(M2_Core)==DataName) # which position of the M2_Core # this is where we start
65     FctRawColNames <- as.vector(names(M2_Core$FctRaw)) #
66     fct.num<-length(FctRawColNames)
67     DATA.fctr<-as.data.frame(M2_Core$FctRaw) # used to have to match by names but now not needed

```

```

67 #_InsCode str(DATA.fctr)
68
69 DATA.fct.bin.List<-as.list(rep(0,length(DATA.fctr)))
70 names(DATA.fct.bin.List)<- names(DATA.fctr)
71
72
73 for (f in seq(length(DATA.fctr)))
74 {
75   DATA.fct.bin<- as.data.frame(dummy(DATA.fctr[[f]],drop=FALSE))
76   names(DATA.fct.bin)<- paste(names(DATA.fctr[f]),"_",levels(DATA.fctr[[f]]), sep="")
77   str(DATA.fct.bin)
78   DATA.fct.bin.List[[f]] <- DATA.fct.bin # as in other scripts, could assign things directly to this list, this iteration of tools
79   translates most objects into simpler structures
80   write.csv(DATA.fct.bin.List[[f]],
81     file=paste(dir.fct,
82       if(DAC>0){DAC},
83       RVName,
84       #_OptCode dat,
85       #_OptCode "n",
86       "FBinEV",
87       f,
88       "n",
89       names(DATA.fctr[f]),
90       ".csv",sep=""),
91     row.names=FALSE)
92   # end of f
93
94   #_InsCode str(DATA.fct.bin.List[[f]])
95   M2_Core$FBin<-DATA.fct.bin.List
96   dim(M2_Core$FBin[[f]])
97   names(M2_Core$FBin)
98   names(M2_Core$FBin[[f]])
99   #_InsCode str(M2_Core$FBin)
100   #####
101   Write out resulting CSV ###
102   write.csv(M2_Core$FBin,
103     file=paste(dir.r.out,
104       if(DAC>0){DAC},
105       RVName, # not specific to RV
106       "n",
107       #_OptCode dat,
108       ScriptNameCode,
109       "n",
110       DataName,
111       "All.csv",
112       sep=""),
113     row.names=FALSE) # Up to user to decide file name!
114   #####
115   Extract a copy one level up
116   write.csv(M2_Core$FBin,
117     file=paste(dir.fct, # Extract a copy one level up
118       if(DAC>0){DAC},
119       #_OptCode RVName, # not specific to RV
120       #_OptCode "n",
121       ScriptNameCode,
122       "n",
123       DataName,
124       "All.csv",
125       sep=""),
126     row.names=FALSE) # Up to user to decide file name!
127   #####
128   Save Objects and Clean up ###
129   #####
130   #####
131   rm(DATA.fctr,DATA.fct.bin,DATA.fct.bin.List,FctRawColNames,f.fct.num)
132   M2_CoreLists_RV[[r.v]] <- M2_Core
133   rm(M2_Core)
134   } # end of if Categorical Data present
135   print(paste("checking rv",r.v))

```



```
136 } # end of RV
137
138 save.image(paste(dir.rdata,
139             if(DAC>0){DAC},
140             ScriptNameCode,
141             " ",
142             "FBin",
143             ".RData",sep=""))
144
145 #####
146 #####
147 ##### Next Script #####
148 #####
149 #####
150
151 ### M2b_Lin.R ##
152
153 #####
154 #####
155 #####
156 ##### MIA7 Module 2 - Preparing Derived Variables #####
157 ##### Rescaling each EV to a common range #####
158 ## from 0 to 1
159 ##### to produce the linear DVs
160 ##
161 ##### M2b_Lin.R
162 ##### April 2013
163 #####
164 #####
165 ScriptNameCode <- "M2b" # Just a 'short bit of ScriptName
166 # _OptCode require(scales,MIAPar$dir:Rlib) # used by the Lin and other scripts $smooth and Dev
167
168 for (rv in seq(length(M2_CoreLists_RV)))
169 {
170   M2_Core <- M2_CoreLists_RV[[rv]]
171   names(M2_Core)
172   RVName <- paste("RV",if(rv<10){0}, rv, sep="") # RVNames[rv]
173
174   #####
175   # _InsCode str(DatasetsObjects)
176   # _InsCode str(M2_Core)
177   names(M2_Core)
178   class(M2_Core[[1]])
179   head(M2_Core$FcrRaw)
180   head(M2_Core$NumRaw)
181
182   DataName<-"Lin"
183   dat <- which(names(M2_Core)==DataName) # this is where we start
184   DATA<-as.data.frame(M2_Core$NumRaw) #
185   head(DATA)
186   for (ev in seq(length(DATA)))
187   {
188     cat(paste("\nRange of EV",ev," ",sep=""))
189     cat(range(DATA[ev]))
190
191   }
192
193   #####
194   ##### Rescaling each variables values from 0 (min) to 1 (max) #####
195   #####
196   #####
197   for (ev in seq(length(DATA)))
198   {
199     DATA[ev]<-rescale(DATA[ev],
200                       c(0,1)) #
201   }
```

```
202
203 cat(paste("\nAfter Rescaling Range of EV",ev," ",sep=""))
204 cat(range(DATA[ev]))
205
206 #####
207 # _InsCode str(DATA)
208 M2_Core$Lin<-DATA
209 names(M2_Core$Lin)
210 dim(na.omit(M2_Core$Lin))
211 range(M2_Core$Lin)
212 # _InsCode str(M2_Core)
213 ev<-1
214
215 for (ev in seq(length(M2_Core$Lin)))
216 {
217   cat(paste("\nRange of EV",ev," ",sep=""))
218   cat(range(M2_Core$Lin[[ev]]))
219 }
220 #####
221 ##### Write out resulting CSV #####
222 ###
223 #####
224 write.csv(M2_Core$Lin,
225           file=paste(dir.rout,
226                     if(DAC>0){DAC},
227                     ScriptNameCode,
228                     " ",
229                     DataName,
230                     " ",
231                     RVName,
232                     ".CSV",
233                     sep=""),
234           row.names=FALSE) # Up to user to decide file name!
235
236
237 M2_CoreLists_RV[[rv]] <- M2_Core
238 rm(DATA,ev)
239 rm(M2_Core)
240 } # end of RV
241 #####
242 ##### Save Objects and Clean up #####
243 ###
244 #####
245 if(SaveTempFlag==TRUE){
246   save(M2_CoreLists_RV, #
247       file=paste(dir.rdata,
248                 "Obj", # distinguish single objects from entire Images
249                 ScriptNameCode,
250                 " ",
251                 "M2_Core", # object name
252                 # _OptCode
253                 DataName,
254                 ".RData",sep="")) #
255   } # end of SaveTempFlag condition
256
257
258 save.image(paste(dir.rdata,
259                 if(DAC>0){DAC},
260                 # _OptCode "AllObj", # object name
261                 ScriptNameCode, # script name or portion of it
262                 " ",
263                 DataName,
264                 ".RData",sep=""))
265
266 #####
267 ##### Next Script #####
268 #####
269 #####
270 #####
271
```

```
272 ##### M2c.R #####
273 #####
274 #####
275 #####
276 #####
277 #####
278 #####
279 #####
280 #####
281 #####
282 #####
283 #####
284 #####
285 #####
286 #####
287 #####
288 #####
289 #####
290 #####
291 #####
292 #####
293 #####
294 #####
295 #####
296 #####
297 #####
298 #####
299 #####
300 #####
301 #####
302 #####
303 #####
304 #####
305 #####
306 ScriptNameCode <- "M2c" # this can be just a "short bit of ScriptName
307
308 #require(e1071,MIAPar$dir:Rlib) # used by the zeroskewness script, also Smooth Dev
309 #require(scales,MIAPar$dir:Rlib) # used by the Lin and other scripts Smooth and Dev
310
311 #_OptCodestr(M2_Core) # Good to review your data
312 #_OptCodestr(M2_Core$Lin)
313 names(M2_CoreLists_RV)
314 names(M2_CoreLists_RV[[rv]])
315
316 #####
317
318 for (rv in seq(length(M2_CoreLists_RV))) # repeat for every rv
319 {
320   M2_Core <- M2_CoreLists_RV[[rv]]
321   names(M2_Core)
322   RVName <- paste("RV",if(rv<10){0, rv, sep=""}) # RVNames[rv]
323   DATA<-as.data.frame(M2_Core$Lin) #
324   DataName<-"ZSk"
325   dat <- which(names(M2_Core)==DataName)
326   names(M2_Core[dat])
327   names(DATa)
328   print(paste("checking rv",rv))
329
330   for (ev in seq(length(DATa)))
331   {
332     x<-DATA[ev]
333     scalex <- function(x, c)
334     {
335       if(skewness(x,na.rm=TRUE,type=2) < 0) # specify the type 2 ... different default on other software like excel
336         return(exp(-x))
337       else
338         return(log(x+c))
339     }
340     minskew <- function(x) {
341       cmin <- min(x)-10*(max(x)-min(x))
```

```
342 cmax <- max(x)+10*(max(x)-min(x));
343 if(skewness(x,na.rm=TRUE,type=2) >= 0 &&& cmin < -min(x))
344   cmin <- -min(x)
345   cmid <- (cmin+cmax)/2;
346   skew <- skewness(calex(x, cmid), na.rm=TRUE);
347   # specify the type 2 ... different default on other
348   # software like excel
349   while (abs(skew) > 1*10^-05 &&& min(abs(c(cmax, cmin)-cmid)) > 10^-10) {
350     #_OptCode print(c(cmin,cmid,cmax,skew));
351     sleft <- skewness(scalex(x, (cmid+cmin)/2), na.rm=TRUE,type=2)
352     sright <- skewness(scalex(x, (cmid+cmax)/2), na.rm=TRUE,type=2)
353     if (abs(sleft) < abs(skew) &&& abs(sleft) < abs(sright)) {
354       cmax <- cmid
355       skew <- sleft
356     } else if (abs(sright) < abs(skew)) {
357       cmin <- cmid
358       skew <- sright
359     } else {
360       cmin <- (cmin+cmid)/2;
361       cmax <- (cmax+cmid)/2;
362     }
363     cmid <- (cmin+cmax)/2;
364     return(list(c=cmid, skew=skew));
365   }
366   res<-minskew(x)
367   ##_DesCode Standardised skewness of the original (untransformed) variable
368   skewness(x, type=2)/(6/length(x))^0.5
369   ##_DesCode c value and skewness after zero skewness transformation
370   res$C
371   res$Skew
372   ##_DesCode standardised skewness of the transformed variable
373   (res$skew)/(6/length(x))^0.5
374   ##_DesCode Ranging and printout of the ranged variable
375   standy<-scalex(x,res$C)
376   rangx<-{standx-min(standx)}/(max(standx)-min(standx))
377   #_OptCode hist(rangx)
378   ##_DesCode We obtain the z-transformed version of original DATA. " as:
379   DATA[ev]<-rangx
380 } # end of EV loop
381 #####
382 #####
383 #####
384 M2_Core$ZSk<-DATA
385 #_InsCode str(M2_Core$ZSk)
386 head(M2_Core$ZSk)
387 head(M2_Core$Lin) # check that it is different!
388 range(M2_Core$ZSk[1])
389 write.csv(M2_Core$ZSk,
390           file=paste(dir:r.out,
391                     if(DAC>0){DAC,
392                       ScriptNameCode,
393                       " ",
394                       DataName,
395                       " ",
396                       RVName,
397                       ".CSV",
398                       sep=""},
399           row.names=FALSE) #
400
401 M2_CoreLists_RV[[rv]] <- M2_Core
402
403 #####
404 #####
405 #####
406 #####
407 #####
408 rm(standx,rangx,DATA,res,x,ev, minskew,scalex,M2_Core)
409 } # End of RV loop for M2_c
410
411 #####
412 #####
413 #####
414 #####
415 #####
416 #####
417 #####
418 #####
419 #####
420 #####
421 #####
422 #####
423 #####
424 #####
425 #####
426 #####
427 #####
428 #####
429 #####
430 #####
431 #####
432 #####
433 #####
434 #####
435 #####
436 #####
437 #####
438 #####
439 #####
440 #####
441 #####
442 #####
443 #####
444 #####
445 #####
446 #####
447 #####
448 #####
449 #####
450 #####
451 #####
452 #####
453 #####
454 #####
455 #####
456 #####
457 #####
458 #####
459 #####
460 #####
461 #####
462 #####
463 #####
464 #####
465 #####
466 #####
467 #####
468 #####
469 #####
470 #####
471 #####
472 #####
473 #####
474 #####
475 #####
476 #####
477 #####
478 #####
479 #####
480 #####
481 #####
482 #####
483 #####
484 #####
485 #####
486 #####
487 #####
488 #####
489 #####
490 #####
491 #####
492 #####
493 #####
494 #####
495 #####
496 #####
497 #####
498 #####
499 #####
500 #####
501 #####
502 #####
503 #####
504 #####
505 #####
506 #####
507 #####
508 #####
509 #####
510 #####
511 #####
512 #####
513 #####
514 #####
515 #####
516 #####
517 #####
518 #####
519 #####
520 #####
521 #####
522 #####
523 #####
524 #####
525 #####
526 #####
527 #####
528 #####
529 #####
530 #####
531 #####
532 #####
533 #####
534 #####
535 #####
536 #####
537 #####
538 #####
539 #####
540 #####
541 #####
542 #####
543 #####
544 #####
545 #####
546 #####
547 #####
548 #####
549 #####
550 #####
551 #####
552 #####
553 #####
554 #####
555 #####
556 #####
557 #####
558 #####
559 #####
560 #####
561 #####
562 #####
563 #####
564 #####
565 #####
566 #####
567 #####
568 #####
569 #####
570 #####
571 #####
572 #####
573 #####
574 #####
575 #####
576 #####
577 #####
578 #####
579 #####
580 #####
581 #####
582 #####
583 #####
584 #####
585 #####
586 #####
587 #####
588 #####
589 #####
590 #####
591 #####
592 #####
593 #####
594 #####
595 #####
596 #####
597 #####
598 #####
599 #####
600 #####
601 #####
602 #####
603 #####
604 #####
605 #####
606 #####
607 #####
608 #####
609 #####
610 #####
611 #####
612 #####
613 #####
614 #####
615 #####
616 #####
617 #####
618 #####
619 #####
620 #####
621 #####
622 #####
623 #####
624 #####
625 #####
626 #####
627 #####
628 #####
629 #####
630 #####
631 #####
632 #####
633 #####
634 #####
635 #####
636 #####
637 #####
638 #####
639 #####
640 #####
641 #####
642 #####
643 #####
644 #####
645 #####
646 #####
647 #####
648 #####
649 #####
650 #####
651 #####
652 #####
653 #####
654 #####
655 #####
656 #####
657 #####
658 #####
659 #####
660 #####
661 #####
662 #####
663 #####
664 #####
665 #####
666 #####
667 #####
668 #####
669 #####
670 #####
671 #####
672 #####
673 #####
674 #####
675 #####
676 #####
677 #####
678 #####
679 #####
680 #####
681 #####
682 #####
683 #####
684 #####
685 #####
686 #####
687 #####
688 #####
689 #####
690 #####
691 #####
692 #####
693 #####
694 #####
695 #####
696 #####
697 #####
698 #####
699 #####
700 #####
701 #####
702 #####
703 #####
704 #####
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
741 #####
742 #####
743 #####
744 #####
745 #####
746 #####
747 #####
748 #####
749 #####
750 #####
751 #####
752 #####
753 #####
754 #####
755 #####
756 #####
757 #####
758 #####
759 #####
760 #####
761 #####
762 #####
763 #####
764 #####
765 #####
766 #####
767 #####
768 #####
769 #####
770 #####
771 #####
772 #####
773 #####
774 #####
775 #####
776 #####
777 #####
778 #####
779 #####
780 #####
781 #####
782 #####
783 #####
784 #####
785 #####
786 #####
787 #####
788 #####
789 #####
790 #####
791 #####
792 #####
793 #####
794 #####
795 #####
796 #####
797 #####
798 #####
799 #####
800 #####
801 #####
802 #####
803 #####
804 #####
805 #####
806 #####
807 #####
808 #####
809 #####
810 #####
811 #####
812 #####
813 #####
814 #####
815 #####
816 #####
817 #####
818 #####
819 #####
820 #####
821 #####
822 #####
823 #####
824 #####
825 #####
826 #####
827 #####
828 #####
829 #####
830 #####
831 #####
832 #####
833 #####
834 #####
835 #####
836 #####
837 #####
838 #####
839 #####
840 #####
841 #####
842 #####
843 #####
844 #####
845 #####
846 #####
847 #####
848 #####
849 #####
850 #####
851 #####
852 #####
853 #####
854 #####
855 #####
856 #####
857 #####
858 #####
859 #####
860 #####
861 #####
862 #####
863 #####
864 #####
865 #####
866 #####
867 #####
868 #####
869 #####
870 #####
871 #####
872 #####
873 #####
874 #####
875 #####
876 #####
877 #####
878 #####
879 #####
880 #####
881 #####
882 #####
883 #####
884 #####
885 #####
886 #####
887 #####
888 #####
889 #####
890 #####
891 #####
892 #####
893 #####
894 #####
895 #####
896 #####
897 #####
898 #####
899 #####
900 #####
901 #####
902 #####
903 #####
904 #####
905 #####
906 #####
907 #####
908 #####
909 #####
910 #####
911 #####
912 #####
913 #####
914 #####
915 #####
916 #####
917 #####
918 #####
919 #####
920 #####
921 #####
922 #####
923 #####
924 #####
925 #####
926 #####
927 #####
928 #####
929 #####
930 #####
931 #####
932 #####
933 #####
934 #####
935 #####
936 #####
937 #####
938 #####
939 #####
940 #####
941 #####
942 #####
943 #####
944 #####
945 #####
946 #####
947 #####
948 #####
949 #####
950 #####
951 #####
952 #####
953 #####
954 #####
955 #####
956 #####
957 #####
958 #####
959 #####
960 #####
961 #####
962 #####
963 #####
964 #####
965 #####
966 #####
967 #####
968 #####
969 #####
970 #####
971 #####
972 #####
973 #####
974 #####
975 #####
976 #####
977 #####
978 #####
979 #####
980 #####
981 #####
982 #####
983 #####
984 #####
985 #####
986 #####
987 #####
988 #####
989 #####
990 #####
991 #####
992 #####
993 #####
994 #####
995 #####
996 #####
997 #####
998 #####
999 #####
1000 #####
```

```
411 #_InsCode head(M2_CoreLists_RV[[rv]]$ZSk)
412 #_InsCode range(M2_CoreLists_RV[[rv]]$ZSk[1:7])
413 #####
414 #####
415 #_AltCode if(SaveTempFlag==TRUE){ # optional if you don't want to save all the times
416 save(M2_CoreLists_RV, ##
417 file=paste(dir.rdata,
418 "Obj_", # distinguish single objects from entire Images
419 ScriptNameCode,
420 "M2_CoreLists_", # object name
421 DataName,
422 ".RData",sep="")) #
423 #_AltCode } #if(SaveTempFlag==TRUE)
424
425 save.image(paste(dir.rdata,
426 if(DAC>0){DAC},
427 #_AltCode "AilObj_", # object name
428 ScriptNameCode, # script name or portion of it
429 #_AltCode "_", RVName,
430 " ", DataName,
431 #_AltCode DataName,
432 ".RData",sep="")) #
433 #####
434 ##### Next Script #####
435 #####
436 #####
437 #####
438 #####
439 #####
440 ##### M2d_PrepDV_FrqPres.R #####
441
442 #####
443 #####
444 #####
445 #####
446 ##### MIA7 Module 2 - Preparing Derived Variables #####
447 Exploring Frequency of Presence
448 ## Remove Ties, Quantile, Smoothing, and Deviation Variables in 1 script
449 #####
450 ##### S06_M2d_NTys_Qnt_Smth_Dev.R
451 ##### April 2013
452 ##### sabgls@yahoo.com
453 #####
454 #####
455 #####
456 #####
457 require(e1071) # used by the zero skewness script, also Smooth Dev#
458 require(scales) # used by mg and Dev
459 ScriptNameCode <- "M2cd" # this can be just a "short bit of ScriptName
460
461 ## load previous dataset if needed
462 #_OptCode objlist<-ls()
463 #_OptCode if (length(objlist) == 0){load("~/IDA/A04_Cris/RData/A04_M2_d_NTys_Image.RData")}
464
465 rv<-1
466 str(M2_CoreLists_RV[[rv]])
467 length(M2_CoreLists_RV)
468
469 for (rv in seq(length(M2_CoreLists_RV)))
470 {
471 M2_Core <- M2_CoreLists_RV[[rv]]
472 #_InsCode names(M2_Core)
473 RVName <- paste("RV",if(rv<10){},rv,sep="") # RVNames[rv]
474 #####
475 #####
476 #####
477 #####
478 #####
479 #####
480 #####
```

```
481 DataName<-NTys" #
482 dat <- which(names(M2_Core)==DataName) #_DevCode dat is just a counter of "which dataset from the datasets list
object is being used
483 #####
484 cat(paste("\nSorting and checking for duplicates in your data, for ",RVName,sep=""))
485 DATA2Sort<-cbind(SortCol:=nrow(M2_Core$Lin),M2_Core$Lin)
486 dim(DATA2Sort)
487 num.dups.ev <- lapply(M2_Core$Lin, function(c) sum(duplicated(c))) # counts the number of duplicates so you can see
how what to expect
489 orig.dups.ev <- num.dups.ev
490 cat(paste("\nHere are the number of duplicated records in your data (per column)",sep=""))
491 str(orig.dups.ev) # original number of duplicates in your data
492
493 #####
494 oev==2 ### oev is ordering column, which is e (EV) + 1 due to the additional sorting column
495 for (oev in oev:(length(DATA2Sort))) #
496 {
497 Ordrrd_EV <- DATA2Sort[order(DATA2Sort[,oev],c(1,oev))] ##1 is the "sorting column" to join back to, col is the
"ordered/sorted" ev column
498 z <- (Ordrrd_EV) ## use z to "simplify" future references and future objects
499 Unik <-unique(z[,2])
500 #_InsCode str(Unik)
501 if (length(Unik)< length(z[,2])) {
502 cat(paste("\n ",length(Unik), " records with duplicates found in EV",oev-1," (", names(DATA2Sort[,oev]),")",sep=""))
503 for (u in (seq(Unik)))
504 {
505 z_pos u <- which(z[,2]==Unik[u]) # the individual positions of the first"group" of duplicates
506 #_InsCode length(z_pos_u) # the number of duplicates in that group
507 if (length(z_pos_u) > 1) { # only interested in removing duplicates(even more than 2 perhaps) actually, not going
through all the uniques
#_InsCode print("dups") #_InsCode str(z_pos_u)
value u <- Unik[u] # the value
value u_1 <- Unik[u+1] # at +1
z_group2change_u <- z[z_pos_u,2]
head(z_group2change_u)
for (g in seq(length(z_group2change_u)-1)))
{
z_group2change_u[g+1] <- z_group2change_u[g]+.0000000000000001 #_AltCode .0000000001 #
} # end of each value within each duplicated group to increment
z[z_pos_u,2] <- z_group2change_u # now that the values have been "minimally incremented" ... place them back in
original place
}
} # end of "greater than 2" duplicates if statement (length(z_pos_u) > 10)
} # end of u "unik/duplicated group to increment"
} # exit the remove duplicates IF statement
DATA2Sort[,oev] <- z[order(z[,1],2)] # return value for that ordering column used which is the second column
}
#_InsCode str(DATA2Sort)
#####
#_AltCode checking if dups have been removed
527 #_AltCode dups<-duplicated(M2_Core$NTys[2])
528 #_AltCode dups.pos <-which(dups==TRUE)
529 #_AltCode check <-M2_Core$NTys[2]
530 #_AltCode pos<- which(check==value) # 2 is ok!
531 #####
532 #####
533 #####
534 M2_Core$NTys <- DATA2Sort[,1] # remove the sorting column
535 num.dups.noties <- lapply(M2_Core$NTys, function(c) sum(duplicated(c))) # counts the number of dups so you can see
how what to expect
536 cat(paste("\nAfter Removing Ties, the number of duplicated records in your data (per column) should be zero",sep=""))
537 str(num.dups.noties) # your no ties data
538 #_InsCode str(M2_Core$NTys)
539 #####
540 ##### Cleaning up and saving
541 #####
542
543 rm(DATA2Sort,Ordrrd_EV,Unik,value,u_1,z_pos_u,z_group2change_u,z,oev,ev,g,u)
544 #####
```

```

545 names(M2_Core)
546 str(M2_Core$NTYs)
547 #####
548 #####
549 #####
550 ##### Quantile Portion script   ###
551 ##### Setting objects for quantile analysis
552 #####
553 #####
554 #####
555 qn.meth <- MIAPar$qn.meth  ## transfer level down to keep scripts as simple as possible
556 qnt.cls.seq <- seq(1/qn.meth,1, by=(1/qn.meth))
557 dir.qn <- paste(dir.r.out,"Qnt_", MIAPar$qn.meth, "\", sep="")  # in here too, as you may want to repeat this
558 dir.create(dir.qn)
559 ## with NoTys data -- ties have been removed
560 Dtfm <- M2_Core$NTYs  # set this at the beginning so you can "change" less things!
561 DataName<- "RVFrqNTYs"
562 dat <- which(names(M2_Core)==DataName)  # which position of the M2_Core # this is where we start
563 QntCls <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  # a matrix of quantile classes x EV
564 names(QntCls) <- names(Dtfm)
565 QntClsFr <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  # a matrix of quantile classes x EV
566 names(QntClsFr) <- names(Dtfm)
567 QntClsTo <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  # a matrix of quantile classes x EV
568 names(QntClsTo) <- names(Dtfm)
569 QntClsMid <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  # a matrix of quantile classes x EV
570 names(QntClsMid) <- names(Dtfm)
571
572 for (ev in seq(length(Dtfm)))
573 {
574   QntCls[ev] <- quantile(Dtfm[ev],qnt.cls.seq) # repeat quantile class
575   QntClsFr[ev] <- c(min(Dtfm[ev]),QntCls[1:qn.meth-1,ev])  ##### the first component is the "minimum of your data",
                    which is 0 in a ranged dataset,
576   QntClsTo[ev] <- c(QntCls[1:qn.meth-1,ev],max(Dtfm[ev]))  # this means that it will always look at min and max
577   QntClsMid[ev] <- (QntClsFr[ev]+QntClsTo[ev])/2  # this is the "mid value" of this quantile class
578 } # end of e loop, number of environmental variables
579 #_InsCode QntCls # head(QntClsMid)#head(QntCls)
580 #####
581 #####
582 ### Calculate the frequency
583 ### of presence of the Response Variable
584 ### per quantile class
585 ###
586 ### For this we need to
587 ### attach the presence data to the EVData.
588 ### This is similar to MaxEnt's SWD (Sample with Data)"
589 #####
590 #####
591 #####
592 cat(paste("\nCalculating Frequency of RV",if(rv<10){3, rv, "per EV quantile class"}))
593 DtfmSWD <- cbind(M2_Core$RV,
594                 M2_Core$XY,
595                 Dtfm) # need the SWD as we're checking the density of RV per quantile class
596
597 DtfmSWD.Pres <- DtfmSWD[DtfmSWD[,1]!="UB",] # extract only the portion of EV where the RV (always col 1 for SWD)
598 #_InsCode str(DtfmSWD.Pres)
599 DtfmSWD.Pres <- DtfmSWD.Pres[-c(2:3)] # remove the RV and XY which were used just to figure out where present
600 dim(DtfmSWD.Pres)
601
602 SumRV <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  #data.frame()  # holds a count of RV rows (therefore
603 presences) per EV's quantile classes
604 SumEV <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  # holds a count of EV rows in the EV Notices data per EV's
                    quantile classes
605 RVxEVFrq <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))
606 RVxEVOptima <- as.data.frame(array(dim=c(qn.meth,length(Dtfm))))  # RV's MaximumFrequency Class 1, the value 2: qntl
                    class 3, optimum value for EV at that Class
607 names(SumRV) <- names(Dtfm)
608 names(SumEV) <- names(Dtfm)
609 names(RVxEVFrq) <- names(Dtfm)
610 names(RVxEVOptima) <- names(Dtfm)

```

```

611 #_InsCode str(Dtfm)
612 #_InsCode str(DtfmSWD.Pres)  # ev+1
613 for (ev in seq(length(Dtfm)))
614 {
615   i<-1
616   for (i in 1:qn.meth)  ### the quantile class
617   {
618     SumRV[1,ev] <- sum(DtfmSWD.Pres[ev+1]>QntClsFr[1,ev]&DtfmSWD.Pres[ev+1]<=QntClsTo[1,ev])  # sum of RV pres
619     x quantile class
620     SumEV[1,ev] <- sum(Dtfm[ev]>QntClsFr[1,ev]&Dtfm[ev] <= QntClsTo[1,ev])  #sum EV pixels per quantile class
621     RVxEVFrq[1,ev] <- SumRV[1,ev]/SumEV[1,ev]  ### RV as a proportion of total in that class
622   }
623   RVxEVOptima[1,ev] <- max(RVxEVFrq[ev])  # 1 is the actual value of the highest frequency of presence
624   RVxEVOptima[2,ev] <- as.integer(which.max(RVxEVFrq[ev]))  # 2 is the quantile class where this occurs
625   RVxEVOptima[3,ev] <- QntClsMid[RVxEVOptima[2,ev],ev]  # 3 is the value from ev.qncls.mid for that quant-pos
626 }
627 ##### save these objects into new M2_Core Objects -- inside rv loop her
628 M2_Core$RVFrqNTYs <- RVxEVFrq
629 M2_Core$RVOptNTYs <- RVxEVOptima
630 ##### save these objects into new M2_Core Objects -- these are independent of rv
631 M2_Core$QntCls <- QntCls
632 head(M2_Core$QntCls)
633 M2_Core$QntClsFr <- QntClsFr
634 M2_Core$QntClsTo <- QntClsTo
635
636 M2_Core$QntClsMid <- QntClsMid
637 rownames(M2_Core$QntCls) <- rownames(M2_Core$QntCls, do.NULL = FALSE, prefix="QnCls_")
638 write.csv(M2_Core$QntCls,file=paste(dir.qn,
639                                     "QntileClass_Per_EV.csv",sep=""))
640 write.csv((M2_Core$QntCls),file=paste(dir.qn,
641                                     "RV",if(rv<10){3, rv,
642                                     "QuantileClass_Trans.csv",sep=""))
643 #####
644 ##### Run Smoothing algorithm  #####
645 #####
646 #####
647 #####
648 #####
649 ##### define number of "smoothing" rounds
650 num.smoothing <- MIAPar$num.smoothing
651 YsmoothList <- as.list(rep(0,num.smoothing))
652
653 DataName<- "RVFrqSmthAll" #
654 dat <- which(names(M2_Core)==DataName)
655 names(M2_Core[dat])
656
657 cat(paste("\nSmoothing out FreqPres of RV",if(rv<10){3, rv, "\n"}))
658 RVName <- RVName[rv]
659 YData <- M2_Core$RVFrqNTYs
660 YSmooth <- as.data.frame(array(dim=c(dim(YData))))
661 colnames(YSmooth) <- colnames(YData)
662 rownames(YSmooth) <- rownames(YData)
663 #_InsCode dim(YSmooth) #_InsCode head(YSmooth)
664
665 YSmoothFunct <- function(YData,i)
666 return(YSmooth[i,1]<-YData[i-2,1]+ 2*YData[i-1,1]+ 3*YData[i,1]+ 2*YData[i+2,1])/9)
667
668 #####
669 ##### Smoothing loops  #####
670 #####
671 #####
672 #####
673 s=1  # Smoothing Number starting point
674 for (s in 1:num.smoothing)

```



```
676 {
677   cat(paste("Smoothing Round",s,"\n"))
678   if (s > 1) YData <- Y$smoothList[[s-1]]
679   i=3
680   for (i in 1:(qn.meth-2)) ##
681     {
682       Y$smooth[i,] <- Y$smoothFunc(YData,i)
683     }
684
685   Y$smooth[1:2,] <- YData[1:2,]
686   Y$smooth[(qn.meth-1):qn.meth,] <- YData[(qn.meth-1):qn.meth,]
687
688   # InsCode head(Y$smooth)
689   # InsCode head(YData)
690   Y$smoothList[[1]]<- Y$smooth
691   names(Y$smoothList[s])<-paste("Snth",s, sep="") # optional
692   names(Y$smoothList[[1]])<- colnames(YData)
693   names(Y$smoothList[s]) <- paste("Qnt", qn.meth, "_Snth", s, sep="")
694 }
695
696 Y$smoothList[[3]][1:20,] ##
697 # InsCode Y$smoothList[[2]][1:20,8:14] ## _InsCode M2_Core$Qnt[1:20,8:14]
698 # _InsCode s<-1
699 # InsCode dim(Y$smoothList[[2]])
700 # InsCode Y$smoothList[[1]][1:5] ## just checking, this is the "syntax for lists"
701 M2_Core$RVFrq$nthAll <- Y$smoothList
702 #####
703 #####
704 ## Defining the new Optima -- ###
705 ## from the smoothed tied removed freq pres x quantile class object
706 #####
707 #####
708
709 Y <- M2_Core$RVFrq$nthAll[[num.smoothing]] ## should be the "last one" of the smoothed data
710 # InsCode length(Y) # _InsCode dim(Y)
711
712 RVx$EVFrq$nth <- Y # the new smoothed things!
713 RVx$EVOptima$nth <- as.data.frame(array(dim=c(3,length(Y)))) # RV's MaximumFrequency Class 1. the value 2: qnt1
714   class 3: optimum value for EV at that Class
715   names(RVx$EVFrq$nth) <- names(Y)
716   names(RVx$EVOptima$nth) <- names(Y)
717
718 # ev<-1
719 for(ev in seq(length(Y)))
720 {
721   RVx$EVOptima$nth[,ev] <- max(RVx$EVFrq$nth[,ev]) # 1 is the actual value of the highest frequency of presence
722   RVx$EVOptima$nth[,2:ev] <- as.integer(which.max(RVx$EVFrq$nth[,ev])) # 2 is the quantile class where this occurs
723   RVx$EVOptima$nth[,3:ev] <- QntClsMid[RVx$EVOptima$nth[,2:ev],ev] # 3 is the value from ev.qnds.mid for that quant-pos
724 }
725
726 M2_Core$RVFrq$nthLast <- RVx$EVFrq$nth
727 M2_Core$RVOp$nthLast <- RVx$EVOptima$nth
728 head(M2_Core$RVFrq$nthLast)
729 head(M2_Core$RVFrqN1ty)
730
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 rm(qnt.ds.seq,QntCls,QntClsFr,QntClsTo,QntClsMid,i)
738 rm(RVx$EVFrq,RVx$EVOptima,SumEV,SumRV)
739 rm(Dfrfm,Dfrfm$WD,Dfrfm$WD.Pres)
740 ##### cleaning up $Smoothing
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```
741 rm(s,Y)
742 rm(YData)
743 rm(Y$smooth)
744 rm(Y$smoothList)
745 rm(Y$smoothFunc)
746 rm(RVx$EVFrq$nth)
747 rm(RVx$EVOptima$nth)
748 #####
749 #####
750 #####
751 ## Next section is deriving the deviation Variables after quantiles and smoothing ###
752
753 #####
754 #####
755 #####
756 ##### MIA7 Module 2 - Preparing Derived Variables #####
757 ##### Deviation Variables #####
758 #####
759 #####
760 #####
761 #####
762 ##### Deviation Variables will be built if and only the frequency of presence shows a distinct optimum
763 ##### The script does the smoothing of the Quantile data (which gets the no ties data) then determines
764 ##### if there's an optima of sorts, and if so, then "select" these as potential DevPred Input,
765 ##### finally calculates what the actual value of these devpred with the formula provided
766 #####
767 #####
768 #####
769 #####
770 DataName <- "Dev"
771 dai <- which(names(M2_Core)==DataName)
772 dir.dev <- paste(dir.r.out,"DevPred\\",sep="")
773 dir.create(dir.dev)
774
775 #####
776 #####
777 #####
778 ##### Building the Deviation Variables #####
779 #####
780 #####
781 #####
782 #####
783 #####
784 ## DesCode what data do you want to start with?
785 ## DesCode choose if you want to do the smoothing first or if this has already been done
786
787 ## _InsCode str(M2_Core$RVFrq$nthLast)
788 #####
789 #####
790 #####
791 ## picking the smoothed out values ###
792 #####
793 #####
794 #####
795 num.smoothing <- MIA7Par$num.smoothing
796 # AitCode num.smoothing <- 3 # or you choose
797 RVName <- paste("RV",if(rv<10){0},rv, sep="")
798 cat(paste("\nDetermining Dev Variables for RV:",if(rv<10){0},rv, RVName))
799 Y <- M2_Core$RVFrq$nthAll[[num.smoothing]] ## should be the "last one" of the smoothed data
800 # AitCode Y <- M2_Core$RVFrq$nthLast #
801 # _InsCode length(Y) # _InsCode dim(Y)
802
803 #####
804 #####
805 #####
806 #####
807 #####
808 #####
809 #####
810 #####
811 #####
812 #####
813 #####
814 #####
815 #####
816 #####
817 #####
818 #####
819 #####
820 #####
821 #####
822 #####
823 #####
824 #####
825 #####
826 #####
827 #####
828 #####
829 #####
830 #####
831 #####
832 #####
833 #####
834 #####
835 #####
836 #####
837 #####
838 #####
839 #####
840 #####
841 #####
842 #####
843 #####
844 #####
845 #####
846 #####
847 #####
848 #####
849 #####
850 #####
851 #####
852 #####
853 #####
854 #####
855 #####
856 #####
857 #####
858 #####
859 #####
860 #####
861 #####
862 #####
863 #####
864 #####
865 #####
866 #####
867 #####
868 #####
869 #####
870 #####
871 #####
872 #####
873 #####
874 #####
875 #####
876 #####
877 #####
878 #####
879 #####
880 #####
881 #####
882 #####
883 #####
884 #####
885 #####
886 #####
887 #####
888 #####
889 #####
890 #####
891 #####
892 #####
893 #####
894 #####
895 #####
896 #####
897 #####
898 #####
899 #####
900 #####
901 #####
902 #####
903 #####
904 #####
905 #####
906 #####
907 #####
908 #####
909 #####
910 #####
911 #####
912 #####
913 #####
914 #####
915 #####
916 #####
917 #####
918 #####
919 #####
920 #####
921 #####
922 #####
923 #####
924 #####
925 #####
926 #####
927 #####
928 #####
929 #####
930 #####
931 #####
932 #####
933 #####
934 #####
935 #####
936 #####
937 #####
938 #####
939 #####
940 #####
941 #####
942 #####
943 #####
944 #####
945 #####
946 #####
947 #####
948 #####
949 #####
950 #####
951 #####
952 #####
953 #####
954 #####
955 #####
956 #####
957 #####
958 #####
959 #####
960 #####
961 #####
962 #####
963 #####
964 #####
965 #####
966 #####
967 #####
968 #####
969 #####
970 #####
971 #####
972 #####
973 #####
974 #####
975 #####
976 #####
977 #####
978 #####
979 #####
980 #####
981 #####
982 #####
983 #####
984 #####
985 #####
986 #####
987 #####
988 #####
989 #####
990 #####
991 #####
992 #####
993 #####
994 #####
995 #####
996 #####
997 #####
998 #####
999
```



```
935 #####
936 #####
937 rm(X.Y.Xopt,XoptAll)
938 rm(d1,dx,ln,num,Inside_TopFreqEdges,TopFreqIn,Edge,Edges,Edge.num,FreqEdges(Mat.FreqInMat)
939 rm(DevPredPotInPut,DevPredInput,DevPred,DevPredList)
940 rm(DevPredNames,DevPredVarNum,EVPos,OptPos,DevExp)
941 M2_CoreLists_RV[!rv] <- M2_Core
942 rm(M2_Core)
943 } # end of Datasets lists RV loop
944 rv<-1
945 str(M2_CoreLists_RV[!rv])
946 #####
947
948 save.image(paste(dir,rdata,
949 if(DAC>0){DACf,
950 #_OptCode"Obj_",
951 ScriptNameCode,
952 #_OptCode" ",dat,
953 " ",DataName,
954 "rData",sep="})
955
956 save(M2_CoreLists_RV, ##
957 file=paste(dir,rdata,
958 "Obj_",
959 ScriptNameCode," ",
960 "M2_CoreLists_RV", # object name
961 #_OptCode" ",dat,
962 " ",
963 DataName,
964 "rData",sep="})) #
965 ## may choose to pick specific objects to save as before
966 #_OptCode Obj2Save <- ls(
967 #_OptCode as.data.frame(Obj2Save)
968 #####
969 #####
970 #####
971 Next Script #####
972 #####
973 #####
974 #####
975 ## The M2e script is not included here, as it is needs to be run separately to inspect and modify
976 # However the variable preparations can continue with the next script, M2f
977 ## The sequence is kept to urge users to inspect their data, but does not have to occur all the times
978 #####
979 ## M2_All_MultiRV Part f
980 #####
981 #####
982 #####
983 Next Script #####
984 #####
985 #####
986 #####
987 #####
988 #####
989 #####
990 MIAT Module 2 - Preparing Derived Variables #####
991 ##### All Spline Variables #####
992 #####
993 S08_M2f_Splines.R #####
994 April 2013 #####
995 sabgis@yahoo.com #####
996 #####
997 #####
998 ##### Object do you want to work with? or inspect your data
999 # load("D:/IDA/A01_T4/RData/A01_M2d_Dev.RData")
1000 #####
```

```
1001 ScriptNameCode<-"M2f"
1002 rv<-1
1003 as.data.frame(names(M2_CoreLists_RV[!rv])) # easier to read
1004
1005 rv<-1
1006 for (rv in seq(length(M2_CoreLists_RV)))
1007 {
1008 ## Extract the splines into its own list to help with large list objects, this enables removal of M2_Core if necessary
1009 M2_Core <- M2_CoreLists_RV[!rv]
1010 names(M2_Core)
1011 RVName <- paste("RV",
1012 if(rv<10){0,
1013 rv, sep=""}) # RVNames[rv]
1014
1015 DataName <- "HFk" # for Splines start with Hinge Forward
1016 dat <- which(names(M2_Core)==DataName)
1017 #dat <- as.numeric(readline("\nWhich object do you want to work with? Type the corresponding DatasetNum \n "))
1018
1019 Knots<-MIAPar$Knots # or N.Iterations
1020 #####
1021 ##### Spline prep #####
1022 ## Setting the variables for
1023 ## hinge forward/backward and threshold loops #####
1024 #####
1025 #####
1026
1027 #####
1028 ##### Hinge Forward #####
1029 ##### # This version is "simpler" and split up, as lists of lists for very large datasets take longer to process
1030 ##### than as dataframes and removing temporary bits
1031 #####
1032 #####
1033 Splist <- list0 # this name can be generic (Spl) or specific (HF). Generic as it will be temporary, contents will be transferred
1034 # to the list
1035 Splistfr <- as.data.frame(array(dim=c(length(M2_Core$lin[!rv]),Knots)))# this name can be generic or specific to HF as it
1036 # will be temporary, contents will be transferred to the list
1037 #InsCode str(Splistfr)
1038
1039 lp <- 1 # list position
1040 for (lp in seq(length(M2_Core$lin)))
1041 {
1042 i=1
1043 for (i in 1:Knots) # for (i in seq(length(kvalues))) # if you have this figured out ahead
1044 {
1045 k<-(2*i-1)/(2*Knots) #
1046 # DesCode k in this case is the MIAPar$KValues at iteration i (or at knot i)
1047 Splistfr[,i] <- ((M2_Core$lin[!rv])*(M2_Core$lin[!rv]>k))/(1-k)
1048 names(Splistfr[i]) <- paste(names(M2_Core$lin[!rv]),"_",DataName,if(i<10){0},i,sep="") #
1049 }
1050 Splist[!lp] <- Splistfr
1051 }
1052 names(Splist) <- names(M2_Core$lin)
1053 #_InsCode str(Splistfr)
1054 M2_Core[!dat] <- Splist
1055 rm(Splistfr,Splist)
1056 #_InsCode str(M2_Core$HFk)
1057 #####
1058 ##### Hinge Reverse
1059 ##### aka as Backward Hinge, but handy to keep things ordered
1060 #####
1061 dat <- dat +1 # doing it 1 by one rather than as a loop ... simpler for now
1062 DataName<-names(M2_Core[dat])
1063 Splist <- list0 # this name can be generic (Spl) or specific (HF). Generic as it will be temporary, contents will be transferred
1064 # to the list
```

```

1064 SplDftr <- as.data.frame(array(dim=c(length(M2_Core$Lin[[1]], Knots))))# this name can be generic or specific to HF as it
1065 will be temporary, contents will be transferred to the list
1066 #InsCode str($pIDftr)
1067
1068 lp <- 1 # list position
1069 for (lp in seq(length(M2_Core$Lin)))
1070 {
1071   i=1
1072   for (i in 1:Knots) # for (i in seq(length(kvalues))) # if you have this figured out ahead
1073   {
1074     k<-(2*i-1)/(2*Knots)
1075     #_DesCode k in this case is the MIAPar$KValues at iteration i (or at knot i)
1076     SplDftr[i,j] <- (M2_Core$Lin[[i]-k]*((M2_Core$Lin[[ip]]>k))/(-k)))/(-k)
1077     names($pIDftr)[j] <- paste(names(M2_Core$Lin[ip]), "_", DataName, if(i<10){}, i, sep="") #
1078   }
1079   SplList[[ip]] <- SplDftr
1080 }
1081 names($pList) <- names(M2_Core$Lin)
1082 #_InsCode str($pIDftr)
1083 M2_Core[[dat]] <- $pList
1084 rm($pDftr, $pList)
1085 #_InsCode str(M2_Core[[dat]])
1086
1087 #####
1088 #####
1089 ##### Threshold DVs #####
1090 #####
1091 #####
1092
1093 dat = dat +1 # doing it 1 by one rather than as a loop ... simpler for now
1094 DataName<-names(M2_Core[dat])
1095 SplList <- list() # this name can be generic (Spl) or specific (HF). Generic as it will be temporary, contents will be transferred
to the list
1096 SplDftr <- as.data.frame(array(dim=c(length(M2_Core$Lin[[1]], Knots))))# this name can be generic or specific to HF as it
will be temporary, contents will be transferred to the list
1097 #InsCode str($pIDftr)
1098
1099 lp <- 1 # list position
1100 for (lp in seq(length(M2_Core$Lin)))
1101 {
1102   i=1
1103   for (i in 1:Knots) # for (i in seq(length(kvalues))) # if you have this figured out ahead
1104   {
1105     k<-(2*i-1)/(2*Knots)
1106     #_DesCode k in this case is the MIAPar$KValues at iteration i (or at knot i)
1107     SplDftr[i,j] <- (M2_Core$Lin[[i]-k]*((M2_Core$Lin[[ip]]>k))/(-k)))/(-k)
1108     names($pIDftr)[j] <- paste(names(M2_Core$Lin[ip]), "_", DataName, if(i<10){}, i, sep="") #
1109   }
1110   SplList[[ip]] <- SplDftr
1111 }
1112 names($pList) <- names(M2_Core$Lin)
1113 #_InsCode str($pIDftr)
1114 #_InsCode str($pList)
1115 M2_Core[[dat]] <- $pList
1116 rm($pDftr, $pList)
1117 #_InsCode str(M2_Core[[dat]])
1118 #_InsCode str(M2_Core)
1119 #####
1120 ##### Tidy and save #####
1121 #####
1122 #####
1123 #####
1124
1125 M2_CoreLists RV[[rv]] <- M2_Core
1126 rm(DataName, M2_Core)

```

```

1127 rm(k.i.lp, Knots)
1128 } # End of RV for splines loop
1129 #####
1130 #####
1131 ##### Extra bit extracting the Splines out from the M2_Core for several reasons,
1132 including memory issues on large datasets, and to upgrade M2 object
1133 ##### Could be incorporated directly by user
1134 #####
1135
1136 M2f_SplineList_RV <- list() # Extra object to handle large data files (will superceed the big 'M2_Core' object)
1137
1138 rv<-1 # decide where to start
1139 for (rv in seq(length(RVNames))) ##
1140 {
1141   M2f_SplineList_RV[[rv]] <- list(HFk=1, HRk=2, Thk=3)
1142   #_InsCode length(M2f_SplineList_RV) #_InsCode names(M2f_SplineList_RV[[rv]]) #_InsCode
length(M2_CoreLists_RV[[rv]]$HFk)
1143   spl<-1
1144   for (spl in seq(length(M2f_SplineList_RV[[rv]])))
1145   {
1146     M2f_SplineList_RV[[rv]][spl] <- M2_CoreLists_RV[[rv]][names(M2f_SplineList_RV[[rv]][spl])]
1147     DataName <- names(M2f_SplineList_RV[[rv]][spl])
1148     cat(paste("\nChecking Feature", DataName))
1149     #_InsCode str(M2f_SplineList_RV[[rv]][spl]) #_InsCode str(M2f_SplineList_RV[[rv]][spl])
1150     #_InsCode names(M2f_SplineList_RV[[rv]]) #_InsCode str(M2f_SplineList_RV[[rv]])
1151     #_InsCode length(M2f_SplineList_RV[[rv]][spl]) #_InsCode length(M2f_SplineList_RV[[rv]])
1152     #_InsCode names(M2f_SplineList_RV[[rv]][spl]) #_InsCode str(M2f_SplineList_RV[[rv]][spl])
1153   } # end of spline loop
1154   names(M2f_SplineList_RV)[rv] <- RVNames[[rv]]
1155 } # end of rv loop for resorting splines
1156
1157 #_InsCode str(M2f_SplineList_RV)
1158 save(M2f_SplineList_RV, ## this is the equivalent to M2_Core$HFk, HRk and Thk, again split out to help data efficiency
file=paste(dir, rdta,
#_OptCode if(DAC>0){DAC},
"Obj_",
ScriptNameCode, "_",
"SplineList_RV", # object name
".RData", sep="") #
#####
1166 save.image(paste(dir, rdta,
if(DAC>0){DAC},
ScriptNameCode, "_",
"AllSplinesReady",
".RData", sep=""))
1172 #####
1173 #####
1174 ##### Next Script #####
1175 #####
1176 #####
1177 #####
1178 ##### Script 509_M2g #####
1179 #####
1180 #####
1181 #####
1182 #####
1183 ##### MIAT Module 2 - Preparing Derived Variables #####
1184 ##### Selection of Spline Variables #####
1185 ##### maximum background removed, thus default
1186 #####
1187 ##### 509_M2g_MP3R_SelSpline.R #####
1188 ##### April 2013 #####
1189 ##### sabgs@yahoo.com #####
1190 #####
1191 #####
1192 ##### Script 509_M2g #####
1193 ##### As in this section we start running maxent.jar, place this check and instruct use
1194 #####
1195 jarFileFlag <- paste(dir.in, "maxent.jar", sep="") # checking if maxent can be run (normally not part of your script)

```

```

1196
1197 if file.exists(jarFileFlag)==FALSE{cat("!!!! WARNING !!!! \nPlease place a copy of maxent.jar in your dir.in directory\n!!!!  

1198 WARNING !!!!) else {"All's good, carry on"}  

1199  

1200 ScriptNameCode <- "M2g" # this can be just a 'short bit of ScriptName  

1201 SaveTempFlag <- FALSE # keep code for complicated issues or testing temporary things but remove from final option  

1202  

1203 str(M2f_SplineList_RV)  

1204 length(M2f_SplineList_RV[[1]][1:10])  

1205  

1206 memory.limit(80000000)  

1207 gc0  

1208  

1209 #####  

1210 #####  

1211 ##### M2g_MP3R_EV_Spl_RV object #####  

1212 ##### is the "key to run and track and interpreting trail of models"  

1213 ##### the MP3R object will be used to:  

1214 ##### Model building/selection, model parametrization, run and rank each 1 variable models using Maxent.jar )  

1215 ##### SPA stands for "single variable analysis", the start of this  

1216 ##### Testing which spline Variables to select #####  

1217  

1218 ##### However splitting the RUNNING and the Evaluating to literally DECOUPLE things  

1219 ##### This may be merged later #####  

1220 #####  

1221 #####  

1222 M2g_MP3R_EV_Spl_RV <-list0 ##### is the "key to run and track and interpreting trail of models"  

1223 M2g_FinNameList_Spl_RV <-list0 # this will store the names of the "best" Variables selected  

1224 M2g_FinNamesVect_Spl_RV <-list0 # all three at once [[dat]]  

1225  

1226  

1227 dir.max.M2g_All <- paste(dir.max,  

1228 "M2g_All",  

1229 "\\",  

1230 sep="")  

1231 dir.create(dir.max.M2g_All)  

1232  

1233 OutNames <- as.vector(c("ListPosition", # Dataframe's position in the list (also EV position)  

1234 "DataframeName", # Loop2Names .. Just 1 item at a time  

1235 "ListDir",  

1236 "PVDir", # PV stands for Predictor Variable, also known as DV Derived Variable. Currently PV is used  

1237 throughout, not advisable to change to DV yet at this stage  

1238 "PV_Column", #  

1239 "PV_Name", # Loop1Names  

1240 "RV_Name",  

1241 #_OptCode "MaxRunBar", # don't need to save this right now, but keep column location to match other  

1242 model tracking tables  

1243 "MaxOut",  

1244 "MaxResultsFile",  

1245 ## add lambda file and results csv so we can pull them out and map them later  

1246 "MaxResE",  

1247 "MaxResN",  

1248 "MaxResAUC",  

1249 "SPA_dfr", # spa stands for "single predictor analysis", the start of this  

1250 "FTVE", ## Fraction of the Variation Explained, also FTVA in manuscript. Currently used throughout several  

1251 scripts, not advisable to change at this stage  

1252 "FTTest",  

1253 #_OptCode "_P_Value", # not used in M2g but later in M3  

1254 "Sign_PVName")) # Significant variable for that run  

1255  

1256 #####  

1257 #####  

1258 #####  

1259 #####  

1260 #####  

1261 #####  

1262 #####  

1263 #####  

1264 #####  

1265 #####  

1266 #####  

1267 #####  

1268 #####  

1269 #####  

1270 #####  

1271 #####  

1272 #####  

1273 #####  

1274 #####  

1275 #####  

1276 #####  

1277 #####  

1278 #####  

1279 #####  

1280 #####  

1281 #####  

1282 #####  

1283 #####  

1284 #####  

1285 #####  

1286 #####  

1287 #####  

1288 #####  

1289 #####  

1290 #####  

1291 #####  

1292 #####  

1293 #####  

1294 #####  

1295 #####  

1296 #####  

1297 #####  

1298 #####  

1299 #####  

1300 #####  

1301 #####  

1302 #####  

1303 #####  

1304 #####  

1305 #####  

1306 #####  

1307 #####  

1308 #####  

1309 #####  

1310 #####  

1311 #####  

1312 #####  

1313 #####  

1314 #####  

1315 #####  

1316 #####  

1317 #####  

1318 #####  

1319 #####  

1320 #####  

1321 #####  

1322 #####  

1323 #####  

1324 #####  

1325 #####  

1326 #####  

1327 #####  

1328 #####  

1329 #####  

1330 #####  

1331 #####  

1332 #####  

1333 #####  

1334 #####  

1335 #####  

1336 #####  

1337 #####  

1338 #####  

1339 #####  

1340 #####  

1341 #####  

1342 #####  

1343 #####  

1344 #####  

1345 #####  

1346 #####  

1347 #####  

1348 #####  

1349 #####  

1350 #####  

1351 #####  

1352 #####  

1353 #####  

1354 #####  

1355 #####  

1356 #####  

1357 #####  

1358 #####  

1359 #####  

1360 #####  

1361 #####  

1362 #####  

1363 #####  

1364 #####  

1365 #####  

1366 #####  

1367 #####  

1368 #####  

1369 #####  

1370 #####  

1371 #####  

1372 #####  

1373 #####  

1374 #####  

1375 #####  

1376 #####  

1377 #####  

1378 #####  

1379 #####  

1380 #####  

1381 #####  

1382 #####  

1383 #####  

1384 #####  

1385 #####  

1386 #####  

1387 #####  

1388 #####  

1389 #####  

1390 #####  

1391 #####  

1392 #####  

1393 #####  

1394 #####  

1395 #####  

1396 #####  

1397 #####  

1398 #####  

1399 #####  

1400 #####  

1401 #####  

1402 #####  

1403 #####  

1404 #####  

1405 #####  

1406 #####  

1407 #####  

1408 #####  

1409 #####  

1410 #####  

1411 #####  

1412 #####  

1413 #####  

1414 #####  

1415 #####  

1416 #####  

1417 #####  

1418 #####  

1419 #####  

1420 #####  

1421 #####  

1422 #####  

1423 #####  

1424 #####  

1425 #####  

1426 #####  

1427 #####  

1428 #####  

1429 #####  

1430 #####  

1431 #####  

1432 #####  

1433 #####  

1434 #####  

1435 #####  

1436 #####  

1437 #####  

1438 #####  

1439 #####  

1440 #####  

1441 #####  

1442 #####  

1443 #####  

1444 #####  

1445 #####  

1446 #####  

1447 #####  

1448 #####  

1449 #####  

1450 #####  

1451 #####  

1452 #####  

1453 #####  

1454 #####  

1455 #####  

1456 #####  

1457 #####  

1458 #####  

1459 #####  

1460 #####  

1461 #####  

1462 #####  

1463 #####  

1464 #####  

1465 #####  

1466 #####  

1467 #####  

1468 #####  

1469 #####  

1470 #####  

1471 #####  

1472 #####  

1473 #####  

1474 #####  

1475 #####  

1476 #####  

1477 #####  

1478 #####  

1479 #####  

1480 #####  

1481 #####  

1482 #####  

1483 #####  

1484 #####  

1485 #####  

1486 #####  

1487 #####  

1488 #####  

1489 #####  

1490 #####  

1491 #####  

1492 #####  

1493 #####  

1494 #####  

1495 #####  

1496 #####  

1497 #####  

1498 #####  

1499 #####  

1500 #####  

1501 #####  

1502 #####  

1503 #####  

1504 #####  

1505 #####  

1506 #####  

1507 #####  

1508 #####  

1509 #####  

1510 #####  

1511 #####  

1512 #####  

1513 #####  

1514 #####  

1515 #####  

1516 #####  

1517 #####  

1518 #####  

1519 #####  

1520 #####  

1521 #####  

1522 #####  

1523 #####  

1524 #####  

1525 #####  

1526 #####  

1527 #####  

1528 #####  

1529 #####  

1530 #####  

1531 #####  

1532 #####  

1533 #####  

1534 #####  

1535 #####  

1536 #####  

1537 #####  

1538 #####  

1539 #####  

1540 #####  

1541 #####  

1542 #####  

1543 #####  

1544 #####  

1545 #####  

1546 #####  

1547 #####  

1548 #####  

1549 #####  

1550 #####  

1551 #####  

1552 #####  

1553 #####  

1554 #####  

1555 #####  

1556 #####  

1557 #####  

1558 #####  

1559 #####  

1560 #####  

1561 #####  

1562 #####  

1563 #####  

1564 #####  

1565 #####  

1566 #####  

1567 #####  

1568 #####  

1569 #####  

1570 #####  

1571 #####  

1572 #####  

1573 #####  

1574 #####  

1575 #####  

1576 #####  

1577 #####  

1578 #####  

1579 #####  

1580 #####  

1581 #####  

1582 #####  

1583 #####  

1584 #####  

1585 #####  

1586 #####  

1587 #####  

1588 #####  

1589 #####  

1
```

```

1260 M2g_FinNameList_Spl <- list(HFk=1,HRk=2,Thk=3)
1261 M2g_FinNamesVect_Spl <- list(HFk=1,HRk=2,Thk=3)
1262
1263 M2_Core <- M2_CoreLists_RV[[rv]] #
1264 PosRV <- which(M2_Core$RV=="UB") # Just the training (Response Variable)
1265 PosUB <- which(M2_Core$RV=="UB") # Full (original) ALL UB positions
1266 # InsCode length(PosUB)
1267 if (length(PosUB)>= MIAPar$Mx8kgNum) { # only pick random sample if more than the
  Mx8kgNum
  PosUB <- na.omit(sample(PosUB, MIAPar$Mx8kgNum))} # Row position Randomly selected from full UB
  # InsCode length(PosUB)
1268
1269 RVXY <- rbind(M1_N_SWD_RV [[rv]][PosRV.2:4], # the Training only
1270               M1_N_SWD_RV [[rv]][PosUB.2:4]) ## the Uninformed background
1271
1272 # InsCode str(RVXY)
1273 RVName <- paste("RV",
1274                 if(rv<10){0},
1275                 rv.sep="")
1276
1277
1278 dir.max.M2g <- paste(dir.max.M2g_All,
1279                     "M2g_RV",
1280                     if(rv<10){0},
1281                     rv, # so it's numeric order rather than alphabetic (add the 0 if statemtn later to be foolproof) -- oops .. no
1282                     # " ", RVName, # keep generic and shorter
1283                     "\\",
1284                     sep="")
1285
1286 dir.create(dir.max.M2g)
1287
1288 #####
1289 ##### M2g_Spline loop (Loop1) - Setting up the Data #####
1290 #####
1291 spl<-1
1292 for (spl in seq(length(M2f_SplineList_RV[[rv]])))
1293 {
1294   cat(paste("\nResponse Variable: ", if(rv<10){0}, rv.RVName))
1295   DataName <- names(M2f_SplineList_RV[[rv]][spl])
1296   cat(paste("\nChecking Feature", DataName))
1297   M2g_DataList <- M2f_SplineList_RV[[rv]][spl]
1298   Loop2Names <- names(M2g_DataList) # the items to be fed in LOOP 2 # list of lists or of dataframes
1299   Loop2 <- length(M2g_DataList) # number of items in this object
1300
1301   M2g_MP3R_EV <- as.list(!:length(Loop2Names))
1302   names(M2g_MP3R_EV) <- Loop2Names
1303
1304   dir.max.M2g_Spl <- paste(dir.max.M2g,
1305                             "AM_",DataName,
1306                             "\\",
1307                             sep="")
1308   dir.create(dir.max.M2g_Spl)
1309
1310   #####
1311   ##### Individual EV loop (Loop2) #####
1312   #####
1313   for (i2 in seq(length(M2g_DataList))) # Loop2
1314   {
1315     cat(paste("\nLoading EV", i2, names(M2g_DataList[i2])) #
1316         ## Note: ONLY random sample of Maximum Background points (eg 10000) (Mx8kgNum) not the entire
1317         background. # AltCode Threfull background SPADData <- M2g_DataList[i2])
1318     SPADData <- rbind(M2g_DataList[i2][PosRV.1, # the Training only - note WITH ID
1319                     M2g_DataList[i2][PosUB.1, # uninformed background
1320                     str(SPADData)
1321     SPADDataName <- names(M2g_DataList[i2]) # name of the first item of the list, which is now a dataframe
1322     Loop1Names <- as.vector(names(M2g_DataList[i2])) # the items to be fed in LOOP 1 (a dataframe in the first position
1323     in the list)
1324     Loop1 <- length(Loop1Names) # number of items in this object
1325

```



```

1326 M2g_MP3R <- as.data.frame(array(dlim=c(length(Loop1Names),length(OutNames))))
1327 colnames(M2g_MP3R) <- OutNames
1328 rownames(M2g_MP3R) <- Loop1Names
1329
1330 dir.max.M2g_Spl.data<-paste(dir.max.M2g_Spl, ### Dataset folder to be fed into Maxent.jar
1331 "EV",
1332 I2,
1333 " ",
1334 Loop2Names[2],
1335 "\\",
1336 sep="") ## EV or Dat for datasets depending
1337 dir.create(dir.max.M2g_Spl.data)
1338 #####
1339 #####
1340 ##### Running Maxent.jar #####
1341 ### for each individual variable for each EV (Loop2)
1342 ##### and grouping by Spline (Loop 1)
1343 #####
1344 #####
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
211
```

-21-

```

1395 M2g_MaxSetting <- "mxent.jar threads=6 pictures=FALSE writebackgroundpredictions=FALSE autofeature=FALSE
1396 beta_categorical=0 beta_hinge=0 beta_lqp=0 betamultiplier=0 beta_threshold=0 beta_dodamp=FALSE
1397 extrapolate=FALSE hinge=FALSE jackknife=FALSE nowarnings outputformat=raw outputgrids=FALSE
1398 perspedresults=TRUE product=FALSE quadratic=FALSE randomtestpoints=0 redoifexists=TRUE
1399 removeduplicates=TRUE responsecurves=FALSE responsecurvesexponent=FALSE skipifexists=FALSE
1400 threshold=FALSE writeplotdata=FALSE samplesfile=""
1401 MaxRunBat <- paste("java -Xmx512m jar ", # choose memory to match system
1402 dir.in,
1403 M2g_MaxSetting,
1404 Max.Sample.Name,
1405 " outputdirectory=",
1406 MaxOut,
1407 " environmentalayers=",
1408 Max.SWD.Name,
1409 " visible=FALSE", # stick at the end, easier to find and toggle later in bat file
1410 " autorun",
1411 sep="")
1412 write(MaxRunBat,
1413 paste(dir.max.M2g_Spl.data,
1414 "EV",
1415 I2,
1416 " ",
1417 " ",
1418 " ",
1419 " ",
1420 " ",
1421 " ",
1422 " ",
1423 " ",
1424 " ",
1425 " ",
1426 " ",
1427 " ",
1428 " ",
1429 " ",
1430 " ",
1431 " ",
1432 " ",
1433 " ",
1434 " ",
1435 " ",
1436 " ",
1437 " ",
1438 " ",
1439 " ",
1440 " ",
1441 " ",
1442 " ",
1443 " ",
1444 " ",
1445 " ",
1446 " ",
1447 " ",
1448 " ",
1449 " ",
1450 " ",
1451 " ",
1452 " ",
1453 " ",
1454 " ",
1455 " ",
1456 " ",
1457 " ",
1458 " ",
1459 " ")
1460
1461 #gc()
1462 # gc(TRUE)
1463
1464 system(paste(MaxRunBat,wait=T) # This runs Maxent.jar from a batch file
1465
1466 #####
1467 #####
1468 #####
1469 #####
1470 #####
1471 #####
1472 #####
1473 #####
1474 #####
1475 #####
1476 #####
1477 #####
1478 #####
1479 #####
1480 #####
1481 #####
1482 #####
1483 #####
1484 #####
1485 #####
1486 #####
1487 #####
1488 #####
1489 #####
1490 #####
1491 #####
1492 #####
1493 #####
1494 #####
1495 #####
1496 #####
1497 #####
1498 #####
1499 #####
1500 #####
1501 #####
1502 #####
1503 #####
1504 #####
1505 #####
1506 #####
1507 #####
1508 #####
1509 #####
1510 #####
1511 #####
1512 #####
1513 #####
1514 #####
1515 #####
1516 #####
1517 #####
1518 #####
1519 #####
1520 #####
1521 #####
1522 #####
1523 #####
1524 #####
1525 #####
1526 #####
1527 #####
1528 #####
1529 #####
1530 #####
1531 #####
1532 #####
1533 #####
1534 #####
1535 #####
1536 #####
1537 #####
1538 #####
1539 #####
1540 #####
1541 #####
1542 #####
1543 #####
1544 #####
1545 #####
1546 #####
1547 #####
1548 #####
1549 #####
1550 #####
1551 #####
1552 #####
1553 #####
1554 #####
1555 #####
1556 #####
1557 #####
1558 #####
1559 #####
1560 #####
1561 #####
1562 #####
1563 #####
1564 #####
1565 #####
1566 #####
1567 #####
1568 #####
1569 #####
1570 #####
1571 #####
1572 #####
1573 #####
1574 #####
1575 #####
1576 #####
1577 #####
1578 #####
1579 #####
1580 #####
1581 #####
1582 #####
1583 #####
1584 #####
1585 #####
1586 #####
1587 #####
1588 #####
1589 #####
1590 #####
1591 #####
1592 #####
1593 #####
1594 #####
1595 #####
1596 #####
1597 #####
1598 #####
1599 #####
1600 #####
1601 #####
1602 #####
1603 #####
1604 #####
1605 #####
1606 #####
1607 #####
1608 #####
1609 #####
1610 #####
1611 #####
1612 #####
1613 #####
1614 #####
1615 #####
1616 #####
1617 #####
1618 #####
1619 #####
1620 #####
1621 #####
1622 #####
1623 #####
1624 #####
1625 #####
1626 #####
1627 #####
1628 #####
1629 #####
1630 #####
1631 #####
1632 #####
1633 #####
1634 #####
1635 #####
1636 #####
1637 #####
1638 #####
1639 #####
1640 #####
1641 #####
1642 #####
1643 #####
1644 #####
1645 #####
1646 #####
1647 #####
1648 #####
1649 #####
1650 #####
1651 #####
1652 #####
1653 #####
1654 #####
1655 #####
1656 #####
1657 #####
1658 #####
1659 #####
1660 #####
1661 #####
1662 #####
1663 #####
1664 #####
1665 #####
1666 #####
1667 #####
1668 #####
1669 #####
1670 #####
1671 #####
1672 #####
1673 #####
1674 #####
1675 #####
1676 #####
1677 #####
1678 #####
1679 #####
1680 #####
1681 #####
1682 #####
1683 #####
1684 #####
1685 #####
1686 #####
1687 #####
1688 #####
1689 #####
1690 #####
1691 #####
1692 #####
1693 #####
1694 #####
1695 #####
1696 #####
1697 #####
1698 #####
1699 #####
1700 #####
1701 #####
1702 #####
1703 #####
1704 #####
1705 #####
1706 #####
1707 #####
1708 #####
1709 #####
1710 #####
1711 #####
1712 #####
1713 #####
1714 #####
1715 #####
1716 #####
1717 #####
1718 #####
1719 #####
1720 #####
1721 #####
1722 #####
1723 #####
1724 #####
1725 #####
1726 #####
1727 #####
1728 #####
1729 #####
1730 #####
1731 #####
1732 #####
1733 #####
1734 #####
1735 #####
1736 #####
1737 #####
1738 #####
1739 #####
1740 #####
1741 #####
1742 #####
1743 #####
1744 #####
1745 #####
1746 #####
1747 #####
1748 #####
1749 #####
1750 #####
1751 #####
1752 #####
1753 #####
1754 #####
1755 #####
1756 #####
1757 #####
1758 #####
1759 #####
1760 #####
1761 #####
1762 #####
1763 #####
1764 #####
1765 #####
1766 #####
1767 #####
1768 #####
1769 #####
1770 #####
1771 #####
1772 #####
1773 #####
1774 #####
1775 #####
1776 #####
1777 #####
1778 #####
1779 #####
1780 #####
1781 #####
1782 #####
1783 #####
1784 #####
1785 #####
1786 #####
1787 #####
1788 #####
1789 #####
1790 #####
1791 #####
1792 #####
1793 #####
1794 #####
1795 #####
1796 #####
1797 #####
1798 #####
1799 #####
1800 #####
1801 #####
1802 #####
1803 #####
1804 #####
1805 #####
1806 #####
1807 #####
1808 #####
1809 #####
1810 #####
1811 #####
1812 #####
1813 #####
1814 #####
1815 #####
1816 #####
1817 #####
1818 #####
1819 #####
1820 #####
1821 #####
1822 #####
1823 #####
1824 #####
1825 #####
1826 #####
1827 #####
1828 #####
1829 #####
1830 #####
1831 #####
1832 #####
1833 #####
1834 #####
1835 #####
1836 #####
1837 #####
1838 #####
1839 #####
1840 #####
1841 #####
1842 #####
1843 #####
1844 #####
1845 #####
1846 #####
1847 #####
1848 #####
1849 #####
1850 #####
1851 #####
1852 #####
1853 #####
1854 #####
1855 #####
1856 #####
1857 #####
1858 #####
1859 #####
1860 #####
1861 #####
1862 #####
1863 #####
1864 #####
1865 #####
1866 #####
1867 #####
1868 #####
1869 #####
1870 #####
1871 #####
1872 #####
1873 #####
1874 #####
1875 #####
1876 #####
1877 #####
1878 #####
1879 #####
1880 #####
1881 #####
1882 #####
1883 #####
1884 #####
1885 #####
1886 #####
1887 #####
1888 #####
1889 #####
1890 #####
1891 #####
1892 #####
1893 #####
1894 #####
1895 #####
1896 #####
1897 #####
1898 #####
1899 #####
1900 #####
1901 #####
1902 #####
1903 #####
1904 #####
1905 #####
1906 #####
1907 #####
1908 #####
1909 #####
1910 #####
1911 #####
1912 #####
1913 #####
1914 #####
1915 #####
1916 #####
1917 #####
1918 #####
1919 #####
1920 #####
1921 #####
1922 #####
1923 #####
1924 #####
1925 #####
1926 #####
1927 #####
1928 #####
1929 #####
1930 #####
1931 #####
1932 #####
1933 #####
1934 #####
1935 #####
1936 #####
1937 #####
1938 #####
1939 #####
1940 #####
1941 #####
1942 #####
1943 #####
1944 #####
1945 #####
1946 #####
1947 #####
1948 #####
1949 #####
1950 #####
1951 #####
1952 #####
1953 #####
1954 #####
1955 #####
1956 #####
1957 #####
1958 #####
1959 #####
1960 #####
1961 #####
1962 #####
1963 #####
1964 #####
1965 #####
1966 #####
1967 #####
1968 #####
1969 #####
1970 #####
1971 #####
1972 #####
1973 #####
1974 #####
1975 #####
1976 #####
1977 #####
1978 #####
1979 #####
1980 #####
1981 #####
1982 #####
1983 #####
1984 #####
1985 #####
1986 #####
1987 #####
1988 #####
1989 #####
1990 #####
1991 #####
1992 #####
1993 #####
1994 #####
1995 #####
1996 #####
1997 #####
1998 #####
1999 #####
2000 #####

```

-22-


```

1526 ##### Applying the model selection criteria:
1527 l1<-3
1528 for (i1 in 3:(nrow(M2g_MP3R)-2)) # Needs to start at 3 and stop at 18, for twenty knots - can't compare the first and last 2
1529 {
1530   if (M2g_MP3R$FTVE[i1] > M2g_MP3R$FTVE[i1+1]&M2g_MP3R$FTVE[i1] > M2g_MP3R$FTVE[i1+2]&M2g_MP3R$
1531     FTVE[i1] > M2g_MP3R$FTVE[i1-1]&M2g_MP3R$FTVE[i1] > M2g_MP3R$FTVE[i1-2]) {M2g_MP3R$sign_PVName[i1]
1532     <-as.character(rownames(M2g_MP3R[i1,]))
1533   }
1534
1535   M2g_MP3R_EV[[i2]] <- as.data.frame(M2g_MP3R)
1536   M2g_FinNamesEV[[i2]] <- as.vector(na.omit(M2g_MP3R_EV[[i2]]$sign_PVName))
1537
1538   #####
1539   ##### Writing csv files out #####
1540   #####
1541   #####
1542   write.csv(M2g_MP3R_EV[[i2]],
1543     paste(dir,max.M2g,
1544       "M2g_MP3R_EV",
1545       if(rv<10){},
1546       rv,"_",
1547       RVName,
1548       "_",
1549       DataName,
1550       "n",
1551       l2,
1552       "n",
1553       M2g_MP3R$DataframeName[i], # temporary see line below
1554       "n.csv",
1555       sep=""),row.names=F) #
1556
1557   } # end of l2 loop
1558
1559   NamesTmp <- as.vector("")
1560   lp<-1
1561   for (lp in seq(M2g_FinNamesEV))
1562   {
1563     if(length(M2g_FinNamesEV[[lp]]> 0) &NamesTmp <- rbind(NamesTmp,names(M2g_FinNamesEV[lp])) # picking only
1564       those that have data ( > 0)
1565   }
1566   EVNamesBest <- NamesTmp[-1] #
1567
1568   M2g_FinNamesList <- as.list(M2g_FinNamesEV[EVNamesBest]) # pick out only those that match by name (and bring in
1569     whatever data inside)
1570   M2g_MP3R_EV_Spl[[spl]] <- M2g_MP3R_EV
1571   M2g_FinNamesList_Spl[[spl]] <- M2g_FinNamesList
1572   # only interested in EV picked
1573
1574   if(length(M2g_FinNamesList_Spl[[spl]]> 0){M2g_FinNamesVect_Spl[[spl]] <-as.vector(unlist(M2g_FinNamesList))
1575     write.csv(M2g_FinNamesVect_Spl[[spl]],
1576       paste(dir,max.M2g_All,
1577         "M2g_FinNamesVect_Spl",_,
1578         DataName,"_",
1579         #_AltCode "RV", if(rv<10){0}, rv,"_",
1580         RVName,
1581         "n.csv",
1582         sep=""),row.names=F)} # only do this if there is something stored
1583
1584   write.csv(M2g_MP3R_EV_Spl[[spl]], # extra prob
1585     paste(dir,max.M2g_All,
1586       "M2g_MP3R_Spl",_,
1587       DataName,"_",
1588       #_AltCode "RV", if(rv<10){0}, rv,"_",
1589       RVName,
1590       "n.csv",
1591       sep=""),row.names=F) #

```

-23-

-24-

```
1592 } # end of spl loop
1593
1594 M2g_MP3R_EV_Spl_RV[rv] <- M2g_MP3R_EV_Spl
1595 M2g_FinNamesList_Spl_RV[rv] <- M2g_FinNamesList_Spl # assign the names ... decide if FinNames just RVName or a
combination of RV and what the list is
1596 M2g_FinNamesVect_Spl_RV[rv] <- M2g_FinNamesVect_Spl
1597 #####
1598 ##### Cleaning -- removing unnecessary bits #####
1599 rm(M2g_MP3R_EV_M2g_FinNamesList_DataName, M2g_MP3R) #
1600 rm(spl_OutNames, l12, lp, RVXY, MaxRunBat_Prespos, PosRV, PosUB)
1601 rm(M2g_DataList.SPADatName.SPADatName.EVNamesBest)
1602 rm(SPA.df.Loopi.LoopiNames.LoopiPVName.Loop2.Loop2Names)
1603 rm(Max.SampleName.Max.Sample)
1604 rm(Max.SwD.Name.Max.SwD.MaxOut.MaxRes.MaxResAUC.MaxResE.MaxResFile.MaxResN.MaxResT)
1605 rm(FTest.P.Value.NamesTmp)
1606 rm(M2g_MP3R_EV_Spl.M2g_FinNamesVect_Spl.M2g_FinNamesList_Spl)
1607
1608 if(SaveTempFlag==TRUE){
1609   print("This should only print if SaveTempFlag is True")
1610   save.image(paste(dir.rdata,
1611                 if(DAC>0){DAC},
1612                 ScriptNameCode, "_",
1613                 "Spl_AllDoneTemp",
1614                 " ", "RV", " ",
1615                 if(rv<10){0}, rv,
1616                 ".RData", sep=""))
1617   } # SaveTmpFlag Condition
1618
1619 } # end of rv loop
1620 names(M2g_FinNamesList_Spl_RV) <- RVNames
1621 names(M2g_FinNamesVect_Spl_RV) <- RVNames
1622 names(M2g_MP3R_EV_Spl_RV) <- RVNames
1623
1624 M2g_Obj<- objects(pattern="M2g")
1625 save(list=M2g_Obj, ## All Starting Sets of DV for Each of the EV
1626      file=paste(dir.rdata,
1627                 # OptCode   if(DAC>0){DAC},
1628                 "Obj", " ",
1629                 ScriptNameCode,
1630                 "_AllObj", " # object name
1631                 ".RData", sep="")) #
1632
1633 save.image(paste(dir.rdata,
1634                  if(DAC>0){DAC},
1635                  ScriptNameCode, "_",
1636                  "End_AllObj",
1637                  # _OptCode _Spl All ",
1638                  # _OptCode "RV All",
1639                  ".RData", sep=""))
1640
1641 #####
1642 ##### Next Script #####
1643 #####
1644 #####
1645 #####
1646
1647 #####
1648 #####
1649 #####
1650 #####
1651 ##### MIA2 Module 2 - Preparing Derived Variables #####
1652 ##### Grouping back all Transformations by "EV" #####
1653 ##### Using M2_Core and Spline Lists
1654 #####
1655 ##### M2h_FullSet_DVs.R #####
1656 ##### _Sep 2013 #####
1657 ##### sabgs@yahoo.com #####
1658 ##### #####
1659 ##### #####
```

```
1660 #####
1661 #####
1662 ScriptNameCode <- "M2h" #a "short bit of ScriptName
1663 ##### load up objects if not present!
1664 # _AltCode load("D:/IDA/A06_InEast/RData/A06_M2h_End_AllCollated.RData")
1665 #####
1666 #####
1667 ##### New List structures #####
1668 ##### This changes the list grouping from the existing structure of
1669 ##### into Lists of Derived Variables with dataframes containing the values for all EV's
1670 ##### Lists of Types of EV's grouping all types of DV, where the dataframe will contain all types of DVs for each variables
1671 ##### Thus need to make the names of the EVs as the top level rather the the "type of data for all EV"
1672 ##### NOTE PV is equivalent to DV, but still kept for backwards compatibility of tools
1673 #####
1674
1675 g<0
1676
1677 EVPVFBin_RV <- list0 # all these start w EVPV to indicate the grouping by EV and DV (Derived variable)
1678 EVPVLin_RV <- list0
1679 EVPVZSk_RV <- list0
1680 EVPVLinZSk_RV <- list0
1681 EVPVDev_RV <- list0
1682 EVPVLinZSkDev_RV <- list0
1683 EVPVLinZSkDevSpl_RV <- list0
1684 EVPVSpHfK_RV <- list0 # not all these are mandatory ... user decides
1685 EVPVSpHfK_RV <- list0
1686 EVPVSpThk_RV <- list0
1687
1688 M2h_FullSet_Data_RV <-list0
1689
1690 rv<-1 # decide where to start
1691
1692 for (rv in seq(length(RVNames))) ##
1693 {
1694   M2_Core <- M2_CoreLists_RV[rv]
1695   RVName <- paste("RV", if(rv<10){0}, rv, sep="")
1696   print(paste("Response Variable n.", if(rv<10){0}, rv, RVName), sep="")
1697   EVPVFBin_RV[rv] <- M2_Core$FBin ##### Binary categorical data # it is already in a 2 level format # _InCode
object.size(M2_Core$FBin)
1698   EVPVLin <- as.list(1:length(M2_Core$Lin)) ##### Linear (Ranged) variable data
1699   names(EVPVLin) <- names(M2_Core$Lin)
1700   EVPVZSk <- as.list(1:length(M2_Core$ZSk)) ##### Zero Skewness
1701   names(EVPVZSk) <- names(M2_Core$ZSk)
1702   EVPVLinZSk <- as.list(1:length(M2_Core$Lin)) ## Having the Linear and Zero Skewness Variables together
1703   names(EVPVLinZSk) <- names(M2_Core$Lin)
1704
1705   ## Matching the Derived Variables for each EV list by Dataframe
1706   lp<-1
1707   for (lp in seq(EVPVLin))
1708   {
1709     EVPVLin[lp] <- M2_Core$Lin[lp] # a list of dataframes
1710     names(EVPVLin[lp]) <- paste(names(EVPVLin[lp]), "Lin", sep="")
1711     EVPVZSk[lp] <- M2_Core$ZSk[lp]
1712     names(EVPVZSk[lp]) <- paste(names(EVPVZSk[lp]), "ZSk", sep="")
1713     EVPVLinZSk[lp] <- cbind(EVPVLin[lp], EVPVZSk[lp]) # as a list of 2 dataframes
1714     # _AltCode if you want a list of lists: EVPVLin[lp] <- list(M2_Core$Lin[lp])
1715   }
1716
1717   # _InCode str(EVPVLin[[lp]]) # _InCode str(EVPVZSk[[lp]]) # _InCode str(EVPVLinZSk[[lp]])
1718
1719   #####
1720   ##### Adding the Deviation Variables Separately #####
1721   ##### Not all Continous EVs have Deviation Variables
1722   #####
1723   #####
1724   #####
```

```
1723  EVPVLinZskDev <- EVPVLinZsk # quick and dirty
1724  # _InsCode str(EVPVLinZskDev)
1725
1726  EVTag <-names(M2_Core$Dev) #
1727
1728  lp <-1
1729  for (lp in seq(EVTag))
1730  {
1731    EVPVDev <- M2_Core$Dev # just to simplify addresses
1732    EVPTag <- as.vector(names(EVPVDev[[lp]]))
1733    EVPVCol <- as.data.frame(EVPVDev[[lp]])
1734    LinZsk <- as.data.frame(EVPVLinZsk[[EVTag[[lp]]])
1735    names(LinZsk) <- names(EVPVLinZsk[[EVTag[[lp]]])
1736    EVPVLinZskDev[[EVTag[[lp]]] <- list(cbind(LinZsk,EVPVCol)) # adding just where there are dev Variables
1737    # _InsCode names(EVPVCol)
1738    # _InsCode names(EVPVLinZskDev[[EVTag[[lp]]])
1739    # _InsCode str((EVPVLinZskDev[[EVTag[[lp]]])
1740  } # end of lp in EVTag
1741
1742  # _InsCode str(EVPVLinZskDev)
1743  # _InsCode names(EVPVLinZskDev[[1]])
1744  # _InsCode names(EVPVLinZskDev[[EVTag[[1]]])
1745
1746  ### Assign all to the right RV list position
1747
1748  EVPVLin_RV[[rv]] <- EVPVLin #### Linear (Ranged) variable data
1749  EVPVZsk_RV[[rv]] <- EVPVZsk #### Zero Skewness
1750  EVPVLinZsk_RV[[rv]] <- EVPVLinZsk #### Having the Linear and Zero Skewness Variables together
1751  EVPVLinZskDev_RV[[rv]] <- EVPVLinZskDev #### adding the Deviation Variables
1752
1753  names(EVPVLin_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="") # and Add EVPVLin if wanted or RVNames
1754  names(EVPVZsk_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="") # or RVNames
1755  names(EVPVZsk_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="") # or RVNames
1756
1757  rm(EVPVLin,EVPVZsk,EVPVLinZsk) #### clean up some things ####
1758
1759  if(length(EVTag)>0){EVPVDev_RV[[rv]] <- EVPVDev
1760    rm(EVPVCol,EVPVDev,EVTag,EVPVTag,LinZsk,EVPVLinZskDev) # only remove if something was done
1761
1762  } # end of rv loop for m2h
1763
1764  # _InsCode str(EVPVLinZskDev_RV)
1765
1766  #####
1767  #####
1768  #####
1769  ##### Picking the best of the spline DV's from M2g to add #####
1770  #####
1771  #####
1772  #####
1773  str(M2g_FinNamesList_Spl_RV[[rv]])
1774
1775  str(M2g_FinNamesList_Spl_RV)
1776  str(M2f_SplineList_RV[[rv]])
1777
1778  rv<-1 # or start where you want
1779
1780  for (rv in seq(length(RVNames))) #
1781  {
1782    SplineList <- M2f_SplineList_RV[[rv]] #
1783    EVPVLinZskDevSpl <- EVPVLinZskDev_RV[[rv]] # not interested in separating Spl for now .. take them all together
1784    M2g_FinNamesList_Spl <- M2g_FinNamesList_Spl_RV[[rv]] # peeling the first layer ... spline
1785    spl<-1
1786    for (spl in seq(length(M2g_FinNamesList_Spl)))
1787    {
1788      EVSplBest <- M2g_FinNamesList_Spl_RV[[rv]][spl]# not interested in separating Spl for now .. take them all together
```

```
1789  EVTag <- names(EVSplBest) # the first
1790  EVPVSpl <- EVSplBest # quick way to create the object; this will hold the actual data for the best spline set
1791
1792  for (lp in seq(EVSplBest)) # first get the EVPV for just the M2h
1793  {
1794    EVPTag <- as.vector(EVSplBest[[lp]])
1795    EVPVSpl[[EVTag[[lp]]] <- list(SplineList[[spl]][[EVTag[[lp]]][[EVPTag]] # Three levels of tags #
1796    # _InsCode names(EVPVSpl[[EVTag[[lp]]])
1797    # _InsCode names(EVPVSpl)
1798  }
1799
1800  if(spl == 1) {EVPVSplHfK <- EVPV$pl}
1801  if(spl == 2) {EVPVSplHRk <- EVPV$pl}
1802  if(spl == 3) {EVPV$plThk <- EVPV$pl}
1803
1804  #####
1805  for (lp in seq(EVTag)) # go through all the EV and attach them
1806  {
1807    # _OptCode print("Only if EVTag exist")
1808    EVPVLinZskDevSpl[[EVTag[[lp]]] <- cbind(EVPVLinZskDevSpl[[EVTag[[lp]]],EVPV$pl[[EVTag[[lp]]]) # adding just where
      there are spline DVs
1809    # _InsCode names(EVPVLinZskDevSpl[[1]])
1810  } # end of EVTag
1811
1812  } # end of spl loop
1813
1814  EVPV$plHfK_RV [[rv]] <- EVPV$plHfK
1815  EVPV$plHRk_RV [[rv]] <- EVPV$plHRk
1816  EVPV$plThk_RV [[rv]] <- EVPV$plThk
1817  EVPVLinZskDevSpl_RV[[rv]] <- EVPVLinZskDevSpl #
1818  # _InsCode str(EVPVLinZskDevSpl_RV)
1819  names(EVPVLinZskDevSpl_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="") #
1820  # _InsCode str(EVPVLinZskDevSpl_RV)
1821  # _InsCode names(EVPVLinZskDevSpl_RV[[1]][5])
1822
1823  names(EVPV$plHfK_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="")
1824  names(EVPV$plHRk_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="")
1825  names(EVPV$plThk_RV)[rv] <- paste("RV",if(rv<10){0},rv,sep="")
1826
1827  #rm(EVNamesBest)
1828  rm(EVPV$pl,EVPV$plHfK,EVPV$plHRk,EVPV$plThk)
1829  rm(EVPVLinZskDevSpl)
1830  rm(EVTag,EVPVTag,EVSplBest)
1831  rm(lp,spl)
1832  } # end of rv in the Spline section
1833
1834  str(EVPVLinZskDevSpl_RV[[rv]])
1835
1836  #####
1837  ##### Add the FBin -- Categorical Data back in
1838
1839  # _InsCode MIAPar$EVCatNames
1840  # _InsCode EVCatNum
1841
1842  for (rv in seq(length(RVNames)))
1843  {
1844    if(length(EVAlinNames)> EVCntNum){M2h_FullSet_Data_RV[rv] <- list(c(EVPVLinZskDevSpl_RV[[rv]], EVPV$Bin_RV[[rv]]))
1845    if(length(EVAlinNames)== EVCntNum){M2h_FullSet_Data_RV[rv] <- list(EVPVLinZskDevSpl_RV[[rv]])
1846    names(M2h_FullSet_Data_RV)[rv]]] <- paste("M2h_FullSet_Data_RV",if(rv<10){0},rv,sep="")
1847  } # end of rv loop
1848
1849  # _InsCode str(M2h_FullSet_Data_RV)
1850  # _InsCode str(M2h_FullSet_Data_RV[[rv]])
1851
1852  #####
1853  #####
1854  ##### Inspection scripts outside the rv loop ###
1855  #####
```

```
1856 #####
1857 # InsCode str(M2h_FulSet_Data_RV[[rv]])
1858 print(length(M2h_FulSet_Data_RV[[rv]]))
1859 length(M2h_FulSet_Data_RV[[rv]])
1860 str(M2h_FulSet_Data_RV)
1861 length(M2h_FulSet_Data_RV[[rv]])
1862 rv
1863 head(M2h_FulSet_Data_RV[[rv]][1:10]) # rv
1864 head(M2h_FulSet_Data_RV[[rv]][1:10]) # RV2
1865 head(M2h_FulSet_Data_RV[[rv]][1:10]) # RV1 - ev=2
1866 head(M2h_FulSet_Data_RV[[rv]][1:length(M2h_FulSet_Data_RV[[rv]])]) # RV1 - ev=last
1867 head(M2h_FulSet_Data_RV[[rv]][1:length(M2h_FulSet_Data_RV[[rv]])]) # RV1 - ev=last
1868
1869 #####
1870 #####
1871 #####
1872 #####
1873 ##### Printing out some .csv files if needed and
1874 ##### Extracting just the names of the final sets ####
1875 ##### of DV's per each EV and respective RV
1876 #####
1877 #####
1878 #####
1879 dir.max.M2h <- paste(dir.max, "M2h_FulSetDV\\", sep="")
1880 dir.create(dir.max.M2h)
1881
1882 M2h_FulSet_NamesRV <- list()
1883
1884 rv<-1
1885 for(rv in seq(M2h_FulSet_Data_RV))
1886 {
1887   dir.max.M2h.rv <- paste(dir.max.M2h, "M2h_FulSetDV_", sep="")
1888   dir.create(dir.max.M2h.rv)
1889   cat("\nWriting out RV ", if(rv<10){}, rv, " starting sets of DV", sep="")
1890
1891   M2h_FulSet_NamesEV <- list()
1892   for (ev in seq(M2h_FulSet_Data_RV[[rv]]))
1893   {
1894     write.csv(M2h_FulSet_Data_RV[[rv]][ev],
1895               paste(dir.max.M2h.rv, if(rv<10){}, rv, "M2h_FulSetDV_",
1896                   # RVNames[rv], " ", names(M2h_FulSet_Data_RV[[rv]][ev]), # full name or abbreviation
1897                   "RV", if(rv<10){}, rv, "_", "EV", ev,
1898                   ".csv", sep=""), row.names=FALSE)
1899     M2h_FulSet_NamesEV[[ev]] <- names(M2h_FulSet_Data_RV[[rv]][ev])
1900   } # end of ev loop
1901
1902   write.csv(M2h_FulSet_Data_RV[[rv]],
1903             paste(dir.max.M2h.rv, if(rv<10){}, rv, "M2h_FulSetDV_",
1904                 # OptCode RVNames[rv], "_",
1905                 ".csv", sep=""), row.names=FALSE)
1906   "RV", if(rv<10){}, rv, "_", "AllEVs",
1907   ".csv", sep=""), row.names=FALSE)
1908
1909 M2h_FulSet_NamesRV[[rv]] <- unlist(M2h_FulSet_NamesEV)
1910 names(M2h_FulSet_NamesRV)[rv][1] <- paste("M2h_FulSet_Names_RV", if(rv<10){}, rv, sep="")
1911 write.csv(M2h_FulSet_NamesRV[rv],
1912           paste(dir.max.M2h.rv, if(rv<10){}, rv, "M2h_All_Names_FinDVSet",
1913               # OptCode RVNames[rv], "_",
1914               ".csv", sep=""), row.names=FALSE)
1915   "RV", if(rv<10){}, rv, #
1916   ".csv", sep=""), row.names=FALSE)
1917 write.csv(M2h_FulSet_NamesRV[rv],
1918           paste(dir.max.M2h, "M2h_All_Names_FinDVSet",
1919               # OptCode RVNames[rv], "_", names(M2h_FulSet_Data_RV[[rv]][ev]), # full name or abbreviation
1920               "RV", if(rv<10){}, rv, #
1921               ".csv", sep=""), row.names=FALSE)
```

```
1922 rm(M2h_FulSet_NamesEV)
1923 } # end of rv loop
1924
1925 M2h_FulSet_NamesRV # Lists names of all EV's derived variables to test, start the next process
1926
1927 str(M2h_FulSet_Data_RV[[rv]])
1928 str(M2h_FulSet_NamesRV)
1929 length(M2h_FulSet_Data_RV[[rv]])
1930
1931 #####
1932 ##### Turning M2_Core_List into a M2_N_SWD format #####
1933 ##### aka the "projection files" created later on before M3A selection #####
1934 ## Needed also for creating M2SM_N_SWD and M3SM_N_SWD
1935 ## Option 1 Different Backmd Ptns and Different N for each RV
1936 #####
1937 #####
1938 #####
1939 M2_N_SWD_RV <- as.list(1:length(M2h_FulSet_Data_RV)) # define so you can assign names right away
1940 names(M2_N_SWD_RV) <- paste("M2_N_SWD_RV", 1:length(M2h_FulSet_Data_RV), sep="")
1941
1942 rv<-1
1943 for (rv in seq(length(M1_N_SWD_RV)))
1944 {
1945   M2_Core <- M2_CoreLists_RV[[rv]]
1946   str(M2_Core$RV)
1947   RVName <- paste("RV", if(rv<10){}, rv, sep="")
1948   print(paste(Response.Variable.n.", if(rv<10){}, rv, RVName), sep="")
1949   RVXY <- M1_N_SWD_RV [[rv]][1:i]
1950   ev<-1 # just for testing purposes
1951   M2_N_SWD_EV <- as.list(1:length(M2h_FulSet_Data_RV[[rv]])) # define so you can assign names right away
1952   names(M2_N_SWD_EV) <- names(M2h_FulSet_Data_RV[[rv]])
1953   for (ev in seq(length(M2h_FulSet_Data_RV[[rv]])))
1954   {
1955     M2_N_SWD_EV[[ev]] <- cbind(RVXY,
1956                                M2h_FulSet_Data_RV[[rv]][[ev]])
1957   }
1958   names(M2_N_SWD_EV)
1959   str(M2_N_SWD_EV[[ev]])
1960   count(M2_N_SWD_EV[[1:11]])
1961   str(M2_N_SWD_EV)
1962   M2_N_SWD_RV [[rv]] <- M2_N_SWD_EV
1963   rm(M2_Core.M2_N_SWD_EV.RVXY)
1964   } # end of RV Loop
1965
1966 rv<-1
1967 ev<-26
1968 str(M2_N_SWD_RV)
1969 str(M2_N_SWD_RV[[rv]])
1970 str(M2_N_SWD_RV[[rv]][[ev]]))
1971 names(M2_N_SWD_RV)
1972 names(M2_N_SWD_RV[[rv]])
1973 names(M2_N_SWD_RV[[rv]][[ev]])
1974 names(M2_N_SWD_RV[[rv]][[ev]])
1975 str(M2_N_SWD_RV[[rv]][[ev]])
1976 (M2_N_SWD_RV[[rv]][[1:4,1000:10004]])
1977 (M2_N_SWD_RV[[rv]][[1:4,1000:10004]])
1978 (M2_N_SWD_RV[[rv]][[1:4,1000:10004]])
1979 (M2_N_SWD_RV[[rv]][[1:4,1000:10004]])
1980
1981
1982
1983
1984 rv<-1
1985
1986 M2_Obj<- objects(pattern="M2") # Consider saving these together across process
1987 SWD_Obj<- objects(pattern="SWD") # Consider saving these together
1988
1989 save(list=M2_Obj, ## All Starting Sets of DV for Each of the EV
1990      file=paste(dir.rdata,
```

```

1992 #_OptCode if(DAC>0){DAC},
1993 "Obj_",
1994 ScriptNameCode,
1995 " _All", # object name
1996 ".RData".sep="") #
1997
1998 #####
1999 #####
2000 ##### Group all your results together ####
2001 ##### Some of these objects are only used this module and user can decide if to keep or remove in later modules
2002
2003
2004 EVPVObj<- objects(pattern="EVPV") # these can be saved for later use, but will not be needed for M3 right now
2005
2006 save(list=EVPVObj, ## All Starting Sets of DV for Each of the EV
2007 file=paste(dir.rdata,
2008 #_OptCode if(DAC>0){DAC},
2009 "Obj_",
2010 ScriptNameCode,
2011 "EVPVObj", # object name
2012 ".RData".sep="")) #
2013
2014
2015 M2h_Obj<- objects(pattern="M2h")
2016 save(list=M2h_Obj, ## All Starting Sets of DV for Each of the EV
2017 file=paste(dir.rdata,
2018 #_OptCode if(DAC>0){DAC},
2019 "Obj_",
2020 ScriptNameCode,
2021 "AllLists", # object name
2022 #_OptCode "FullSets_All",
2023 ".RData".sep="")) #
2024
2025
2026 save.image(paste(dir.rdata,
2027 if(DAC>0){DAC},
2028 ScriptNameCode,
2029 #_OptCode "Image_All",
2030 "End_All",
2031 #_OptCode "Collated",
2032 ".RData".sep=""))
2033
2034 ##### Final clean up and save #####
2035
2036 ##### Now that they are all "grouped together" we remove what is not used in the next set of modules
2037 ##### easier to find and list and save them this way
2038
2039 ## lots of duplication right now. with EVDV and other lists, as a result of ongoing development. ok to keep or delete as needed
2040 rm(EVPVObj,list=EVPVObj)
2041
2042 rm(Bkgnd_Difrm.Bkgnd_SVD.Bkgnd_SP.ev.stack) # not needed for now as modules stand right now
2043 rm(ras.rvt.obj, fil.fct.pos.ev.col.DimStack, InType.n.datasets) # rm little bits left behind
2044 rm(num.dups.ev, num.dups.noties, orig.dups.ev)
2045
2046 rm(list=objects(pattern="M2f")) # Remove the full splines and even the M2g objects are not needed at this point and they
2047 tend to be very large
2048 rm(list=objects(pattern="M2g")) # remove M2g objects are not needed at this point
2049
2050
2051 save.image(paste(dir.rdata,
2052 if(DAC>0){DAC},
2053 ScriptNameCode,
2054 "_End_",
2055 "BareMin",
2056 ".RData".sep=""))
2057
2058
2059 ##### Next Script #####
2060

```



```
1 2 #####
3 3 #####
4 4 #####
5 5 ##### MIAT Module 2 - Preparing Derived Predictors #####
6 6 ##### Visualising your Results #####
7 7 ##### FREQUENCY OF PRESENCE PLOTS #####
8 8 #####
9 9 ##### M2e FOPPlot.R #####
10 10 ##### 2015 #####
11 11 ##### Author: Sabrina Mazzoni #####
12 12 ##### ##### ##### ##### ##### ##### #####
13 13 ##### ##### ##### ##### ##### ##### #####
14 14 ##### Review your data to #####
15 15 ##### Decide which data to plot #####
16 16 ##### ##### ##### ##### ##### ##### #####
17 17 names(M2_CoreLists_RV[[rv]]$QntCIsMid[ev])
18 18 dim(M2_CoreLists_RV[[rv]]$RV)
19 19 head(M2_CoreLists_RV[[rv]]$RV)
20 20 str(M2_CoreLists_RV)
21 21 names(M2_CoreLists_RV)
22 22 names(M2_CoreLists_RV[[rv]])
23 23 str(M2_CoreLists_RV[[rv]]$RV)
24 24
25 25 min(M2_CoreLists_RV[[rv]]$QntCIsMid[ev])
26 26 min(M2_CoreLists_RV[[rv]]$QntCIsFr[ev])
27 27 M2_CoreLists_RV[[rv]]$QntCIsFr[ev]
28 28 M2_CoreLists_RV[[rv]]$QntCIsTo[20,ev]
29 29 M2_CoreLists_RV[[rv]]$QntCIs[20,ev]
30 30 str(M2_CoreLists_RV[[rv]]$QntCIs)
31 31 tail(M2_CoreLists_RV[[rv]]$QntCIs)
32 32
33 33 rv<-1
34 34
35 35 #####
36 36 #####
37 37 ##
38 38 ### FREQUENCY OF PRESENCE PLOTS ###
39 39 #### Two sections
40 40
41 41 ##### Part 1: the Original (no ties) results ###
42 42 ##### DataName = RVFrq_Orig
43 43 #####
44 44 #####
45 45
46 46 ##### Define our x- and y-limit
47 47 XlimFlag <-FALSE # TRUE
48 48 YlimFlag <-FALSE #TRUE
49 49 XlimSet <- 1
50 50 YlimSet <- .04 # can be assigned by hand to be same for all, or dataspecific if Flag is False
51 51
52 52 EVNames <- names(M2_CoreLists_RV[[rv]]$Lin) # just the linear
53 53 EVLabels <- EVNames
54 54
55 55 rv<-1
56 56 for (rv in seq(length(M2_CoreLists_RV)))
57 57 {
58 58   RVName<- Train.Data[[rv]][1,1] # just to be sure
59 59   ev <-1
60 60   for (ev in seq(length(M2_CoreLists_RV[[rv]]$RVFrqNTys)))
61 61   {
62 62     X <- M2_CoreLists_RV[[rv]]$QntCIsMid # the midvalues of each EV quantile class
63 63     Y <- M2_CoreLists_RV[[rv]]$RVFrqNTys # The Frequency of presence of the RV for EV's quantile class*Original, no
64 64     # ties", this changes to RVFrqLast in next bit
65 65     YOpt<- M2_CoreLists_RV[[rv]]$RVOptNTys # the reputed species optima for last "orinal no ties" values ... this changes to
66 66     # RVOpt5mthLast (or RVOpt5mth3) in next bit
67 67     # 1 is the actual value of the highest frequency of presence
68 68     # 2 is the quantile class where this occurs
69 69     # 3 is the value from ev.qncis.mid for that quant-pos
70 70     # head(YOpt)
71 71     # names(M2_CoreLists_RV[[rv]][1:15])
```

```
69 # setting this by hand for now, but can go back to "if statement with names like in previous version)
70 DataName <- paste("RVFrq_Orig", sep="") # The original values after removing the ties
71 #str(X)
72 Xlim <- max(X) # set it as dataspecific, unless otherwise flagged above
73 Ylim <- max(Y) # set it as dataspecific, unless otherwise flagged above
74 if (XlimFlag==TRUE) { Xlim <- XlimSet} # set it all the same for all, otherwise column specific
75 if (YlimFlag==TRUE) { Ylim <- YlimSet} # set it all the same for all, otherwise column specific
76
77 plot(X[ev],Y[ev],
78 xlab=EVLabels[ev],
79 ylab=paste("Frequency of Presence RV",rv, RVName),
80 main= paste("RV",rv, RVName),
81 # Dataspecific... "NoXYLim", #
82 xlim=c(0,Xlim),
83 ylim=c(0,Ylim),
84 font.main=4
85 # sub="Mid Value per Quantile Class in red")
86 )
87
88 # length(M2_CoreLists_RV[[rv]]$QntCIsTo[ev]) -- or qn.meth
89
90 lines(X[ev],Y[ev],col="gray",lwd=1)# end of plot
91
92 points(YOpt[3,ev],YOpt[1,ev],col="dark red", pch=16,cex=1.4) # the Maximum Value of the FoP filled in
93 mtext(side = 2, at = YOpt[1,ev],line=1.8, # # the Maximum Value of the FoP as text, on the Y axis
94 text = paste("MaxFreq\n",round(YOpt[1,ev],digits=4),sep=""),
95 cex=.7, col = "dark red")
96
97 mtext(side = 1, at = YOpt[3,ev], line = -1, # the Mid Value of the Maximum FoP value as text, on the X axis
98 text = paste("Sp Opt\n",round(YOpt[3,ev],digits=3),sep=""), # A label for it, to indicate "Species Optima" just
99 # below the X axis
100 cex=.7, col = "dark red")
101
102 mtext(side = 1, at = YOpt[3,ev] , line = -.3,
103 text = "|", cex=.6,col = "dark red",font = 2) # A tickmark just to help visualise
104
105 dev.print(win.metafile,filename=paste(dir,plot.out,
106 # "NoXYLim",
107 "RV",rv, "_",
108 RVName,"_",
109 names(Y[ev]), "_",
110 DataName,
111 # "obj", obj,pageNum, "of", nPages,
112 ".wmf", sep=""), res=300)
113
114 graphics.off()
115 } # end of ev
116 } # end of rv
117
118 #####
119 #####
120 ### FREQUENCY OF PRESENCE PLOTS ###
121 ### Two sections
122 ### Part 2: the smoothed results
123 ### DataName = RVFrqSmooth
124 #####
125 #####
126 #####
127
128 ### Define our x- and y-limit
129 XlimFlag <-FALSE # TRUE
130 YlimFlag <-FALSE #TRUE
131 XlimSet <- 1
132 YlimSet <- .04 # can be assigned by hand to be same for all, or dataspecific if Flag is False
133
134 EVNames <- names(M2_CoreLists_RV[[rv]]$Lin) # just the linear
135 EVLabels <- EVNames
136
137
```

```
138 num.smoothing<-MIAPar$num.smoothing
139 EVLabels <- EVNames
140
141 rv<-1
142 for (rv in seq(length(M2_CoreLists_RV)))
143 {
144   RVName<- Train.Data[[rv][1,1] # just to be sure
145   ev <-1
146   for (ev in seq(length(M2_CoreLists_RV[[rv]]$RVFrqSmthLast))) # or RVFrqSmthAll[[num.smoothing]] of your choice
147   {
148     X <- M2_CoreLists_RV[[rv]]$QntClsMid ## the midvalues of each EV quantile class
149     Y <- M2_CoreLists_RV[[rv]]$RVFrqSmthLast # The Frequency of presence of the RV for EV's quantile class"Original. no
      ties", this changes to RVFrqLast in next bit
150     YOpt<- M2_CoreLists_RV[[rv]]$RVOptSmthLast # the reputed species optima for last "original no ties" values ... this
      changes to RVOptSmthLast (or RVOptSmth3) in next bit
151     # 1 is the actual value of the highest frequency of presence
152     # 2 is the quantile class where this occurs
153     # 3 is the value from ev.qncls.mid for that quant-pos
154     # InsCode head(YOpt)
155     DataName <- paste(RVFrq$smooth",num.smoothing, sep="") # The values after the $smoothing has occurred
156
157     plot(X[ev],Y[ev],
158          xlab=EVLabels[ev],
159          ylab=paste("Frequency of Presence RV",rv, RVName),
160          main= paste("RV",rv, RVName),
161          #_AltCode xlim=c(0.1), #ylim=c(0.1),
162          xlim=c(0,XLim),
163          ylim=c(0,YLim),
164          font.main=4
165          #_OptCode sub="Mid Value per Quantile Class in red"
166          )
167     lines(X[ev],Y[ev],col="gray",lwd=1)# end of plot
168     points(YOpt[3,ev],YOpt[1,ev],col="dark red",pch=16,cex=1.4) # the Maximum Value of the FoP filled in
169     mtext(side = 2, at = YOpt[1,ev],line=-1.8, # # the Maximum Value of the FoP as text, on the Y axis
170           text = paste("MaxFreq",round(YOpt[ev,digits=4],sep=""),
171             cex=.7, col = "dark red")
172
173     mtext(side = 1, at = YOpt[3,ev], line = -1, # the Mid Value of the Maximum FoP value as text, on the X axis
174           text = paste("5p Opt",round(YOpt[3,ev,digits=3],sep=""), # A label for it, to indicate "Species Optima" just
175             below the X axis
176             cex=.7, col = "dark red")
177
178     mtext(side = 1, at = YOpt[3,ev], line = -.3,
179           text = "|", cex=.6,col = "dark red",font = 2) # A tickmark just to help visualise
180
181     dev.print(win.metafile, filename=paste(dir,plot.out,
182                                           #
183                                           "RV",rv,"_",
184                                           RVName,"_",
185                                           names(Y[ev]),",",
186                                           DataName,
187                                           #"obj", obj,PageNum, "of", NPages,
188                                           ".wmf", sep=""), res=300)
189
190     graphics.off()
191   } # end of ev
192 }
```



```

1 #####
2 #####
3 #####
4 ##### for flexible implementation of the maximum likelihood explanation for MaxEnt
5 #####
6 #####
7 #####
8 #####
9 #####
10 ##### MIA7 Module 3 - MaxEnt Modelling Modules #####
11 ##### Module 3a: Parsimonious Set of Derived variables for each EV
12 #####
13 #####
14 ##### 2015 #####
15 ##### sabgis@yahoo.com #####
16 #####
17 #####
18 #####
19 #####
20 #####
21 #####
22 ##### NOTE on terms and code abbreviations #####
23 ##### Some of the code abbreviations refer to old terminology, however,
24 ##### the descriptive comments use, as much as possible, the latest terms.
25 ##### Thus, abbreviations like P, stood for "predictor", but are now simply referred to as
26 ##### "variable", "derived variable" or "significant variable"
27 #####
28 # OptCode require(plyr);MIAPar$dir: Rlib) # used for joining data
29 ScriptNameCode <- "M3A"
30
31 memory.limit(8000000)
32 gc()
33
34 SaveTempFlag <- FALSE #TRUE #FALSE
35
36 ## As in this section we are running maxent.jar, place this check and instruct use
37 jarFileFlag <- paste(dir.in, "maxent.jar", sep="")
38 if (file.exists(jarFileFlag) == FALSE){
39   cat("!!!! WARNING !!!! \nPlease place a copy of maxent.jar in your dir.in directory
40   !!!! WARNING !!!!") else {All's good, carry on, carry on}
41
42 ## Remind users of their maxent.jar settings chosen before
43 JarPar$M3A_JarFlags
44
45 #####
46 #####
47 ##### Select Background Pts for maxent.jar to use #####
48 #####
49 #####
50 str(M2h_FullSet_Data_RV[[rv]])
51 length(M2h_FullSet_Data_RV[[rv]])
52
53 #####
54 #####
55 ##### Option 1 Different Background Points and Different N for each RV #####
56 ##### Other options explored elsewhere
57 #####
58 #####
59 #####
60 M3_N_SWD_RV <-list()
61 M3SM_N_SWD_RV <-list()
62 names(M3SM_N_SWD_RV)
63
64 str(M1_N_SWD_RV[[rv]]) # the order between M1 and M2 should not change
65 str(M2_N_SWD_RV[[rv]]) # the order between M1 and M2 should not change
66
67
68 PosRV<-list() # Willl hold the position of the RV and UB that get picked, so you can "project later", but maybe not needed

```

```

69 PosUB<-list()
70
71 rv<-1
72 MxBkgNum <-MIAPar$MxBkgNum
73 for (rv in seq(length(M2_N_SWD_RV)))
74 {
75   RVName <- paste("RV",
76     if(rv<10){0},
77     rv,sep="")
78   print(paste("Response Variable n:",rv, RVName),sep="")
79   ev<-1 # technically all RVXY same for all EV
80   PosRV <- which(M2_N_SWD_RV[[rv]][ev])$SWD1=="UB") # Just the training (Response Variable)
81   PosUB <- which(M2_N_SWD_RV[[rv]][ev])$SWD=="UB") # Full (original) all UB (uninformed background) positions
82   if (length(PosUB)>= MIAPar$MxBkgNum){
83     PosUB <- na.omit(sample(PosUB, MIAPar$MxBkgNum))} # Row position randomly selected from full UB
84   length(PosUB)
85   ## Extract the derived variables Dataframe (EVPV) for the same positions
86   ## One loop for each EV
87
88   M3_N_SWD_RV [[rv]] <- as.list(1:length(M2_N_SWD_RV[[rv]]) # Define so you can assign names right away
89   #length(M3_N_SWD_RV [[rv]])
90
91   ev<-1 # Useful to add for testing purposes
92   M3_N_SWD_EV <- as.list(1:length(M2_N_SWD_RV[[rv]]) # Define so you can assign names right away
93   names(M3_N_SWD_EV) <- names(M2_N_SWD_RV[[rv]])
94
95   for (ev in seq(length(M2_N_SWD_RV[[rv]])))
96   {
97     M3_N_SWD_EV[[ev]] <- rbind(M2_N_SWD_RV[[rv]][ev]][PosRV.]. # The training portion (could just use this as presence
98     too)
99     M2_N_SWD_RV[[rv]][ev]][PosUB.])
100   }
101   #_InsCode names(M3_N_SWD_EV)
102   #_InsCode str(M3_N_SWD_EV[[ev]])
103   #_InsCode count(M3_N_SWD_EV[[2]][1]])
104   M3_N_SWD_RV [[rv]] <- M3_N_SWD_EV
105   rm(M3_N_SWD_EV)
106
107   ## Making SWD for "standard MaxEnt" to run with the "original raw variables"
108   ## This means that the same XY and training&background are used, but original data without transformations are extracted
109   ## Because there is 1 less level in this list, don't need to go in EV portion
110   M3SM_N_SWD_RV[[rv]] <- rbind(M1_N_SWD_RV[[rv]][PosRV.]. # the Training portion (of response-variable data)
111     M1_N_SWD_RV[[rv]][PosUB.]) # the Background portion (of response-variable data)
112   names(M3SM_N_SWD_RV)[[rv]] <- paste("M3SM_N_SWD_RV",
113     rm(PosRV.PosUB)
114     ) # end of RV Loop
115
116   #_InsCode str(M3_N_SWD_RV)
117   #_InsCode str(M3_N_SWD_RV[[rv]])
118   #_InsCode count(M3_N_SWD_RV[[rv]][ev])
119   #_InsCode str(M3_N_SWD_RV[[rv]][[ev]])
120
121   #_InsCode str(M3SM_N_SWD_RV)
122   #_InsCode str(M3SM_N_SWD_RV[[rv]])
123   #_InsCode count(M3SM_N_SWD_RV[[rv]][ev])
124   #_InsCode str(M3SM_N_SWD_RV[[rv]][[ev]])
125
126   save.image(paste(dir.rdata,
127     if(DAC>0){DAC},
128     ScriptNameCode,
129     "Start_1",
130     ".RData",sep=""))
131
132 #####
133 #####
134 ##### End of M3_N_SWD preparation ###
135 #####
136 #####
137 #####

```



```
138 AlphaValues <- MIAPar$AlphaValues
139 ReguValues <- MIAPar$ReguValues
140 rz <- 1 # ReguValues[rz] -- rz is the position holder, in this module we only test the first position (Regularization Value of 0)
141
142 dir.M3A <- paste(dir.max,
143   "M3A_All",
144   "\\",
145   sep = "")
146 dir.create(dir.M3A)
147
148 #####
149 #####
150 ##### Alpha Objects #####
151 ##### for Model Tracking Table ###
152 ##### Model Selection and Assessment ###
153 #####
154 #####
155 M3A_MP3R_Al <- as.list(f:length(AlphaValues)) # Model parameterisation Run Results/Rank -- beginning iteration table
156   -- Model tracking table
157 M3A_MP3RSEL_Al <- as.list(f:length(AlphaValues)) # Model parameterisation Run Results/Rank -- restart with selected rows
158   -- end of iteration table
159 M3A_MP3RSEL_Fin_Al <- as.list(f:length(AlphaValues)) # Final model parameter table
160
161 M3A_MP3RSEL_1st_Al <- as.list(f:length(AlphaValues)) # In case you want to trace the "best" significant variable, the 1st line,
162   for each individual EV, (and corresponding the RV, and Alpha value (Al))
163 names(M3A_MP3RSEL_1st_Al) <- paste("M3A_MP3RSEL_1st_Al",f:length(AlphaValues),sep = "")
164
165 M3A_FinPredData_List_Al <- as.list(f:length(AlphaValues)) # Will store the actual data for the variables of the final model - in
166   the form of a list
167 M3A_FinPredNames_List_Al <- as.list(f:length(AlphaValues)) # Will store just the names of the variables for the final model -
168   in the form of a list
169 M3A_FinPredData_Dfrfm_Al <- as.list(f:length(AlphaValues)) # Will store the actual data for the final model - in the form of
170   a dataframe
171 M3A_FinPredNames_Dfrfm_Al <- as.list(f:length(AlphaValues)) # Will store just the names of the variables for the final model
172   - in the form of a dataframe
173 names(M3A_FinPredData_List_Al) <- paste("M3A_FinPredData_List_Al",
174   f:length(AlphaValues),sep = "")
175 names(M3A_FinPredNames_List_Al) <- paste("M3A_FinPredNames_List_Al",f:length(AlphaValues),sep = "")
176 names(M3A_FinPredData_Dfrfm_Al) <- paste("M3A_FinPredData_Dfrfm_Al",f:length(AlphaValues),sep = "")
177 names(M3A_FinPredNames_Dfrfm_Al) <- paste("M3A_FinPredNames_Dfrfm_Al",f:length(AlphaValues),sep = "")
178
179 for (al in seq(length(AlphaValues)))
180 {
181   P_ValueCutOff <- AlphaValues[al] # alpha value ... can also be set differently to a less or more stringent test
182   cat(paste("\nChecking Alpha",al,P_ValueCutOff))
183   #####
184   #####
185   M3A_MP3R_RV <- as.list(f:length(M3_N_SWD_RV))
186   names(M3A_MP3R_RV) <- paste("M3A_MP3R_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "")
187
188   M3A_MP3RSEL_RV <- as.list(f:length(M3_N_SWD_RV))
189   names(M3A_MP3RSEL_RV) <- paste("M3A_MP3RSEL_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "")
190
191   M3A_MP3RSEL_Fin_RV <- as.list(f:length(M3_N_SWD_RV))
192   names(M3A_MP3RSEL_Fin_RV) <- paste("M3A_MP3RSEL_1st_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "")# Which Alpha
193   and RV
194
195   M3A_MP3RSEL_1st_RV <- as.list(f:length(M3_N_SWD_RV)) # In case you want to trace the "best" significant variable, the 1st
196     line, for each individual EV, (and corresponding the RV, and Alpha value (Al))
197   names(M3A_MP3RSEL_1st_RV) <- paste("M3A_MP3RSEL_1st_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "")# Which Alpha
```

```
and RV
197
198 M3A_FinPredData_List_RV <- as.list(f:length(M3_N_SWD_RV))
199 names(M3A_FinPredData_List_RV) <- paste("M3A_FinPredData_List_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "") # Which
   Alpha and RV
200
201 M3A_FinPredNames_List_RV <- as.list(f:length(M3_N_SWD_RV))
202 names(M3A_FinPredNames_List_RV) <- paste("M3A_FinPredNames_List_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "") #
   Which Alpha and RV
203
204 M3A_FinPredNames_Dfrfm_RV <- as.list(f:length(M3_N_SWD_RV)) # This only contains the EV chosen (replaces
   M3A_FinPredNames_List_RV)
205 names(M3A_FinPredNames_Dfrfm_RV) <- paste("M3A_FinPredNames_Dfrfm_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "")
206
207 M3A_FinPredData_Dfrfm_RV <- as.list(f:length(M3_N_SWD_RV)) # As dataframe duplicate of the list above but better for
   printing in column names
208 names(M3A_FinPredData_Dfrfm_RV) <- paste("M3A_FinPredData_Dfrfm_Al",al,"RV",f:length(M3_N_SWD_RV),sep = "")
209
210 #####
211 ## Start of RV loop #####
212 #####
213 rv <- 1 # Decide where to start
214
215 for (rv in seq(length(M3_N_SWD_RV)))
216 {
217   SWDList <- M3_N_SWD_RV [[rv]] # With 3 extra columns - this is the data with combined mg, zsk, dev, step 4 selected
   variables, and categorical bin data
218   RVXY <- M3_N_SWD_RV [[rv]][1:2:-1] # THIS is the RVXY for all in Module 3
219   # InsCode str(RVXY)
220   # InsCode unique(RVXY[1])
221   # AltCode RVName <- names(RVXY[1]) # original
222   RVName <- paste("RV",if(rv<10){},rv,"Al",sep = "")
223
224   cat(paste("\nModule 3a: Checking", length(SWDList), "grouped variables for rv n.",if(rv<10){},rv,RVName,"Alpha",al,
   MIAPar$P_ValueCutOff))
225
226   dir.M3A_GP <- paste(dir.M3A,
227     "M3A_Al",
228     al,
229     # AltCode "_",
230     "RV",
231     if(rv<10){},rv,
232     "\\",
233     sep = "")
234   dir.create(dir.M3A_GP)
235
236   M3A_MP3R_GPList <- list()
237   M3A_MP3RSEL_GPList <- list()
238   M3A_MP3RSEL_Fin <- list()
239   M3A_FinPredData_List <- list()
240
241   #####
242   #####
243   ##### Preparing for the MaxEnt output results and picking the best EV combination #####
244   ##### A dataframe which will store all key information from each MaxEnt run
245   ##### This will be used in many ways.
246   ## It will be saved as a list of lists of dataframes
247   ##### depending on how many levels are considered
248   #####
249   #####
250   ##### MaxResNames #####
251   MaxResNames <- as.vector(c("EV_Name",
252     "GP_Name", # GP stands for Grouped variables (grouped by EV in this case), which is synonymous with 'a
253     set of DVs representing one EV'
254     "GP_Num", # Number of GPs
255     "PV_Col", # could be 1 or more
256     "PV_NameGP", # Individual variables used, separated by space
257     "RV_Name", # The refined name which will store model parameter info (will be ModName)
258     "GP_Best",
259     "NPred_Used", # Not used in M3A but in M3B and M3C -- as it is the same as PV_Used
```

```
259 "NPred_Test", # Not used in M3A as it is always 1
260 "MaxHTMLFile",
261 "MaxPredFile",
262 "MaxLambdaFile",
263 "MaxResultsFile",
264 "MaxRes_E",
265 "MaxRes_n",
266 "MaxRes_N",
267 "MaxRes_AUC",
268 "FTVE_PV",
269 "FTVE_PrevPV",
270 "FTVE_AddPV",
271 "dtuPV",
272 "dfePV",
273 "FRatioTestPV",
274 "P_ValuePV",
275 "P_Value_BestPV",
276 "Name_BestPV",
277 "Name_AddPV",
278 "Sign_RankPV",
279 "Sign_PVNamePV",
280 "AlphaValue",
281 "Regulariz"
282
283 ))
284
285 BestGPNames<-""
286
287 #####
288 ##### Running MaxEnt and applying forward selection for individual EV's #####
289 #####
290 #####
291 #####
292
293 lp<-1 # Using lp for list position, but here it is implied to be ev
294 ev<-lp # lp for list position, or ev for environment variable. Keep both for backward compatibility with other modules
295
296 for (lp in seq(length($WDList))) # This creates the new data for step 2, which is to pick the first (best),
297 {
298   ev<-lp ## lp for list position, or ev for environment variable. Keep both for backward compatibility with other modules
299   DataGP <- list()
300   CP <- list() # CP stands for Grouped variables (single derived variables grouped by EV in this case)
301   MP3R_CP <- list()
302   MP3RSEL_CP <- list()
303
304   DataGP[[1]] <- as.list($WDList[[lp]][5:length($WDList[[lp]])])
305   length(DataGP)
306   str(DataGP[[1]]) # This is the first component of what will be as many gp iterations as needed
307   # _InsCode length(DataGP[[1]])
308   # _InsCode str(DataGP[[1]]) # This is the actual contents of the first gp component, which is a list of components itself
309
310   EVName <- names($WDList[[lp])
311   names(DataGP[[1]] <- paste("EV",if(lp<10){},lp,"_",names($WDList[[lp]), "_GP","0",1, sep="")) # this is the first round
312   # _InsCode str(DataGP)
313
314   #####
315   ##### Directories
316   #####
317   #
318
319   dir.max.ev<-paste(dir,M3A_CP,
320     "EV",
321     if(lp<10){},
322     lp,
```

```
323 "\\",
324 sep="")
325 dir.create(dir.max.ev)
326
327 #####
328 #####
329 ##### Derived Variables (DVs) grouped by each environmental variable (EV) Name ###
330 ##### Pick up from here after you've selected the best and most significant EVs, 'Grouped variables changed into 'EVs'
331 ##### gp (grouped variables) loop
332 #####
333 #####
334 #####
335 #####
336 #####
337 ##### gp loop #####
338 gp<-0
339 GPOutputExit<-FALSE
340 while (GPOutputExit==FALSE)
341 {
342   gp<-gp+1
343   RVName <- paste("RV",if(rv<10){},rv,"A",if(al=="E",if(lp<10){},lp,
344     "P",if(gp<10){},gp, sep="")
345   cat(paste("\nChecking EV",lp,names($WDList[[lp]), "with", (length($WDList[[lp]])-), "derived variables"))
346   GP[[lp]] <- as.list(1:length(DataGP[[lp]])-(gp-1)) # To be used in MaxEnt, will combine variables as specified by gp run
347   names(GP[[gp]] <- paste("EV",if(lp<10){},lp,"_",names($WDList[[lp]), "_",if(gp<10){},gp, sep="")
348   # _InsCode str(GP)
349
350   for (pv in seq(length(GP[[gp]]))) # This creates the new data for step 2, which is to pick the first (best),
351   {
352     if (gp==1) {GP[[gp]] <- DataGP[[gp]]} # This is for the loop version
353     # _InsCode str(GP)
354     if (gp>=2) {
355       GP[[gp]][[pv]] <- as.data.frame(cbind(DataGP[[gp]][BestPVNames[GP],DataGP[[gp]][pv+(gp-1)])
356         # _InsCode head(GP[[gp]] [[pv]])
357         names(GP[[gp]][pv]) <- paste("EV",if(lp<10){},lp,"_", # name of this list
358           "pv",1,"",if(pv<10){},gp,"_",
359           "pv",1,"",if(pv<10){},pv+1, sep="") # decide numbers or names
360     } # end of gp>= 2 statement
361   } # end of pv in building GP dataset
362   # _InsCode str(GP[[gp]])
363   # _InsCode str(GP)
364
365   #####
366   ##### Preparing the MaxEnt results holder #####
367   ###
368   MP3R <- as.data.frame(array(dim=c(length(GP[[gp]]),length(GP[[gp]]),length(MaxResNames)))
369   colnames(MP3R) <- MaxResNames
370
371   if (gp==1) {rownames(MP3R) <- names(DataGP[[gp]])}
372   if (gp>=2) {rownames(MP3R) <- names(DataGP[[gp]])[gp:length(DataGP[[gp]])]}
373   # _InsCode str(MP3R)
374   # _InsCode str(MP3R_CP)
375   # _InsCode str(GP[[gp]])
376   # _InsCode names(GP[[gp]])[pv]
377   # _InsCode str(GP[[gp]][[pv]][[gp]]) # THIS is the first column of the first pv combination in the gp iteration of EV!!!
378   # _InsCode names(GP[[gp]][[pv]][[1:2]])
379   # _InsCode names(GP[[gp]][[pv]][1:2])
380   # _InsCode names(GP[[gp]][[pv]])
381   GPName <- names(GP[[gp]]) # Name of list which is a list (or dataframe) of within-group variables (1 or more) that will
382     be tested
383   GPPVNames <- names(GP[[gp]]) # Name of individual variables that will be tested (gp=1, 1; gp =2, 2 etc), names of
384     the objects inside the list which are your PV (1 or more)
385   # _InsCode names(GP[[gp]])
386   # _InsCode names(GP[[gp]][pv])
```

```
335 #_InsCode names(CP[lgp][pv]) # Position pv of the list in position gp
336 #_InsCode names(CP[lgp][pv][2]) # Second position within the pv position of the list in position gp
337 #_InsCode names(CP[[gp]][pv][lgp]) # This is the third level, which for a list of 1 list will show the first record
338 #_InsCode str(CP[[gp]])
339
340
341 #####
342 #####
343 #####
344 #####
345 #####
346 #####
347
348 #_InsCode names(CP[lgp][pv]) # Position pv of the list in position gp
349 #_InsCode names(CP[lgp][pv][2]) # Second position within the pv position of the list in position gp
350 #_InsCode names(CP[[gp]][pv][lgp]) # This is the third level, which for a list of 1 list will show the first record
351 #_InsCode str(CP[[gp]])
352
353 #####
354 #####
355 #####
356 #####
357 #####
358 #####
359 #####
360 #####
361 #####
362 #####
363 #####
364 #####
365 #####
366 #####
367 #####
368 #####
369 #####
370 #####
371 #####
372 #####
373 #####
374 #####
375 #####
376 #####
377 #####
378 #####
379 #####
380 #####
381 #####
382 #####
383 #####
384 #####
385 #####
386 #####
387 #####
388 #####
389 #####
390 #####
391 #####
392 #####
393 #####
394 #####
395 #####
396 #####
397 #####
398 #####
399 #####
400 #####
401 #####
402 #####
403 #####
404 #####
405 #####
406 #####
407 #####
408 #####
409 #####
410 #####
411 #####
412 #####
413 #####
414 #####
415 #####
416 #####
417 #####
418 #####
419 #####
420 #####
421 #####
422 #####
423 #####
424 #####
425 #####
426 #####
427 #####
428 #####
429 #####
430 #####
431 #####
432 #####
433 #####
434 #####
435 #####
436 #####
437 #####
438 #####
439 #####
440 #####
441 #####
442 #####
443 #####
444 #####
445 #####
446 #####
447 #####
```

```
448 NPred_Ured <- PVUsed # Keep for flexibility
449 #####
450 #####
451 #####
452 #####
453 #####
454 #####
455 #####
456 #####
457 #####
458 #####
459 #####
460 #####
461 #####
462 #####
463 #####
464 #####
465 #####
466 #####
467 #####
468 #####
469 #####
470 #####
471 #####
472 #####
473 #####
474 #####
475 #####
476 #####
477 #####
478 #####
479 #####
480 #####
481 #####
482 #####
483 #####
484 #####
485 #####
486 #####
487 #####
488 #####
489 #####
490 #####
491 #####
492 #####
493 #####
494 #####
495 #####
496 #####
497 #####
498 #####
499 #####
500 #####
501 #####
502 #####
503 #####
504 #####
505 #####
506 #####
507 #####
508 #####
509 #####
510 #####
511 #####
512 #####
513 #####
514 #####
515 #####
```

```
516 ### Building the MaxEnt output results dataframe  ###
517 #####
518 #####
519 #####
520 MP3R$EV_Name[pv] <- EVName
521 MP3R$CP_Num[pv] <- gp # Iteration number which will tell you how many variables (number of derived variables
per EV) are picked
522 MP3R$GP_Name[pv] <- names(GP[gpID][pv])
523 MP3R$PV_Coll[pv] <- pv
524 MP3R$PV_NameGP[pv] <- PVNames
525 MP3R$PV_NameBestPV[pv] <- BestCPNames
526 MP3R$Name_AddPV[pv] <- PVText
527 MP3R$NPred_Used[pv] <- NPred_Used
528 MP3R$NPred_Test[pv] <- NPred_Test
529 MP3R$RV_Name[pv] <- RVName
530 MP3R$MaxResultsFile[pv] <- MaxResFile
531 MP3R$MaxHTMLFile[pv] <- MaxHTMLFile
532 MP3R$MaxPredFile[pv] <- MaxPredFile
533 MP3R$MaxLambdafile[pv] <- MaxLambdafile
534 MP3R$MaxResultsFile[pv] <- MaxResFile
535 MP3R$AlphaValue[pv] <- P_ValueCutOff
536 MP3R$Regulariz[pv] <- ReguValues[rz]
537 } # End of pv loop # Inner loop, by column
538
539 cat("\nDone with maxent.jar Run\n")
540 MP3R_GP[[gp]] <- MP3R
541 names(MP3R_GP)[gp] <- names(GP)[gp]
542 #_AltCode } # End of outer loop, lp, by list position if MaxEnt runs and results are run separately, see below
543
544 #####
545 #####
546 #####
547 ##### Results of analysis compiled
548 #####
549 #####
550 #####
551 #####
552 # AltCode for (lp in seq(length(EVPList))) # Loop2 ## Use this if analysis is separated from MaxEnt runs
553 #_AltCode {
554
555 MP3R <- MP3R_GP[[gp]]
556 #####
557 #####
558 ##### Loop 1 #####
559 ##### pv<-2 #####
560 # #####
561 #####
562 #####
563 #####
564 #####
565 #####
566 #####
567 #####
568 #####
569 #####
570 #####
571 #####
572 #####
573 #####
574 #####
575 #####
576 #####
577 #####
578 #####
579 #####
580 #####
```

```
581 }
582
583 if (gp>=2) {FTVE_Prev <- MP3R$Sel_GP[[gp-1]]$FTVE_PV[1]
584 FTVE_Add <- (FTVE-FTVE_Prev)
585
586 dfu <-MaxResN-MaxResN-(PVUsed+2) # Degrees of freedom for the unexplained variation (found in the
denominator), which appear in the numerator (df1)
587 dfe <- PVUsed - (gp-1) # Degrees of freedom for the explained variation (found in the numerator),
which appear in the denominator (df2)
588 FRatioTest <- ((FTVE_Add*dfe)/((1-FTVE)*dfe)
589
590 }
591
592 P_Value<-as.numeric(1-p/(FRatioTest.dfe.dfu))
593
594 MP3R$MaxRes_E[pv] <- MaxResE
595 MP3R$MaxRes_n[pv] <- MaxResn
596 MP3R$MaxRes_N[pv] <- MaxResN
597 MP3R$MaxRes_AUC[pv] <- MaxResAUC
598 MP3R$FTVE_PV[pv] <- FTVE
599 if (gp>=2) {MP3R$FTVE_AddPV[pv] <- FTVE_Add
600 MP3R$FTVE_PrevPV[pv] <- FTVE_Prev}
601 MP3R$dfupV[pv] <- dfu
602 MP3R$dfepV[pv] <- dfe
603 MP3R$FRatioTestPV[pv] <- FRatioTest
604 MP3R$P_ValuePV[pv] <- P_Value
605 MP3R$AlphaValue[pv] <- AlphaValues[al]
606 MP3R$Regulariz[pv] <- ReguValues[rz]
607
608
609
610 if (gp==1) {
611 if (MP3R$P_ValuePV[pv] <= P_ValueCutOff)
612 { # Model selection based on PValue of FTtest
613 MP3R$Sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
614 } # End of model selection if statement
615 } # End of if gp 1 statement
616
617 if (gp>=2) {
618 MP3R$Sign_PVNamePV[pv] <- NA
619 if (MP3R$Sign_PV[pv] >= MP3R$Sel_GP[[gp-1]]$FTVE_PV[1])
620 {
621 MP3R$Sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
622 } # End of model selection if statement
623 } # End of if gp 2 statement
624 } # End of pv loop
625
626 #####
627 #####
628 #####
629 #####
630 #####
631 #####
632 #####
633 #####
634 #####
635 #####
636 #####
637 #####
638 #####
639 #####
640 #####
641 #####
642 #####
643 #####
644 #####
```


833	if(rv<10){0}, rv,
834	#_OptCode RVName,
835	".CSV",
836	sep=",",row.names=FALSE) #
837	#####
838	#####
839	#####
840	#####
841	#####
842	#####
843	#####
844	#####
845	#####
846	#####
847	#####
848	#####
849	#####
850	#####
851	#####
852	#####
853	#####
854	#####
855	#####
856	#####
857	#####
858	#####
859	#####
860	#####
861	#####
862	#####
863	#####
864	#####
865	#####
866	#####
867	#####
868	#####
869	#####
870	#####
871	#####
872	#####
873	#####
874	#####
875	#####
876	#####
877	#####
878	#####
879	#####
880	#####
881	#####
882	#####
883	#####
884	#####
885	#####
886	#####
887	#####
888	#####
889	#####
890	#####
891	#####
892	#####
893	#####
894	#####
895	#####
896	#####
897	#####
898	#####
899	#####
900	#####
901	#####

```

June 2015 Versi
M3_am_Ail_Scripts_MIAIT2015_V2R
764 # InsCode length(M3A_FinPredData_List)
765 # InsCode str(M3A_FinPredData_List[ip])
766 # InsCode length(M3A_FinPredData_List)
767 ev<-lp ## lp has been the list position counter, which in this case stands for ev
768
769 length(M3A_MP3RSEL_GPLIST[[ev,-ID])
770 head(M3A_MP3RSEL_GPLIST[[ev]])
771
772 if (length(M3A_FinPredData_List)>0){
773   for (ev in seq(length(M3A_FinPredData_List))
774     {
775     if(length(M3A_MP3RSEL_GPLIST[[ev]][ID])>0){ # Only if variables were found
776
777     M3A_FinPredNames_List[[ev]] <- names(M3A_FinPredData_List[[ev]]) # The DV names (second level)
778     names(M3A_FinPredNames_List)[ev] <- names(M3A_FinPredData_List[[ev]]) # The EV names (first level)
779     M3A_FinPred_EVName <- rep(as.vector(names(M3A_FinPredData_List[[ev]]),length(M3A_FinPredData_List[[ev]]))
780     M3A_FinPred_DVName <- as.vector(names(M3A_FinPredData_List[[ev]]))
781     Dfrfm <- as.data.frame(cbind(M3A_FinPred_EVName,M3A_FinPred_DVName))
782     M3A_FinPredNames_Dfrfm <- rbind(M3A_FinPredNames_Dfrfm,Dfrfm)
783     M3A_FinPredData_Dfrfm <- cbind(M3A_FinPredData_Dfrfm,as.data.frame(M3A_FinPredData_List[[ev]])) # Sticks all
784     DVs together into 1 data frame
785
786     M3A_MP3RSEL_1st_GP <- M3A_MP3RSEL_GPLIST[[ev]][ID][,1]
787
788     for (gpl in seq(length(M3A_MP3RSEL_GPLIST[[ev]])))
789     {
790       M3A_MP3RSEL_1st_GP[gpl,] <- M3A_MP3RSEL_GPLIST[[ev]][gpl][1,]
791     } # End of M3A_MP3RSEL_1st_GP
792
793     # InsCode head(MM3A_MP3RSEL_1st_GP) #head(M3A_MP3RSEL_GPLIST[[ev]])
794     M3A_MP3RSEL_1st_EV[[ev]] <- unique(M3A_MP3RSEL_1st_GP)
795     names(M3A_MP3RSEL_1st_EV)[ev] <- names(M3A_FinPredData_List)[ev]
796   } # End of if significant variables found
797 } # End of ev loop, assigning results
798 # InsCode str(M3A_FinPredData_Dfrfm)
799 M3A_FinPredData_Dfrfm <- M3A_FinPredData_Dfrfm[-1]
800 rm(Dfrfm,M3A_FinPred_DVName,M3A_FinPred_EVName)
801
802 } # End of if no significant variables are found
803 ##### Saving to RV #####
804
805 M3A_MP3R_RV[[rv]] <- M3A_MP3R_GPList
806 M3A_MP3RSEL_RV[[rv]] <- M3A_MP3RSEL_GPList
807 M3A_MP3RSEL_Fin_RV[[rv]] <- M3A_MP3RSEL_Fin
808 M3A_MP3RSEL_1st_RV[[rv]] <- M3A_MP3RSEL_1st_EV
809
810 M3A_FinPredData_List_RV[[rv]] <- M3A_FinPredData_List
811 M3A_FinPredNames_List_RV[[rv]] <- M3A_FinPredNames_List
812 M3A_FinPredNames_Dfrfm_RV[[rv]] <- M3A_FinPredNames_Dfrfm
813 M3A_FinPredData_Dfrfm_RV[[rv]] <- M3A_FinPredData_Dfrfm
814
815 write.csv(M3A_FinPredNames_Dfrfm_RV[[rv]],
816           paste(dir,M3A,
817                 "M3A_FinPred_Names",
818                 ".Al",
819                 ".al",
820                 "RV",
821                 if(rv<10){},rv,
822                 #_OptCode RVName,
823                 ".CSV",
824                 sep="").row.names=FALSE)
825
826 write.csv(M3A_FinPredData_Dfrfm_RV[[rv]],
827           paste(dir,M3A,
828                 "M3A_FinPred_Dfrfm",
829                 ".Al",
830                 ".al",
831                 "RV",
832                 "RV",

```

```

902 TempEVDV <- TempEV[[1]][names(M3A_FinPredData_List_A)[a]][r]][ev]]))
903 M3A_Proj_FinPredData_List[[ev]] <- as.list(TempEVDV)
904 names(M3A_Proj_FinPredData_List)[ev] <- names(M3A_FinPredData_List_A)[a]][r]][ev]]
905 } # end of ev
906 M3A_Proj_FinPredData_List_RV[[r]] <- M3A_Proj_FinPredData_List
907 rm(M3A_Proj_FinPredData_List_TempEV.TempEVDV)
908 } # end of RV
909 #_InsCode str(M3A_Proj_FinPredData_List_RV)
910 names(M3A_Proj_FinPredData_List_RV) <- paste("M3A_Proj_FinPredData_List_A", i:length(AlphaValues), sep="")
911 M3A_Proj_FinPredData_List_A[[a]] <- M3A_Proj_FinPredData_List_RV # pass it over to al
912 rm(M3A_Proj_FinPredData_List_RV)
913 } # end of al M3A
914
915 names(M3A_Proj_FinPredData_List_A) <- paste("M3A_Proj_FinPredData_List_A", i:length(AlphaValues), sep="")
916
917 str(M3A_Proj_FinPredData_List_A)
918 length(M3A_Proj_FinPredData_List_A)[a]][r]][ev]]
919 length(M3A_Proj_FinPredData_List_A)[i]][r]][ev]]
920
921 ## End of loop for extracting projection data ###
922
923
924 names(M3A_MP3R_A) <- paste("M3A_MP3R_A", i:length(AlphaValues), sep="")
925 names(M3A_MP3Rsel_A) <- paste("M3A_MP3Rsel_A", i:length(AlphaValues), sep="")
926 names(M3A_MP3Rsel_Fin_A) <- paste("M3A_MP3Rsel_Fin_A", i:length(AlphaValues), sep="")
927
928 names(M3A_FinPredData_List_A) <- paste("M3A_FinPredData_List_A", i:length(AlphaValues), sep="")
929 names(M3A_FinPredNames_List_A) <- paste("M3A_FinPredNames_List_A", i:length(AlphaValues), sep="")
930
931
932 Obj2Save <- object(pattern="M3A")
933 Obj2Save # These are all the M3A objects created in this module
934
935 save(list=Obj2Save,
936 file=paste(dir.rdata,
937 "Obj_",
938 if(DAC>0){DAC},
939 ScriptNameCode,"_",
940 "End_All", ".RData", sep=""))
941
942
943 save.image(paste(dir.rdata,
944 if(DAC>0){DAC},
945 ScriptNameCode,"_",
946 "End_All", ".RData", sep=""))
947
948 #####
949 ### Next script ###
950
951 ##
952 #####
953 #####
954 #####
955 #####
956 #####
957 ##### MIA2 Module 3 - MaxEnt modelling modules #####
958 ##### Alternative MaxEnt procedure #####
959 ##### M3B: Building the most Parsimonious MaxEnt model by forward selection #####
960 ##### by forward selection #####
961 ##### No interactions #####
962 #####
963 ##### M3B.R #####
964 ##### April 2015 #####
965 ##### sabgis@yahoo.com #####
966 ##### #####
967 ##### #####
968 ##### #####

```

```

969 #####
970 ##### NOTE on terms and code abbreviations #####
971 ##### Some of the code abbreviations refer to old terminology, however,
972 ##### the descriptive comments use, as much as possible, the latest terms.
973 ##### Thus, abbreviations like P_ stood for "predictor", but are now simply referred to as
974 ##### "variable", "derived variable" or "significant variable"
975 #####
976 #####
977 #_OptCode require(plyr, MIAPar$dir.Rlib) # used for joining data
978 #_AltCode load("D:/IDA/A01_T4/RData/A01_M3A_End_All.RData") # choose this if no RData
979 ScriptNameCode <- "M3B"
980 SaveTempFlag <- FALSE
981
982 ## As in this section we are running maxent.jar, place this check and instruct use
983
984 jarFileFlag <- paste(dir.in, "maxent.jar", sep="")
985 ### Checking if maxent.jar can be run (normally not necessary as part of the script)
986 if (file.exists(jarFileFlag)==FALSE){
987 cat("!!! WARNING !!! \nPlease place a copy of maxent.jar in your dir.in directory
988 !!! WARNING !!!!) else {All's good, carry on }
989
990 ## Remind users of their maxent.jar settings chosen in Module 1
991 JarPar$M3B_JarFlags
992
993 ## Check if you have data
994
995 length(M3A_FinPredData_List_A)
996 str(M3A_FinPredData_List_A)[i]][r]][ev]]
997 length(M3A_Proj_FinPredData_List_A)
998 names(M3A_Proj_FinPredData_List_A)[i]][r]][ev]]
999
1000 str(M1_N_SWD_RV[[r]][ev]]
1001 #####
1002 #####
1003 ##### Start here #####
1004 ##### 1 EVGroup at a time #####
1005 #####
1006 #####
1007 #####
1008 #####
1009 #####
1010 ##### Setting directories #####
1011 ##### including some that will be used to make output from maxent.jar easier to access
1012 #####
1013 #####
1014 #####
1015 dir.M3B <- paste(dir.max,
1016 "M3B_All", # Main Directory
1017 "\\", sep="")
1018 dir.M3B.res <- paste(dir.M3B,
1019 "M3B_AllRes", # Key Results grouped together
1020 "\\", sep="")
1021
1022 dir.create(dir.M3B)
1023 dir.create(dir.M3B.res)
1024
1025 dir.html <- paste(dir.M3B.res,
1026 "HTML",
1027 "\\", sep="")
1028 dir.create(dir.html)
1029 dir.htmlplots <- paste(dir.html,
1030 "plots",
1031 "\\", sep="")
1032
1033 dir.create(dir.htmlplots)
1034
1035 dir.lamdba <- paste(dir.M3B.res,
1036 "Lambda",
1037 "\\", sep="")
1038
1039 dir.create(dir.lamdba)

```

```

1037 M3_am_All_Scripts_MIA12015_V2.R
1038
1039     dir.Pred_CSV <- paste(dir.M3B.res,
1040                           "Pred_CSV",
1041                           "\\", sep="")
1042     dir.create(dir.Pred_CSV)
1043
1044     dir.res <- paste(dir.M3B.res,
1045                     "Res_Max",
1046                     "\\", sep="")
1047     dir.create(dir.res)
1048
1049     dir.Proj_CSV <- paste(dir.M3B.res, # Adding projected prediction file
1050                           "Proj_CSV",
1051                           "\\", sep="")
1052     dir.create(dir.Proj_CSV)
1053
1054     dir.ProjData_SWD <- paste(dir.M3B.res, # Adding projection data file
1055                               "ProjData_SWD",
1056                               "\\", sep="")
1057
1058     dir.create(dir.ProjData_SWD)
1059     #####
1060     #####
1061     #####
1062     #####
1063     AlphaValues <- MIAPar$AlphaValues
1064     _AltCode AlphaValues <- MIAPar$AlphaValues[c(1,3,5)] # This way you can choose a subset of alpha values
1065
1066     RegulValues <- MIAPar$RegulValues
1067     rz <- 1
1068
1069     M3B_MP3RGP_AI <- as.list(1:length(M3A_FinPredData_List_AI)) # Holds all the RV's
1070     names(M3B_MP3RGP_AI) <- paste("M3B_MP3RGP_AI",1:length(AlphaValues),sep="")
1071
1072     M3B_MP3RSeGP_AI <- as.list(1:length(M3A_FinPredData_List_AI))
1073     names(M3B_MP3RSeGP_AI) <- paste("M3B_MP3RSeGP_AI",1:length(AlphaValues),sep="")
1074
1075     M3B_FinPredData_List_AI <- as.list(1:length(M3A_FinPredData_List_AI))
1076     names(M3B_FinPredData_List_AI) <- paste("M3B_FinPredData_List_AI",1:length(AlphaValues),sep="")
1077
1078     M3B_FinPredNames_List_AI <- as.list(1:length(M3A_FinPredData_List_AI))
1079     names(M3B_FinPredNames_List_AI) <- paste("M3B_FinPredNames_List_AI",1:length(AlphaValues),sep="")
1080
1081     M3B_FinPredNames_Dfrim_AI <- as.list(1:length(M3A_FinPredData_List_AI))
1082     names(M3B_FinPredNames_Dfrim_AI) <- paste("M3B_FinPredNames_Dfrim_AI",1:length(AlphaValues),sep="")
1083
1084     M3B_FinPredData_Dfrim_AI <- as.list(1:length(M3A_FinPredData_List_AI))
1085     names(M3B_FinPredData_Dfrim_AI) <- paste("M3B_FinPredData_Dfrim_AI",1:length(AlphaValues),sep="")
1086
1087     M3B_Proj_FinPredData_List_AI <- as.list(1:length(M3A_FinPredData_List_AI))
1088     names(M3B_Proj_FinPredData_List_AI) <- paste("M3B_Proj_FinPredData_List_AI",1:length(AlphaValues),sep="")
1089
1090     #####
1091     #####
1092     #####
1093     #####
1094     #####
1095     #####
1096     rv<-1
1097     al <- 1 # Decide where to start
1098     ev<-1
1099     for (al in seq(length(M3A_FinPredData_List_AI)))
1100     {
1101         M3A_FinPredData_List_RV <- M3A_FinPredData_List_AI[[al]]
1102         # The equivalent for the projection file skips the whole RV temp name, which is assigned directly later on as [[al]][[rv]]
1103         #####
1104         #####

```

```

1105     #####
1106     #####
1107     #####
1108     DataGP_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1109     names(DataGP_RV) <- paste("DataGP_RV",1:length(RVNames),sep="")
1110
1111     GP_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1112     names(GP_RV) <- paste("GP_RV",1:length(RVNames),sep="")
1113
1114     M3B_MP3RGP_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1115     names(M3B_MP3RGP_RV) <- paste("M3B_MP3RGP_RV",1:length(RVNames),sep="")
1116
1117     M3B_MP3RSeGP_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1118     names(M3B_MP3RSeGP_RV) <- paste("M3B_MP3RSeGP_RV",1:length(RVNames),sep="")
1119
1120     M3B_FinPredData_List_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1121     names(M3B_FinPredData_List_RV) <- paste("M3B_FinPredData_List_RV",1:length(RVNames),sep="")
1122
1123     M3B_FinPredNames_List_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1124     names(M3B_FinPredNames_List_RV) <- paste("M3B_FinPredNames_List_RV",1:length(RVNames),sep="")
1125
1126     M3B_FinPredData_Dfrim_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1127     names(M3B_FinPredData_Dfrim_RV) <- paste("M3B_FinPredData_Dfrim_RV",1:length(RVNames),sep="")
1128
1129     M3B_FinPredNames_Dfrim_RV <- as.list(1:length(M3A_FinPredData_List_RV))
1130     names(M3B_FinPredNames_Dfrim_RV) <- paste("M3B_FinPredNames_Dfrim_RV",1:length(RVNames),sep="")
1131
1132     M3B_Proj_FinPredData_List_RV <- as.list(1:length(M3A_FinPredData_List_RV)) # Extracting length only, not the values
1133     names(M3B_Proj_FinPredData_List_RV) <- paste("M3B_Proj_FinPredData_List_RV",1:length(RVNames),sep="")
1134
1135     #####
1136     #####
1137     #####
1138     #####
1139     #####
1140     #####
1141     rv<- 1 # The user can decide where to start
1142     for (rv in seq(length(M3A_FinPredData_List_RV)))
1143     {
1144         if (length(M3A_FinPredData_List_RV[[rv]]==0){ # This is in case there are no significant variables at all
1145             cat(paste("No significant variables picked at all for Alpha",al,"RV",rv))
1146             # Assign even if no significant variables found to maintaing list position
1147             DataGP_RV[[rv]] <- NA
1148             GP_RV[[rv]] <- NA
1149             M3B_MP3RGP_RV[[rv]] <- NA # Ensures that the list position is maintained
1150             M3B_MP3RSeGP_RV[[rv]] <- NA # Ensures that the list position is maintained
1151             # str(M3B_FinPredData_List)
1152             M3B_FinPredData_List_RV[[rv]] <- NA # Ensures that the list position is maintained
1153             M3B_FinPredNames_List_RV[[rv]] <- NA # Ensures that the list position is maintained
1154             M3B_FinPredNames_Dfrim_RV[[rv]] <- NA # Ensures that the list position is maintained
1155             M3B_Proj_FinPredData_List_RV[[rv]] <- NA # Ensures that the list position is maintained
1156             M3B_Proj_FinPredData_Dfrim_RV[[rv]] <- NA # Ensures that the list position is maintained
1157
1158             M3B_Proj_FinPredData_List_RV[[rv]] <- NA
1159         } else {
1160             RVXY <- M3_N_SWD_RV[[rv]][1:2-4]
1161             RVName <- paste("aNI",RV",if(rv<10){0},rv,"AI",al, sep="")
1162             P.ValueCutoff <- AlphaValues[al]
1163             dir.M3B_GP <- paste(dir.M3B,
1164                                "M3B_",
1165                                RVName,
1166                                "\\",
1167                                sep="")
1168             dir.create(dir.M3B_GP)
1169
1170             M3A_FinPredData <- M3A_FinPredData_List_RV[[rv]]
1171             M3A_Proj_FinPredData <- M3A_Proj_FinPredData_List_AI[[al]][[rv]] # This command removes extras in Null EV's
1172
1173             for (lp in seq(length(M3A_FinPredData))) # For NULL EV's only

```

```
1175 {
1176   if(is.na(M3A_FinPredData[[ip]]==TRUE){M3A_FinPredData[[ip]<-NULL
1177     M3A_Proj_FinPredData[[ip]<-NULL
1178   }
1179 } # End Ip ###
1180
1181 SWDList <- M3A_FinPredData # Making this generic so it can be used throughout
1182 SWDList_Proj <- M3A_Proj_FinPredData # Making this generic so it can be used throughout
1183 RVXY_Proj <- M1_N_SWD_RV[rv[1]:4] # this is M3B specific, RVXY in M3C will contain M3B significant variable as well
1184 cat(paste("n","Module 3B Checking",length(SWDList),"grouped variables for rv n.",if(rv<10){0},rv,RVName,"Alpha",al,
1185   P_ValueCutOff))
1186
1187 #####
1188 ##### Defining objects for BRA loops   ###
1189 #####
1190 #####
1191
1192 DataGP <- list()
1193 GP <- list()
1194 GPList <- list()
1195 M3B_MP3RCP <- list()
1196 M3B_MP3RselCP <- list()
1197
1198 DataGPProj<- list()
1199 CPProj <- list()
1200 GPListProj <- list()
1201 #####
1202
1203 DataGP[[1]] <- as.list(SWDList)
1204 DataGPProj[[1]] <- as.list(SWDList_Proj)
1205 # _InsCode str(DataGP[[1]])
1206 # _InsCode str(DataGPProj[[1]])
1207 #####
1208 #####
1209 ##### gp loop starts here,
1210 ##### first by defining objects
1211 #####
1212 #####
1213 #####
1214
1215 names(DataGP) <- paste("M3B", "_GP","0",1,sep="") # This is the first round
1216 # _InsCode names(DataGP[[1]])
1217 # _InsCode str(DataGP)
1218 # _InsCode names(M1_N_DataGP[[1]])
1219 # _InsCode str(M1_N_DataGP[[1]])
1220 #####
1221 #####
1222 ### Preparing a dataframe in which all key information from each MaxEnt run is stored.
1223 This will be used in many ways.
1224 It will be saved as a list of list of dataframes
1225 depending on how many levels are considered
1226 #####
1227 #####
1228 ##### MaxResNames #####
1229 MaxResNames <- as.vector(c(
1230   "GP Name", # GP stands for Grouped significant variable, which is the "set of DVs representing one EV"
1231   "GP Num", # The number of grouped significant variable
1232   "PV_NameGP", # Individual significant variable used, separated by space
1233   "RV_Name",
1234   "GP_Best",
1235   "NPred_Used",
```

```
1236 "NPred_Test",
1237 "MaxOut",
1238 "MaxHTMLFile",
1239 "MaxPredFile",
1240 "MaxLambdaFile",
1241 "MaxResultsFile",
1242 "MaxRes_E",
1243 "MaxRes_n",
1244 "MaxRes_N",
1245 "MaxRes_AUC",
1246 "FTVE_PV",
1247 "FTVE_PrevPV",
1248 "FTVE_AddPV",
1249 "dtuPV",
1250 "dfePV",
1251 "FRatioTestPV",
1252 "P_ValuePV",
1253 "P_Value_BestPV", # This will be used as the "previous" in the next round
1254 "Name_BestPV",
1255 "Name_AddPV",
1256 "Sign_RankPV",
1257 "Sign_PVNamePV",
1258 "AlphaValue",
1259 "Regulariz",
1260 "MaxProjFile", # Just the projection data
1261 "MaxProjPredFile", # The projected prediction
1262 "MaxBatFile"
1263 ))
1264
1265 #####
1266 #####
1267 #####
1268 #####
1269 ##### Pick up from here after you've selected the best and most significant Grouped variable,   ###
1270 ##### (i.e., gp loop)
1271 #####
1272 #####
1273 #####
1274 #####
1275 ##### gp loop - iterations   ###
1276 #####
1277
1278 BestPVNamesGP <- NULL
1279 AddPVNamesGP <- NULL
1280 gp<-0
1281 GPOopExit<-FALSE
1282
1283 while (GPOopExit==FALSE)
1284 {
1285   gp<-gp+1
1286   # _InsCode names(DataGP[[gp]])
1287
1288   RVName <- paste("a","N1","RV",if(rv<10){0},rv,"A",1,sep="") # Adding A for Alternative maxent.jar (AM)
1289   RVName <- paste(RVName,"p",
1290     if(gp<10){0},
1291     gp,sep="")
1292
1293   ### Below, the data are generated that will be used by maxent.jar to test each significant variable. I do this separately
1294   so it can be
1295   ### distinguished from the max.swd data-testing procedure, which will be "improved anyway" later on #Why "" here?
1296   if (gp==1) {GP[[gp]] <- DataGP[[gp]]
```



```

1297   GPProj[[gp]] <- DataGPProj[[gp]]$names(DataGP[[gp]])
1298   # _InsCode str(GPProj[[gp]]) # str(GP[[gp]])
1299   } # End of if gp==1 GP and GPProj data assignment
1300   if (gp>1) {GP[[gp]] <- as.list(t:(length(AddPVNamesGP))) # _InsCode str(GP[[gp]])
1301   GPProj[[gp]] <- as.list(t:(length(AddPVNamesGP))) # Added projection data holder (MI)
1302   pv<-1
1303   for (pv in seq(length(GP[[gp]]))) # This creates the new data for iteration 2, by picking the first (best) GP and
      performing GPProj data assignment
    {
1304       GPNames<-c(BestPVNamesGP,AddPVNamesGP[pv])
1305       GP[[gp]][pv]<- as.list(DataGP[[gp]][GPNames])
1306       # _InsCode names( GP[[gp]][[pv]]) # str(GP[[gp]][[pv]])
1307       GPProj[[gp]][pv]<- as.list(DataGPProj[[gp]][GPNames])
1308       # _InsCode str(GPProj[[gp]][[pv]])
1309       } # End of pv in building GP dataset
1310       } # End of if gp>1 GP and GPProj data assignment
1311   }
1312   cat("\n","Module 38 Iteration: ", gp, ", RV & Alpha:", RVName)
1313   cat("\n","Number of Grouped variable to test: ",(length(GP[[gp]])))
1314
1315   names(GP)[gp] <- paste("M3B_Data","GP",if(gp<10){},gp,sep="")
1316   names(GPProj)[gp] <- paste("M13B_Data","GP",if(gp<10){},gp,sep="")
1317
1318   # _InsCode str(GP[[gp]]) # names(GP[gp]) # names(GP[[gp]][[1]])
1319   # _InsCode names(GP[[gp]][[2]]) # names(GP[[gp]][[2]][[1]])
1320   # _InsCode str(GPProj[[gp]])
1321   # _InsCode names(GPProj[[gp]][[2]])
1322   # _InsCode names(GPProj[[gp]][[2]][[1]])
1323   # _InsCode names(GPProj[[gp]][[2]][[1]])
1324   # _InsCode names(GP[[gp]][[2]][[1]])
1325   # _InsCode names(GPProj[[gp]][[2]][[1]])
1326
1327   #####
1328   ####
1329   ###
1330   #####
1331   #####
1332   #####
1333   #####
1334   #####
1335   #####
1336   #####
1337   #####
1338   #####
1339   #####
1340   #####
1341   dir.max.gp<-paste(dir.M3B_GP,
1342   "GP",
1343   if(gp<10){},
1344   gp,
1345   "\n",
1346   sep="")
1347   dir.create(dir.max.gp)
1348   #####
1349   #####
1350   #####
1351   #####
1352   #####
1353   #####
1354   #####
1355   #####
1356   #####
1357   #####
1358   #####
1359   #####
1360   #####
1361   #####
1362   #####
1363   #####
1364   #####
1365   #####
1366   #####
1367   #####
1368   #####
1369   #####
1370   #####
1371   #####
1372   #####
1373   #####
1374   #####
1375   #####
1376   #####
1377   #####
1378   #####
1379   #####
1380   #####
1381   #####
1382   #####
1383   #####
1384   #####
1385   #####
1386   #####
1387   #####
1388   #####
1389   #####
1390   #####
1391   #####
1392   #####
1393   #####
1394   #####
1395   #####
1396   #####
1397   #####
1398   #####
1399   #####
1400   #####
1401   #####
1402   #####
1403   #####
1404   #####
1405   #####
1406   #####
1407   #####
1408   #####
1409   #####
1410   #####
1411   #####
1412   #####
1413   #####
1414   #####
1415   #####
1416   #####
1417   #####
1418   #####
1419   #####
1420   #####
1421   #####
1422   #####
1423   #####
1424   #####
1425   #####
1426   #####
1427   #####
1428   #####
1429   #####
1430   #####
1431   #####
1432   #####
1433   #####
1434   #####
1435   #####
1436   #####
1437   #####
1438   #####
1439   #####
1440   #####
1441   #####
1442   #####
1443   #####
1444   #####
1445   #####
1446   #####
1447   #####
1448   #####
1449   #####
1450   #####
1451   #####
1452   #####
1453   #####
1454   #####
1455   #####
1456   #####
1457   #####
1458   #####
1459   #####
1460   #####
1461   #####
1462   #####
1463   #####
1464   #####
1465   #####
1466   #####
1467   #####
1468   #####
1469   #####
1470   #####
1471   #####
1472   #####
1473   #####
1474   #####
1475   #####
1476   #####
1477   #####
1478   #####
1479   #####
1480   #####
1481   #####
1482   #####
1483   #####
1484   #####
1485   #####
1486   #####
1487   #####
1488   #####
1489   #####
1490   #####
1491   #####
1492   #####
1493   #####
1494   #####
1495   #####
1496   #####
1497   #####
1498   #####
1499   #####
1500   #####
1501   #####
1502   #####
1503   #####
1504   #####
1505   #####
1506   #####
1507   #####
1508   #####
1509   #####
1510   #####
1511   #####
1512   #####
1513   #####
1514   #####
1515   #####
1516   #####
1517   #####
1518   #####
1519   #####
1520   #####
1521   #####
1522   #####
1523   #####
1524   #####
1525   #####
1526   #####
1527   #####
1528   #####
1529   #####
1530   #####
1531   #####
1532   #####
1533   #####
1534   #####
1535   #####
1536   #####
1537   #####
1538   #####
1539   #####
1540   #####
1541   #####
1542   #####
1543   #####
1544   #####
1545   #####
1546   #####
1547   #####
1548   #####
1549   #####
1550   #####
1551   #####
1552   #####
1553   #####
1554   #####
1555   #####
1556   #####
1557   #####
1558   #####
1559   #####
1560   #####
1561   #####
1562   #####
1563   #####
1564   #####
1565   #####
1566   #####
1567   #####
1568   #####
1569   #####
1570   #####
1571   #####
1572   #####
1573   #####
1574   #####
1575   #####
1576   #####
1577   #####
1578   #####
1579   #####
1580   #####
1581   #####
1582   #####
1583   #####
1584   #####
1585   #####
1586   #####
1587   #####
1588   #####
1589   #####
1590   #####
1591   #####
1592   #####
1593   #####
1594   #####
1595   #####
1596   #####
1597   #####
1598   #####
1599   #####
1600   #####
1601   #####
1602   #####
1603   #####
1604   #####
1605   #####
1606   #####
1607   #####
1608   #####
1609   #####
1610   #####
1611   #####
1612   #####
1613   #####
1614   #####
1615   #####
1616   #####
1617   #####
1618   #####
1619   #####
1620   #####
1621   #####
1622   #####
1623   #####
1624   #####
1625   #####
1626   #####
1627   #####
1628   #####
1629   #####
1630   #####
1631   #####
1632   #####
1633   #####
1634   #####
1635   #####
1636   #####
1637   #####
1638   #####
1639   #####
1640   #####
1641   #####
1642   #####
1643   #####
1644   #####
1645   #####
1646   #####
1647   #####
1648   #####
1649   #####
1650   #####
1651   #####
1652   #####
1653   #####
1654   #####
1655   #####
1656   #####
1657   #####
1658   #####
1659   #####
1660   #####
1661   #####
1662   #####
1663   #####
1664   #####
1665   #####
1666   #####
1667   #####
1668   #####
1669   #####
1670   #####
1671   #####
1672   #####
1673   #####
1674   #####
1675   #####
1676   #####
1677   #####
1678   #####
1679   #####
1680   #####
1681   #####
1682   #####
1683   #####
1684   #####
1685   #####
1686   #####
1687   #####
1688   #####
1689   #####
1690   #####
1691   #####
1692   #####
1693   #####
1694   #####
1695   #####
1696   #####
1697   #####
1698   #####
1699   #####
1700   #####
1701   #####
1702   #####
1703   #####
1704   #####
1705   #####
1706   #####
1707   #####
1708   #####
1709   #####
1710   #####
1711   #####
1712   #####
1713   #####
1714   #####
1715   #####
1716   #####
1717   #####
1718   #####
1719   #####
1720   #####
1721   #####
1722   #####
1723   #####
1724   #####
1725   #####
1726   #####
1727   #####
1728   #####
1729   #####
1730   #####
1731   #####
1732   #####
1733   #####
1734   #####
1735   #####
1736   #####
1737   #####
1738   #####
1739   #####
1740   #####
1741   #####
1742   #####
1743   #####
1744   #####
1745   #####
1746   #####
1747   #####
1748   #####
1749   #####
1750   #####
1751   #####
1752   #####
1753   #####
1754   #####
1755   #####
1756   #####
1757   #####
1758   #####
1759   #####
1760   #####
1761   #####
1762   #####
1763   #####
1764   #####
1765   #####
1766   #####
1767   #####
1768   #####
1769   #####
1770   #####
1771   #####
1772   #####
1773   #####
1774   #####
1775   #####
1776   #####
1777   #####
1778   #####
1779   #####
1780   #####
1781   #####
1782   #####
1783   #####
1784   #####
1785   #####
1786   #####
1787   #####
1788   #####
1789   #####
1790   #####
1791   #####
1792   #####
1793   #####
1794   #####
1795   #####
1796   #####
1797   #####
1798   #####
1799   #####
1800   #####
1801   #####
1802   #####
1803   #####
1804   #####
1805   #####
1806   #####
1807   #####
1808   #####
1809   #####
1810   #####
1811   #####
1812   #####
1813   #####
1814   #####
1815   #####
1816   #####
1817   #####
1818   #####
1819   #####
1820   #####
1821   #####
1822   #####
1823   #####
1824   #####
1825   #####
1826   #####
1827   #####
1828   #####
1829   #####
1830   #####
1831   #####
1832   #####
1833   #####
1834   #####
1835   #####
1836   #####
1837   #####
1838   #####
1839   #####
1840   #####
1841   #####
1842   #####
1843   #####
1844   #####
1845   #####
1846   #####
1847   #####
1848   #####
1849   #####
1850   #####
1851   #####
1852   #####
1853   #####
1854   #####
1855   #####
1856   #####
1857   #####
1858   #####
1859   #####
1860   #####
1861   #####
1862   #####
1863   #####
1864   #####
1865   #####
1866   #####
1867   #####
1868   #####
1869   #####
1870   #####
1871   #####
1872   #####
1873   #####
1874   #####
1875   #####
1876   #####
1877   #####
1878   #####
1879   #####
1880   #####
1881   #####
1882   #####
1883   #####
1884   #####
1885   #####
1886   #####
1887   #####
1888   #####
1889   #####
1890   #####
1891   #####
1892   #####
1893   #####
1894   #####
1895   #####
1896   #####
1897   #####
1898   #####
1899   #####
1900   #####
1901   #####
1902   #####
1903   #####
1904   #####
1905   #####
1906   #####
1907   #####
1908   #####
1909   #####
1910   #####
1911   #####
1912   #####
1913   #####
1914   #####
1915   #####
1916   #####
1917   #####
1918   #####
1919   #####
1920   #####
1921   #####
1922   #####
1923   #####
1924   #####
1925   #####
1926   #####
1927   #####
1928   #####
1929   #####
1930   #####
1931   #####
1932   #####
1933   #####
1934   #####
1935   #####
1936   #####
1937   #####
1938   #####
1939   #####
1940   #####
1941   #####
1942   #####
1943   #####
1944   #####
1945   #####
1946   #####
1947   #####
1948   #####
1949   #####
1950   #####
1951   #####
1952   #####
1953   #####
1954   #####
1955   #####
1956   #####
1957   #####
1958   #####
1959   #####
1960   #####
1961   #####
1962   #####
1963   #####
1964   #####
1965   #####
1966   #####
1967   #####
1968   #####
1969   #####
1970   #####
1971   #####
1972   #####
1973   #####
1974   #####
1975   #####
1976   #####
1977   #####
1978   #####
1979   #####
1980   #####
1981   #####
1982   #####
1983   #####
1984   #####
1985   #####
1986   #####
1987   #####
1988   #####
1989   #####
1990   #####
1991   #####
1992   #####
1993   #####
1994   #####
1995   #####
1996   #####
1997   #####
1998   #####
1999   #####
2000   #####

```

```

1362   # _InsCode str(GP[[gp]])
1363   # _InsCode str(Max.SWD)
1364   Max.SWD.Proj <-as.data.frame(cbind(RVXY_Proj,GPProj[[gp]][pv]))
1365   PVUsed <- ncol(Max.SWD)-3
1366   PVNames <-names(GP[[gp]][pv]) # _InsCode names(GP[[gp]][[pv]])
1367   GP_Name <-names(GP[[gp]][pv]) # _InsCode names(GP[[gp]][[pv]])
1368   BestPVNamesGP <- NA
1369   BestPVNamesGP$string <- GP_Name
1370   PVTest <- names(GP[[gp]][pv])
1371   AddPVNamesGP <- PVTest
1372   NPred_Test<- PVUsed # _InsCode length(GP[[gp]][[pv]])
1373   }
1374
1375   if (gp >= 2) {
1376       Max.SWD<-as.data.frame(cbind(RVXY,GP[[gp]][pv]))
1377       # _InsCode str(Max.SWD)
1378       # _InsCode names(GP[[gp]][pv])
1379       # _InsCode str(GP[[gp]][pv])
1380       # _InsCode names(GP[[gp]][[pv]][[2]])
1381       Max.SWD.Proj <-as.data.frame(cbind(RVXY_Proj,GPProj[[gp]][pv]))
1382       # _InsCode str(Max.SWD.Proj)
1383       names(Max.SWD)
1384       PVUsed <- ncol(Max.SWD)-3
1385       names(Max.SWD)
1386       PVNames <-names(Max.SWD[5:length(Max.SWD)]) # These are the names of the variables
1387       GP_Name <- paste(M3B_MP3R$GP[[gp-1]]$Name_BestPV[1:1],AddPVNamesGP[pv],sep="")
      names(GP[[gp]][[pv]])
1388       BestPVNamesGP$string <- M3B_MP3R$GP[[gp-1]]$Name_BestPV[1]
1389       PVTest <- AddPVNamesGP[pv]
1390       NPred_Test<- length(DataGP[[gp]][AddPVNamesGP[pv]])
1391       EVNames <- BestPVNamesGP$string
1392       } # End of if gp loop
1393
1394   NPred_Used <- PVUsed # or the most complex model
1395
1396   cat("\nMax Checking", gp, "Grouped variables:",na.omit(BestPVNamesGP), PVTest)
1397
1398   MaxRunDir <- paste(dir.max.gp,
1399   paste("C",gp),if(gp<10){},gp,sep=""),
1400   "pv",
1401   pv,
1402   sep="")
1403
1404   ## The directory name/Grouped variable name
1405   MaxRunName <- paste("C",gp",if(gp<10){},gp,
1406   "pv",
1407   pv,
1408   sep="")
1409
1410   MaxOut<-paste(MaxRunDir,"\\",
1411   sep="")
1412   dir.create(MaxOut)
1413   Max.Bkgrnd.SWD.Name <- paste(MaxRunDir,
1414   ".csv",
1415   sep="")
1416   Max.SWD.Proj.Name <- paste(MaxRunDir,
1417   "_Proj",
1418   ".csv",
1419   sep="")
1420
1421   Max.Bkgrnd.SWD<- Max.SWD # The big N, which includes n in the case of forward subset selection
1422   write.csv(Max.Bkgrnd.SWD,file==Max.Bkgrnd.SWD.Name,row.names=FALSE)
1423   write.csv(Max.SWD.Proj,file==Max.SWD.Proj.Name,row.names=FALSE)
1424
1425   MaxResFile<-paste(MaxOut,
1426   RVName,
1427   "Results.csv",
1428   sep="")
1429   Prespos <-which(Max.SWD[1]== "UB") # the first column in an swd file is sample or uninformed background
1430   Max.Sample<-Max.SWD[Prespos,] # Extract only the portion with sample presence

```



```

1431 Max.Sample.Name<-paste(MaxRunDir,
1432   " ",
1433   RVName,
1434   ".csv",
1435   sep="")
1436
1437 Max.Sample[,1]<- RVName
1438 names(Max.Sample[,1])<"Sample"
1439 write.csv(Max.Sample,file=Max.Sample.Name,row.names=FALSE)
1440
1441 MaxHTMLFile <- paste(MaxOut,
1442   RVName,
1443   ".html",
1444   sep="")
1445
1446 MaxPredFile <- paste(MaxOut,
1447   RVName,
1448   ".csv",
1449   sep="")
1450
1451 MaxLambdaFile <- paste(MaxOut,
1452   RVName,
1453   ".lambdas",
1454   sep="")
1455
1456 MaxProjPredFile <- paste(MaxOut, # Projected prediction
1457   RVName,
1458   " ", # Projected prediction file name
1459   MaxRunName, # Projected prediction file name
1460   ".Proj", # This is the new name
1461   ".csv",
1462   sep="")
1463
1464 MaxProjFile <- paste(Max.SWD.Proj.Name,
1465   sep="")
1466 #####
1467 #####
1468 ##### Preparing batch file for maxent.jar #####
1469 ##### with the JarFlags settings chosen in Module 1 #####
1470 #####
1471 #####
1472
1473 MaxRunBat<-paste("java -Xmx512m -jar ",
1474   jarFileFlag,
1475   JarPar$M3B_JarFlags,
1476   " samplesfile=",
1477   Max.Sample.Name,
1478   " outputdirectory=",
1479   MaxOut,
1480   " environmentalayers=",
1481   Max.Bkgnd.SWD.Name, # This is the "uninformed background" file that is called
1482   " projectionlayers=", Max.SWD.Proj.Name,
1483   " logfile=",
1484   RVName,
1485   " log",
1486   " visible=FALSE autorun",
1487   sep="")
1488
1489 write(MaxRunBat,paste(MaxRunDir,
1490   ".bat",
1491   sep=""))
1492
1493 system(paste(MaxRunBat),wait=TRUE) # Running maxent.jar from R as specified in MaxRunBat
1494
1495 #####
1496 #####
1497 ##### Building the MaxEnt output results dataframe #####
1498 ##
1499 #####
1500 #####

```

```

1501 MP3R$GP_Name[pv] <- GP_Name
1502 MP3R$GP_Num[pv] <- gp
1503 MP3R$PV_NameGP[pv] <- paste(GP,".if(gp<|0|0|0|gp,"pv".if(pv<|0|0|0|pv, sep=""))
1504 MP3R$Name_BestPV[pv] <- BestPVNamesGPString
1505 MP3R$Name_AddPV[pv] <- PVTest
1506 MP3R$NPred_Used[pv] <- NPred_Used
1507 MP3R$NPred_Test[pv] <- NPred_Test
1508 MP3R$RV_Name[pv] <- RVName
1509 MP3R$MaxOut[pv] <- MaxOut
1510 MP3R$MaxHTMLFile[pv] <- MaxHTMLFile
1511 MP3R$MaxPredFile[pv] <- MaxPredFile
1512 MP3R$MaxLambdaFile[pv] <- MaxLambdaFile
1513 MP3R$MaxResultsFile[pv] <- MaxResFile
1514 MP3R$AlphaValue[pv] <- P_ValueCutoff
1515 MP3R$Regulariz[pv] <- ReguValues[rz]
1516 MP3R$MaxProjFile[pv] <- MaxProjFile
1517 MP3R$MaxProjPredFile[pv]<- MaxProjPredFile
1518 MP3R$MaxBatFile[pv] <- MaxRunBat
1519 } # End of pv loop # Inner loop, by column
1520
1521 cat("\nDone with maxent.jar Run\n")
1522 M3B_MP3RGP[[gp]] <- MP3R
1523 # _InsCode str(M3B_MP3RGP[[gp]])
1524
1525 #####
1526 #####
1527 ##### MaxEnt
1528 ##### Results compiled for analysis
1529 #####
1530 #####
1531 #####
1532 MP3R <- M3B_MP3RGP[[gp]]
1533 #####
1534 #####
1535 ##### Loop 1
1536 ##### pv - each line of grouped variables
1537 ##### Updating results
1538 #####
1539 #####
1540 #####
1541 #####
1542 #####
1543 MaxResT <- t(MaxRes) # Transposed for two reasons, readability and better coercing of values
1544 MaxResE <- as.numeric(MaxResT$rownames(MaxResT))==Entropy[J] # MaxEnt model's entropy value
1545 MaxResN <- as.numeric(MaxResT$rownames(MaxResT))==X.Training.samples[J] # Get the number of samples
1546 MaxResN <- as.numeric(MaxResT$rownames(MaxResT))==X.Background.points[J] # Get the background points
1547 MaxResAUC <- as.numeric(MaxResT$rownames(MaxResT))==Training.AUC[J] # Get training AUC
1548 FTVE<-((log(MaxResN))-MaxResE)/((log(MaxResN)-log(MaxResN)))
1549
1550 if (gp==1) {dfu <- MaxResN-MaxResN-2-MP3R$NPred_Used[pv] # Degrees of freedom for the unexplained variation
# (found in the denominator) which appear in the numerator (df1)
dfc<- MP3R$NPred_Test[pv] # Degrees of freedom for the explained variation found in the
numerator, which appear in the denominator (df2)
RatioTest <- (FTVE*dfu)/((1-FTVE)*dfc)
}
1551
1552
1553
1554
1555 if (gp>=2) {FTVE_Prev <- M3B_MP3RSELGP[[gp-1]]$FTVE_PVT[J]
FTVE_Add <- (FTVE-FTVE_Prev)
dfu <-MaxResN-MaxResN-2-MP3R$NPred_Used[pv] # Degrees of freedom for the unexplained variation
# (found in the denominator) which appear in the numerator (df1)
dfc <- MP3R$NPred_Test[pv] # Degrees of freedom for the explained variation found in the numerator,
which appear in the denominator (df2)
RatioTest <- ((FTVE_Add*dfu)/((1-FTVE)*dfc)
}
1556
1557
1558
1559
1560 P_Value<-as.numeric(" -p"(FRatioTest,dfc,dfu))
1561
1562
1563 MP3R$MaxRes_E[pv] <- MaxResE
1564 MP3R$MaxRes_n[pv] <- MaxResN
1565 MP3R$MaxRes_N[pv] <- MaxResN
1566 MP3R$MaxRes_AUC[pv] <- MaxResAUC

```

```
1567 MP3R$FTVE_PV[pv] <- FTVE
1568 if (gp>=2){MP3R$FTVE_AddPV[pv] <- FTVE_Add
1569   MP3R$FTVE_PrevPV[pv] <- FTVE_Prev}
1570 MP3R$dfuPV[pv] <- dfu
1571 MP3R$dfuPV[pv] <- dfu
1572 MP3R$FRatioTestPV[pv] <- FRatioTest
1573 MP3R$p_ValuePV[pv] <- p_Value
1574
1575 if (gp==1){
1576   if (MP3R$p_ValuePV[pv] <= p_ValueCutoff)
1577   { # Model selection based on PValue of FTest
1578     MP3R$Sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
1579   } # End of model selection if statement
1580   } # End of if gp 1 statement
1581
1582 if (gp>=2){
1583   MP3R$Sign_PVNamePV[pv] <- NA
1584   if (MP3R$FTVE_PV[pv] >= M3B_MP3RselGP[[gp-1]]$FTVE_PV[1])
1585   {
1586     MP3R$Sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
1587   } # End of model selection if statement
1588   } # End of if gp 2 statement
1589   } # End of pv loop
1590
1591 M3B_MP3RGP[[gp]] <- MP3R # With new information -- after applying model selection criteria
1592 #####
1593 #####
1594 #####
1595 ##### Printing all the Iterations' MP3R #####
1596 #####
1597 write.csv(M3B_MP3RGP[[gp]],
1598   paste(dir.M3B_GP,
1599     if(DAC>0){DAC,
1600       "M3B_",
1601       "MP3R_",
1602       RVName,
1603       ".csv",
1604       sep=""},row.names=FALSE)
1605
1606 #####
1607 #####
1608 ##### Select results that meet selection criteria #####
1609 #####
1610 #####
1611 #####
1612
1613 SignRows <- which((M3B_MP3RGP[[gp]]$p_ValuePV)<=p_ValueCutoff) # Picking only significant rows
1614
1615 if (gp==1&length(SignRows)==0){ cat("\n No significant variables found at all, finished with maxent.jar\n")
1616   gpbest <- NA
1617   M3B_MP3RGP[[gp]]$GP_Best <- gpbest # Number of grouped variables in best model
1618   BestPVNamesGP <- NULL
1619   AddPVNamesGP <- NULL
1620   M3B_FinPredData_Difrm <- NA
1621   M3B_MP3RselGP[[gp]] <- NA
1622   GLoopExit<-TRUE
1623 }
1624
1625 if (gp>=1&length(SignRows)>1){ # Has to stay greater than 1; only to be entered if there are 2 more significant
1626   variables to test against each other
1627   cat("\n additional significant variables found, ranking and testing again")
1628
1629   MP3Rsel <- M3B_MP3RGP[[gp]]$SignRows] # Keeping just those that make the PValueCutoff
1630   MP3RselOrderPValueRatio <- MP3Rsel$order(MP3Rsel$p_ValuePV,-MP3Rsel$FRatioTestPV)] # First by PValue,
1631   then by F-ratio test
1632   MP3RselOrderValueRatio$Sign_RankPV <- 1:nrow(MP3RselOrderPValueRatio) # Assigning order value
1633   M3B_MP3RselGP [[gp]] <- MP3RselOrderPValueRatio
1634   names(M3B_MP3RselGP[[gp]] <- names(M3B_MP3RGP[[gp]] #_InsCode names(M3B_MP3RselGP)
```

```
1635 M3B_MP3RselGP[[gp]]$Name_BestPV <- M3B_MP3RselGP[[gp]]$GP_Name[1]
1636 if (gp==1){BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(M3B_MP3RselGP[[gp]]$Name_AddPV[1,"_"])))
1637   M3B_MP3RselGP[[gp]]$Name_BestPV <- BestPVNamesGP
1638 } # End of if gp==1 for assigning bestPvName
1639 if (gp>1){BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(M3B_MP3RselGP[[gp]]$Name_BestPV[1,"_"])))
1640   } # End of gp >1 for assigning bestPvName
1641
1642 AddPVNamesGP <- as.vector(na.omit(M3B_MP3RselGP[[gp]]$Name_AddPV[2:nrow(M3B_MP3RselGP[[gp]])))
1643
1644 #####
1645 #####
1646 ##### Print the MP3Rsel_GP as you find them in the SignRow>1 Loop #####
1647 #####
1648 #####
1649
1650 write.csv(M3B_MP3RselGP[[gp]],
1651   paste(dir.M3B_GP, ## Write out when left with 1
1652     if(DAC>0){DAC,
1653       "M3B_",
1654       "MP3Rsel_",
1655       RVName,
1656       ".csv",
1657       sep=""},row.names=FALSE)
1658
1659 #####
1660 #####
1661 ##### Creating new datasets for the next iteration
1662 #####
1663 ##### gp +1 is picked from the "best"
1664 #####
1665 #####
1666 #####
1667 DataGP[[gp+1]] <- as.list(DataGP[[gp]][c(BestPVNamesGP, AddPVNamesGP)])
1668 names(DataGP[[gp+1]] <- paste("M3B_", GP,"0",gp+1, sep="") # This is the first round
1669 DataGPProj[[gp+1]] <- as.list(DataGPProj[[gp]][c(BestPVNamesGP,AddPVNamesGP)])
1670 names(DataGPProj)[gp+1] <- paste("M3B_", " _GPPro","0",gp+1, sep="")
1671
1672 } # End of if (gp>=1&length(SignRows)>1) statement (this is the "Select and rank and prepare new variable datasets to
1673   check loop")
1674
1675 if (gp>1&length(SignRows)==0){ cat("\n No additional significant variables found, finished with maxent.jar\n")
1676   BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(M3B_MP3RselGP[[gp-1]]$Name_BestPV[1,"_"]
1677     )))
1678   AddPVNamesGP <- NULL # No Additional significant variables were picked -- none improved
1679   the model
1680   M3B_FinPredData_Difrm <- as.data.frame(DataGP[[gp]][c(BestPVNamesGP,AddPVNamesGP)])
1681
1682   M3B_FinPredData_List <- as.list(DataGP[[gp]][c(BestPVNamesGP,AddPVNamesGP)])
1683   gpbest <- gp-1
1684   M3B_MP3RselGP[[gpbest]]$GP_Best[1] <- gpbest
1685   RVName <- M3B_MP3RselGP[[gpbest]]$RV_Name[1]
1686   GLoopExit<-TRUE # No significant improvement or model - exit the gp loop
1687 }
1688 if (gp>=1&length(SignRows)==1){cat("\n Last significant variable found, finished with maxent.jar\n")
1689   gpbest <- gp
1690   M3B_MP3RGP[[gp]]$GP_Best <- gpbest
1691   #####
1692   ##### Printing this last MP3R which is updated #####
1693   #####
1694   write.csv(M3B_MP3RGP[[gp]],
1695     paste(dir.M3B_GP,
1696       if(DAC>0){DAC,
1697         "M3B_",
1698         "MP3R_",
1699         RVName,
1700         ".csv",
1701         sep=""},row.names=FALSE)
```

```
1697         sep=""),row.names=FALSE)
1698
1699     M3B_MP3RSELGP[[gp]] <- M3B_MP3RGP[[gp]][SignRows,]
1700     write.csv(M3B_MP3RSELGP[[gp]][,1],
1701             paste(dir,M3B_CP,
1702                 if(DAC>0){DAC},
1703                 "M3B_",
1704                 "MP3RSEL_",
1705                 RVName,
1706                 ".csv",
1707                 sep=""),row.names=FALSE)
1708
1709     BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(M3B_MP3RSELGP[[gp]]$Name_BestPV[,1], "___"))))
1710
1711     AddPVNamesGP <- as.vector(na.omit(M3B_MP3RSELGP[[gp]]$Name_AddPV[1]))
1712     if (gp==1&length(SignRows)!=1){
1713         AddPVNamesGP <- NULL
1714     } ## end of if (gp==1&length(SignRows)=1)
1715
1716     M3B_FinPredData_Dfrfm <- as.data.frame(DataCP[[gp]][c(BestPVNamesGP,AddPVNamesGP)])
1717     M3B_FinPredData_List <- as.list(DataCP[[gp]][c(BestPVNamesGP,AddPVNamesGP)])
1718
1719     GLoopExit<-TRUE # No significant improvement to the model
1720     } # End of if (gp>=1&length(SignRows)=1) statement -- goes out of GP Loop!
1721
1722     #####
1723     #####
1724     ##### Above is the gp loop: Grouped variables Iteration Loop
1725     #####
1726     } # Main gp loop - iterations; back up to gp loop if while loop is satisfied
1727
1728     #####
1729     ##### After best gp has been found
1730     ##### Print last significant GP
1731     ##### But only line 1 as this takes care of SignifRow=1 or 2
1732     #####
1733     #####
1734     #####
1735
1736     if(!is.na(gpbest))=FALSE
1737     {
1738         cat("\n Printing ModParTable.csv only if gpbest exists")
1739         write.csv(M3B_MP3RSELGP[[gpbest]][,1],
1740             paste(dir,M3B_CP,
1741                 if(DAC>0){DAC},
1742                 "M3B_",
1743                 "MP3RSEL",
1744                 "Fin_",
1745                 RVName,
1746                 ".csv",
1747                 sep=""),row.names=FALSE)
1748         write.csv(M3B_MP3RSELGP[[gpbest]][,1],
1749             paste(dir,M3B.res,
1750                 if(DAC>0){DAC},
1751                 "M3B_",
1752                 "MP3RSEL",
1753                 "Fin_",
1754                 RVName,
1755                 ".csv",
1756                 sep=""),row.names=FALSE)
1757
1758         write.csv({(M3B_MP3RSELGP[[gpbest]][,1]),
1759             paste(dir,M3B.res,
1760                 if(DAC>0){DAC},
1761                 "M3B_",
1762                 "MP3RSEL",
1763                 "Fin_",
1764                 RVName,
1765                 "Trans",
```

```
1766         ".csv",
1767         sep=""),row.names=TRUE)
1768     } #End of gpbest present if statement and csv printing
1769
1770     #####
1771     #####
1772     ##### M3B Saving the names of the best set of variables: ####
1773     ##### in 2 ways:
1774     ##### 1. simply the 2-level names list, where first level is the EV, second is the DV
1775     ##### 2. By creating a 2-column dataframe, which is easier to read and use further
1776     #####
1777     #####
1778     M3B_FinPredNames_List <- list()
1779     M3B_FinPredNames_Dfrfm <- data.frame()
1780
1781     ev<-lp ## lp for list position, or ev for environmental variable. Keep both for backward compatibility with other modules
1782     if(!is.na(M3B_FinPredData_Dfrfm[1][1][1]))==FALSE)
1783     {
1784         for (ev in seq(length(M3B_FinPredData_List)))
1785         {
1786             M3B_FinPredNames_List[[ev]] <- names(M3B_FinPredData_List[[ev]]) # The DV names (second level)
1787             names(M3B_FinPredNames_List)[ev] <- names(M3B_FinPredData_List[ev]) # The EV names (first level)
1788
1789             M3B_FinPred_EVName <- rep(as.vector(names(M3B_FinPredData_List[[ev]]),length(M3B_FinPredData_List[[ev]])) #
1790                                     Preparing this for a 2-column dataframe which will repeat the EV as long as needed
1791                                     M3B_FinPred_EVName <- as.vector(names(M3B_FinPredData_List[[ev]]))
1792                                     Dfrfm <- as.data.frame(cbind(M3B_FinPred_EVName,M3B_FinPred_DVName))
1793                                     M3B_FinPredNames_Dfrfm <- rbind(M3B_FinPredNames_Dfrfm,Dfrfm)
1794                                 } # End of ev loop
1795                                 rm(Dfrfm,M3B_FinPred_DVName,M3B_FinPred_EVName)
1796                             } # End of if is.na(M3B_FinPredData_Dfrfm) condition
1797
1798                             # _InsCode str(M3B_FinPredNames_Dfrfm)
1799                             # _InsCode str(M3B_FinPredNames_List)
1800
1801                             #####
1802                             ##### Adding projection lists #####
1803                             #####
1804                             #####
1805                             #####
1806                             gpbest <- length(M3B_FinPredData_List) # Final number of grouped variables - or gpbest
1807                             # _InsCode names(M3B_FinPredData_List)
1808                             M3B_Proj_FinPredData_List <- as.list(DataGPProj[[gpbest]][names(M3B_FinPredData_List)])
1809                             # _InsCode str(M3B_Proj_FinPredData_List)
1810                             # _InsCode names(M3B_FinPredData_List)
1811                             # _InsCode names(M3B_Proj_FinPredData_List)
1812
1813                             #####
1814                             #####
1815                             ##### Assigning results to rv lists #####
1816                             #####
1817                             #####
1818                             #####
1819                             DataGP_RV[[rv]] <- DataGP
1820                             GP_RV[[rv]] <- GP
1821                             M3B_MP3RGP_RV[[rv]] <- M3B_MP3RGP
1822                             M3B_MP3RSELGP_RV[[rv]] <- M3B_MP3RSELGP
1823                             # _InsCode str(M3B_FinPredData_List)
1824                             M3B_FinPredData_List_RV[[rv]] <- M3B_FinPredData_List
1825                             M3B_FinPredNames_List_RV[[rv]] <- M3B_FinPredNames_List
1826
1827                             #####
1828                             #####
```



```
1953 ModParTable$AI_Num <- al
1954 ModParTable$RV_Num <- rv
1955 ModParTable$FTCode <- substr(ModParTable$RV_Name,1,3)
1956 ModParTable$M1_Num <- m1
1957 M3B_MP3RSeIGP_1st <- rbind(M3B_MP3RSeIGP_1st,ModParTable[1,]) # Add the first ones ... with the new columns!
1958 } # End of m1
1959 rm(ModParTable,NumIter)
1960 } # End of if NA is false statement
1961 } # End of rv
1962 } # End of al
1963
1964
1965 al
1966 dim(M3B_MP3RSeIGP_1st)
1967 head(M3B_MP3RSeIGP_1st[1:3,7]) # dropping from view column 37 which is the .bat file. Not informative to view, just dogs
    the console
1968 M3B_MP3RSeIGP_1st[M3B_MP3RSeIGP_1st$RV_Num==1,1:5]
1969 M3B_MP3RSeIGP_1st[M3B_MP3RSeIGP_1st$RV_Num==2,1:5]
1970
1971 M3B_MP3RSeIGP_1st[M3B_MP3RSeIGP_1st$AI_Num==1,1:5]
1972 M3B_MP3RSeIGP_1st[M3B_MP3RSeIGP_1st$AI_Num==2,1:5]
1973
1974 #### Then from this, pick any and all "best"/final models
1975
1976 M3B_MP3RSeIGP_FinPos <- which(M3B_MP3RSeIGP_1st$CP_Best>0)
1977 M3B_MP3RSeIGP_Fin <- M3B_MP3RSeIGP_1st[M3B_MP3RSeIGP_FinPos[,1:length(M3B_MP3RSeIGP_FinPos)]]
1978 rm(M3B_MP3RSeIGP_FinPos)
1979 dim(M3B_MP3RSeIGP_Fin)
1980 head(M3B_MP3RSeIGP_Fin)
1981
1982 #####
1983 #####
1984 ##### These batch files (.bat) are extra, but useful for copying and grouping of results
1985 ##### and making them more accessible outside R ####
1986 #####
1987
1988 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxHTMLFile, ", dir.html,M3B_MP3RSeIGP_1st$RV_Name,
1989 "html", sep="")
1990 write(TmpCopyText,paste(dir,M3B,
1991 "M3B_AIIBest_",
1992 "HTML",
1993 ".bat",sep=""))
1994
1995 M3B1st_HTML.bat <- paste(dir,M3B,
1996 "M3B_AIIBest_",
1997 "HTML",
1998 ".bat",sep="")
1999 system(paste(M3B1st_HTML.bat),wait=TRUE)
2000 show.output.on.console=TRUE)
2001
2002 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxOut,"plots",", dir.html,
2003 "plots", sep="")
2004 write(TmpCopyText,paste(dir,M3B,
2005 "M3B_AIIBest_",
2006 "HTMLPlots",
2007 ".bat",sep=""))
2008 M3B1st_HTMLPlots.bat <- paste(dir,M3B,
2009 "M3B_AIIBest_",
2010 "HTMLPlots",
2011 ".bat",sep="")
2012 system(paste(M3B1st_HTMLPlots.bat),wait=TRUE,show.output.on.console=TRUE)
2013
2014 #### Gives access to the lambda file here so that it can be read via the html link already built
2015 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxLambdFile, ", dir.html,M3B_MP3RSeIGP_1st$RV_Name,
2016 "lambdas", sep="")
2017 write(TmpCopyText,paste(dir,M3B,
2018 "M3B_AIIBest_",
2019 "Lamb-HTML",
2020 ".bat",sep=""))
```

```
2021 M3B1st_LambHTML.bat <- paste(dir,M3B,
2022 "M3B_AIIBest_",
2023 "LambHTML",
2024 ".bat",sep="")
2025
2026 system(paste(M3B1st_LambHTML.bat),wait=TRUE,
2027 show.output.on.console=FALSE)
2028 #####
2029 ###
2030 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxLambdFile, ", dir.lambda,M3B_MP3RSeIGP_1st$RV_Name,
2031 "Lam.csv", sep="")
2032 write(TmpCopyText,paste(dir,M3B,
2033 "M3B_AIIBest_",
2034 "Lambda",
2035 ".bat",sep=""))
2036 M3B1st_Lambdab.bat <- paste(dir,M3B,
2037 "M3B_AIIBest_",
2038 "Lambda",
2039 ".bat",sep="")
2040 system(paste(M3B1st_Lambdab.bat ),wait=TRUE,show.output.on.console=TRUE)
2041
2042 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxPredFile, ", dir.Pred_CSV,M3B_MP3RSeIGP_1st$RV_Name,
2043 "Pred.csv", sep="")
2044 write(TmpCopyText,paste(dir,M3B,
2045 "M3B_AIIBest_",
2046 "MaxPred",
2047 ".bat",sep=""))
2048 M3B1st_MaxPred.bat <- paste(dir,M3B,
2049 "M3B_AIIBest_",
2050 "MaxPred",
2051 ".bat",sep="")
2052 system(paste(M3B1st_MaxPred.bat),wait=TRUE,show.output.on.console=TRUE)
2053
2054 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxResultsFile, ", dir.res.M3B_MP3RSeIGP_1st$RV_Name,
2055 "Res.csv", sep="")
2056 write(TmpCopyText,paste(dir,M3B,
2057 "M3B_AIIBest_",
2058 "MaxRes",
2059 ".bat",sep=""))
2060 M3B1st_MaxRes.bat <- paste(dir,M3B,
2061 "M3B_AIIBest_",
2062 "MaxRes",
2063 ".bat",sep="")
2064 system(M3B1st_MaxRes.bat,wait=TRUE,show.output.on.console=TRUE)
2065
2066 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxProjPredFile, ", dir.Proj_CSV,M3B_MP3RSeIGP_1st$RV_Name,
2067 "ProjPred.csv", sep="")
2068 write(TmpCopyText,paste(dir,M3B,
2069 "M3B_AIIBest_",
2070 "MaxProj",
2071 ".bat",sep=""))
2072 M3B1st_MaxProj.bat <- paste(dir,M3B,
2073 "M3B_AIIBest_",
2074 "MaxProj",
2075 ".bat",sep="")
2076 system(paste(M3B1st_MaxProj.bat),wait=TRUE,show.output.on.console=TRUE)
2077
2078 ## Extracing projection files. NOTE THESE ARE THE DATA FILES, not the predictions
2079 TmpCopyText <- paste("copy ", M3B_MP3RSeIGP_1st$MaxProjFile, ", dir.ProjData,M3B_MP3RSeIGP_1st$RV_Name,
2080 "Proj.csv", sep="")
2081 write(TmpCopyText,paste(dir,M3B,
2082 "M3B_AIIBest_",
2083 "MaxProjData",
2084 ".bat",sep=""))
2085 M3B1st_MaxProjData.bat <- paste(dir,M3B,
2086 "M3B_AIIBest_",
2087 "MaxProjData",
2088 ".bat",sep="")
2089 system(paste(M3B1st_MaxProjData.bat),wait=TRUE,show.output.on.console=TRUE)
```



```

2090
2091 save.image(paste(dir,rdata,
2092               if(DAC>0){DAC},
2093               ScriptNameCode,
2094               "aNI_End_All_Alphas_RV",
2095               "RData",sep=""))
2096
2097 #####
2098 #####
2099 ##### Next Script #####
2100 #####
2101 #####
2102 #####
2103 #####
2104 ##### MIA7 Module 3bx - MaxEnt Modelling Modules #####
2105 ##### Prepare the products variables #####
2106 ##### for consideration of models with interactions #####
2107 #####
2108 ##### 2015 #####
2109 ##### sabgis@yahoo.com #####
2110 ##### ##### ##### ##### ##### ##### ##### ##### #####
2111 ##### ##### ##### ##### ##### ##### ##### ##### #####
2112 #####
2113 #####
2114 ##### NOTE on terms and code abbreviations #####
2115 ##### Some of the code abbreviations refer to old terminology, however,
2116 ##### the descriptive comments use, as much as possible, the latest terms.
2117 ##### Thus, abbreviations like P, stood for "predictor", but are now simply referred to as
2118 ##### "variable", "derived variable" or "significant variable"
2119 #####
2120 # OptCode require(plyr,MIAPar$dir,Rlib) # used for joining data
2121
2122 ScriptNameCode <- "M3Bx"
2123
2124 library(scales)
2125
2126 al <-1
2127 rv<-1
2128 # Check data
2129 str(M3B_FinPredNames_List_A[[al]][1:rV])
2130 str(M3B_FinPredData_List_A[[al]][1:rV])
2131 str(M3B_Proj_FinPredData_List_A[[al]][1:rV])
2132
2133
2134 al <-1
2135 length(M3B_FinPredData_List_A)
2136 str(M3B_FinPredData_List_A[[al]])
2137
2138 M3Bx_Prod_Al <- as.list(1:length(M3B_FinPredData_List_A))
2139 names(M3Bx_Prod_Al) <- paste("Prod_Al",names(M3B_FinPredData_List_A),sep="")
2140
2141 M3Bx_Proj_Prod_Al <- as.list(1:length(M3B_Proj_FinPredData_List_A))
2142 names(M3Bx_Proj_Prod_Al) <- paste("ProjProd_Al",names(M3B_Proj_FinPredData_List_A),sep="")
2143
2144 al<-1
2145 for (al in seq(length(M3Bx_Prod_Al)))
2146 {
2147   M3Bx_Beg_Al <- (M3B_FinPredData_List_A[[al]])
2148   length(M3Bx_Beg_Al)
2149   str(M3Bx_Beg_Al)
2150   M3Bx_Proj_Beg_Al <-M3B_Proj_FinPredData_List_A
2151   cat(paste("Alpha N",al))
2152   M3Bx_Prod_RV <-list()
2153   M3Bx_Proj_Prod_RV <-list()
2154   #####
2155   rv <-1
2156   for (rv in seq(length(M3Bx_Beg_Al)))

```

```

2158 {
2159   if(length(M3Bx_Beg_Al[[rv]]<2) {cat(paste("No products to calculate for Alpha",al,"RV",rv))
2160   M3Bx_Prod_RV[[rv]] <- NA # Ensures that the list position is maintained
2161   names(M3Bx_Prod_RV)[rv] <- paste("M3Bx_Prod_RV",if(rv<10){},rv,sep="")
2162   M3Bx_Proj_Prod_RV[[rv]] <- NA # Ensures that the list position is maintained
2163   names(M3Bx_Proj_Prod_RV)[rv] <- paste("M3Bx_Proj_Prod_RV",if(rv<10){},rv,sep="")
2164 }
2165 if(length(M3Bx_Beg_Al[[rv]]>1) {
2166   cat(paste("Calculating products variables for Alpha",al,"RV",rv))
2167   M3Bx_Beg <- M3Bx_Beg_Al[[rv]]
2168   #####
2169   ##### Continuous variables only
2170   #####
2171   #####
2172   M3Bx_BegNames <- names(M3Bx_Beg)
2173   ContNames <- MIAPar$EVCntNames
2174   ContPosVect <- vector()
2175   pos<-1
2176   for(pos in seq(M3Bx_BegNames))
2177   {
2178     ContPos <- which(M3Bx_BegNames[pos]==ContNames)
2179     if(length(ContPos)>0){ContPosVect <-c(ContPosVect,pos) } # Keep track of the position in M3Bx_BegNames
2180   }
2181   M3Bx_Beg_Cont <- M3Bx_Beg[c(ContPosVect)] ## Picks out just the continuous variables, keeps the same order, drops
   categorical variables
   #_InsCode names(M3Bx_Beg_Cont)
2182   M3Bx_Proj_Beg_Cont <- M3B_Proj_FinPredData_List_A[[al]][1:rV]]names(M3Bx_Beg_Cont))
2183   #_InsCode str(M3Bx_Beg_Cont)
2184   #_InsCode str(M3Bx_Beg_Cont)
2185   #_InsCode str(M3Bx_Proj_Beg_Cont)
2186   #_InsCode names(M3Bx_Beg_Cont) # These should be the same (and in the same order) if all are continuous
2187   #_InsCode names(M3Bx_Beg)
2188
2189   EVNum <-length(M3Bx_Beg_Cont)
2190
2191   # OptCode if (EVNum>=5){EVNum<-5} ## Option for considering NOT using all EV's but say just the first 4
2192   PairEVNum <-EVNum*(EVNum-1)/2 # Pairwise combinations of all EV
2193
2194   M3Bx_Prod <- as.list(1:PairEVNum)
2195   #_InsCode str(M3Bx_Prod)
2196   M3Bx_Proj_Prod <- as.list(1:PairEVNum)
2197   Pair1 <- 1:(EVNum-1)
2198   PairPos <-1
2199   j<-Pair1[]
2200   for (j in seq(Pair1))
2201   {
2202     Pair1_Data <- as.data.frame(M3Bx_Beg_Cont[[j]]) # _InsCode str(Pair1_Data)
2203     Pair1_Len <- length(Pair1_Data)
2204     names(Pair1_Data)
2205     Pair1_DataProj <- as.data.frame(M3Bx_Proj_Beg_Cont[[j]]) # _InsCode str(Pair1_Data)
2206     k<-j+1
2207     for (k in k:EVNum)
2208     {
2209       names(M3Bx_Prod)[PairPos] <- paste(names(M3Bx_Beg_Cont[[j]]),"X",names(M3Bx_Beg_Cont[k]),sep="")
2210       names(M3Bx_Proj_Prod)[PairPos] <- paste(names(M3Bx_Proj_Beg_Cont[[j]]),"X",names(M3Bx_Beg_Cont[k]),sep="")
2211       Pair2_Data <- as.data.frame(M3Bx_Beg_Cont[[k]])
2212       Pair2_Len <- length(Pair2_Data)
2213       Pair2_DataProj <- as.data.frame(M3Bx_Proj_Beg_Cont[[k]])
2214       ### Inner loops
2215       j<-1
2216       k<-1
2217
2218       ProdCol <- ""
2219       ProdColProj<-""
2220
2221       for (j in seq(Pair1_Len))
2222       {
2223         for (k in seq(Pair2_Len))
2224         {
2225           Prod <- Pair1_Data[j] * Pair2_Data[k] # First position - first column
2226           DVName1 <- (strsplit(names(Pair1_Data[[j]]),"_"))

```

```
2227 DVName2 <- {rtrsplit(names(Pair2_Data[k]), "_")}
2228 names(Prod) <- paste(
2229   "EV", if (P_Pos < 10) {"0"}, J,      # Option 2 naming for EV
2230   "xEV", if (P_Pos < 10) {"0"}, K,
2231   "J", "x", "k", # option 1
2232   sep="") # Take the names for the group rather than for individual DVs and adds product suffix
2233
2234 ProdCol <- cbind(ProdCol, Prod)
2235 ##### Projection products
2236 names(ProdProj) <- Pair1_DataProj[[ * Pair2_DataProj[k] # First position - first column
2237   names(ProdProj) <- names(Prod) #Get the same names as from the Prod above
2238 ProdColProj <- cbind(ProdColProj, ProdProj)
2239 } # k loop
2240 } # j loop
2241
2242 ProdData <- ProdCol[-1]
2243 ProdLin <- rescale(ProdData, c(0,1))
2244 ProdDataProj <- ProdColProj[-1]
2245 ProdLinProj <- rescale(ProdDataProj, c(0,1))
2246 M3Bx_Prod[[P_Pos]] <- ProdLin
2247 M3Bx_Proj_Prod[[P_Pos]] <- ProdLinProj
2248 P_Pos <- P_Pos+1
2249 } # k loop
2250 } # j loop
2251 M3Bx_Prod_RV[[rv]] <- M3Bx_Prod
2252 names(M3Bx_Prod_RV)[rv] <- paste("M3BxProd_RV", if(rv<10){0}, rv, sep="")
2253 M3Bx_Proj_Prod_RV[[rv]] <- M3Bx_Proj_Prod
2254 names(M3Bx_Proj_Prod_RV)[rv] <- paste("M3Bx_Proj_Prod_RV", if(rv<10){0}, rv, sep="")
2255 } # End of if else statement re calculating products
2256 } ## RV loop
2257 M3Bx_Prod_A[[all]] <- M3Bx_Prod_RV
2258 M3Bx_Proj_Prod_A[[all]] <- M3Bx_Proj_Prod_RV
2259 names(M3Bx_Proj_Prod_A)[all] <- paste("M3Bx_Proj_Prod_A", if(rv<10){0}, rv, sep="")
2260 } ## All Loop
2261
2262 str(M3Bx_Prod_A[[all]])
2263 str(M3Bx_Proj_Prod_A[[all]])
2264
2265 names(M3Bx_Prod_A[[all]])
2266 names(M3Bx_Prod_A[[all]])
2267 names(M3Bx_Prod_A[[all]])
2268 names(M3Bx_Prod_A[[all]])
2269 names(M3Bx_Proj_Prod_A[[all]])
2270 # Projection files should be the same as above
2271 names(M3Bx_Prod_A[[all]][rv][1])
2272 names(M3Bx_Proj_Prod_A[[all]][rv][1]) # Projection files should be the same as above
2273 str(M3Bx_Prod_A[[all]][rv][1])
2274 str(M3Bx_Proj_Prod_A[[all]][rv][1]) # Projection files should be the same as above; only dataframe bigger
2275
2276 names(M3Bx_Prod_A[[all]][rv][1])
2277
2278 names(M3Bx_Prod_A[[all]][rv][1][1])
2279
2280 str(M3Bx_Prod_A[[all]])
2281 str(M3Bx_Proj_Prod_A[[all]]) # Projection files should be the same as above; only dataframe bigger
2282
2283 rm(Pair1, Pair1_Data, Pair1_Len)
2284 rm(Pair2_Data, Pair2_Len)
2285 rm(Pair_EVNum, ProdCol, ProdData)
2286 rm(Prod, ProdLin, P_Pos)
2287 rm(DVName1, DVName2)
2288 rm(M3Bx_Beg, M3Bx_Beg_Cont, M3Bx_Beg_AJ, M3Bx_BegNames, M3Bx_Prod, M3Bx_Prod_RV)
2289 rm(Pair1_DataProj, Pair2_DataProj, ProdColProj, ProdDataProj, ProdProj, ProdLinProj)
2290 rm(M3Bx_Proj_Beg_Cont, M3Bx_Proj_Prod, M3Bx_Proj_Prod_RV, M3Bx_Proj_Beg_AJ)
2291 rm(J, K, k, ContPos, ContPosVect, ContNames)
2292 str(M3Bx_Prod_AJ)
2293 str(M3Bx_Proj_Prod_AJ)
2294
2295 save.image(paste(dir, rdata,
2296   if(DAC>0){DAC},
```

```
2297 ScriptNameCode, "_",
2298   "All_Prod_Ready", ".RData", sep="")
2299 #####
2300 ##### Next script #####
2301 #####
2302 #####
2303 #####
2304 #####
2305 #####
2306 #####
2307 #####
2308 #####
2309 #####
2310 #####
2311 #####
2312 #####
2313 #####
2314 #####
2315 #####
2316 #####
2317 #####
2318 #####
2319 #####
2320 #####
2321 #####
2322 #####
2323 #####
2324 #####
2325 #####
2326 #####
2327 #####
2328 # OptCode require(plyr, MIAPar$dir, Rlib) # used for joining data
2329
2330 #load("D:/IDA/A01_T4/RData/A01_M3A_End_All.RData") # Choose this if no RData
2331 #load("D:/IDA/A01_T4/RData/A01_M3A_End_SmallerDataset2Test.RData")
2332 #load("D:/IDA/A02_55_John/RData/A02_M3Bx_REDUCED_5EV_All_Prod_Ready.RData")
2333
2334 ScriptNameCode <- "M3C"
2335 SaveTempFlag <- FALSE
2336 # As in this section we are running maxent.jar, place this check and instruct use
2337
2338 JarFileFlag <- paste(dir.in, "maxent.jar", sep="")
2339 # Checking if maxent.jar can be run (normally not necessary as part of the script)
2340 if (file.exists(JarFileFlag) == FALSE){
2341   cat("!!!! WARNING !!!! \nPlease place a copy of maxent.jar in your dir.in directory
2342     !!!! WARNING !!!!) else {"All's good, carry on, carry on"}
2343 }
2344 ## Remind users of their maxent.jar settings chosen in Module 1
2345 JarPar$M3C_JarFlags
2346
2347 #####
2348 #####
2349 #####
2350 #####
2351 #####
2352 #####
2353 #####
2354 al<-1
2355 rv<-1
2356 # InsCode str(M3Bx_Prod_AJ) # Necessary data
2357 names(M3Bx_Prod_AJ) # Necessary data)
2358 names(M3Bx_Prod_AJ)[all][rv][1] # Necessary data
2359 str(M3Bx_Prod_AJ)[all][rv][1] # Necessary data
2360 str(M3Bx_Proj_Prod_AJ)[all][rv][1] # ## Equivalent projections
2361 names(M3Bx_Proj_Prod_AJ)[all][rv][1]
2362 names(M3B_FinPredData_List_AJ)
2363 names(M3B_FinPredData_List_AJ)[all][rv][1]
2364 str(M3B_FinPredData_List_AJ)[all][rv][1] # Necessary data
```

```

2365 str(M3B_Proj_FinPredData_List_A[[r]][rv]) # Necessary data
2366 #####
2367 #####
2368 #####
2369 #####
2370 #####
2371 #####
2372 #####
2373 dir.M3C <- paste(dir.max,
2374 "M3C_All", # Main directory
2375 "\", sep="")
2376 dir.M3C.res <- paste(dir.M3C,
2377 "M3C_AllRes", # Key results grouped together
2378 "\", sep="")
2379 dir.create(dir.M3C)
2380 dir.create(dir.M3C.res)
2381 dir.html <- paste(dir.M3C.res,
2382 "HTML",
2383 "\", sep="")
2384 dir.create(dir.html)
2385 dir.htmlplots <- paste(dir.html,
2386 "plots",
2387 "\", sep="")
2388 dir.create(dir.htmlplots)
2389 dir.lamdba <- paste(dir.M3C.res,
2390 "Lamdba",
2391 "\", sep="")
2392 dir.create(dir.lamdba)
2393 dir.Pred_CSV <- paste(dir.M3C.res,
2394 "Pred_CSV",
2395 "\", sep="")
2396 dir.create(dir.Pred_CSV)
2397 dir.res <- paste(dir.M3C.res,
2398 "Res_Max",
2399 "\", sep="")
2400 dir.create(dir.res)
2401 dir.Proj_CSV <- paste(dir.M3C.res, # Adding projected prediction file
2402 "Proj_CSV",
2403 "\", sep="")
2404 dir.create(dir.Proj_CSV)
2405 dir.ProjData_SWD <- paste(dir.M3C.res, # Adding projection data file
2406 "ProjData_SWD",
2407 "\", sep="")
2408 dir.create(dir.ProjData_SWD)
2409 #####
2410 #####
2411 #####
2412 #####
2413 #####
2414 #####
2415 AlphaValues <- MIAPar$AlphaValues
2416 RegulValues <- MIAPar$RegulValues
2417 rz <- 1
2418
2419 M3C_MP3RGP_Al <- as.list(length(M3Bx_Prod_All)) # Al which will hold all the RV
2420 names(M3C_MP3RGP_Al) <- paste("M3C_MP3RGP_Al",length(AlphaValues),sep="")
2421
2422 M3C_MP3RSeIGP_Al <- as.list(length(M3Bx_Prod_All))
2423 names(M3C_MP3RSeIGP_Al) <- paste("M3C_MP3RSeIGP_Al",length(AlphaValues),sep="")
2424
2425 M3C_FinPredData_List_Al <- as.list(length(M3Bx_Prod_All))
2426 names(M3C_FinPredData_List_Al) <- paste("M3C_FinPredData_List_Al",length(AlphaValues),sep="")
2427
2428 M3C_FinPredNames_List_Al <- as.list(length(M3Bx_Prod_All))
2429 names(M3C_FinPredNames_List_Al) <- paste("M3C_FinPredNames_List_Al",length(AlphaValues),sep="")
2430
2431 M3C_FinPredNames_Dfrm_Al <- as.list(length(M3Bx_Prod_All))
2432 names(M3C_FinPredNames_Dfrm_Al) <- paste("M3C_FinPredNames_Dfrm_Al",length(AlphaValues),sep="")
2433
2434 M3C_FinPredData_Dfrm_Al <- as.list(length(M3Bx_Prod_All))

```

```

2435 names(M3C_FinPredData_Dfrm_Al) <- paste("M3C_FinPredData_Dfrm_Al",length(AlphaValues),sep="")
2436
2437 M3C_Proj_FinPredData_List_Al <- as.list(length(M3Bx_Proj_Prod_All))
2438 names(M3C_Proj_FinPredData_List_Al) <- paste("M3C_Proj_FinPredData_List_Al",length(AlphaValues),sep="")
2439
2440 #####
2441 #####
2442 #####
2443 #####
2444 #####
2445
2446 rv<-1
2447 al <- 1 # Decide where to start
2448 ev<-1
2449
2450 for (al in seq(length(M3Bx_Prod_All)))
2451 {
2452   M3Bx_Prod_RV <- M3Bx_Prod_All[[al]]
2453   #####
2454   #####
2455   #####
2456   #####
2457   DataGP_RV <- as.list(length(M3Bx_Prod_RV))
2458   names(DataGP_RV) <- paste("DataGP_RV",length(RVNames),sep="")
2459
2460   GP_RV <- as.list(length(M3Bx_Prod_RV))
2461   names(GP_RV) <- paste("GP_RV",length(RVNames),sep="")
2462
2463   M3C_MP3RGP_RV <- as.list(length(M3Bx_Prod_RV))
2464   names(M3C_MP3RGP_RV) <- paste("M3C_MP3RGP_RV",length(RVNames),sep="")
2465
2466   M3C_MP3RSeIGP_RV <- as.list(length(M3Bx_Prod_RV))
2467   names(M3C_MP3RSeIGP_RV) <- paste("M3C_MP3RSeIGP_RV",length(RVNames),sep="")
2468
2469   M3C_FinPredData_List_RV <- as.list(length(M3Bx_Prod_RV)) # As list but also duplicated as printable dataframe in
2470   M3C_FinPred_Data_All_RV (it's just for the column name)
2471   names(M3C_FinPredData_List_RV) <- paste("M3C_FinPredData_List_RV",length(RVNames),sep="")
2472
2473   M3C_FinPredNames_List_RV <- as.list(length(M3Bx_Prod_RV))
2474   names(M3C_FinPredNames_List_RV) <- paste("M3C_FinPredNames_List_RV",length(RVNames),sep="")
2475
2476   M3C_FinPredData_Dfrm_RV <- as.list(length(M3Bx_Prod_RV)) # As dataframe duplicate of the list above but better for
2477   printing in column names
2478   names(M3C_FinPredData_Dfrm_RV) <- paste("M3C_FinPredData_Dfrm_RV",length(RVNames),sep="")
2479
2480   M3C_FinPredNames_Dfrm_RV <- as.list(length(M3Bx_Prod_RV))
2481   names(M3C_FinPredNames_Dfrm_RV) <- paste("M3C_FinPredNames_Dfrm_RV",length(RVNames),sep="")
2482
2483   M3C_Proj_FinPredData_List_RV <- as.list(length(M3Bx_Prod_RV)) # Extracting length only, not the values
2484   names(M3C_Proj_FinPredData_List_RV) <- paste("M3C_Proj_FinPredData_List_RV",length(RVNames),sep="")
2485   #####
2486   #####
2487   #####
2488   #####
2489   #####
2490   rv<-1 # Can decide where to start
2491   for (rv in seq(length(M3Bx_Prod_RV)))
2492   {
2493     if(length(M3Bx_Prod_RV[rv])==0){ #This is in case there are no variables at all
2494       ca(paste("No significant variables picked at all for Alpha ", al, "RV", rv))
2495     }
2496     Assign even if no significant variables found
2497     DataGP_RV[rv] <- NA #DataGP
2498     GP_RV[rv] <- NA # GP
2499     M3C_MP3RGP_RV[rv] <- NA # Ensures that the list position is maintained
2500     M3C_MP3RSeIGP_RV[rv] <- NA # Ensures that the list position is maintained
2501   }

```

```

2501 #_InCode str(M3C_FinPredData_List)
2502 M3C_FinPredData_List_RV[rv] <- NA # Ensures that the list position is maintained
2503 M3C_FinPredNames_List_RV[rv] <- NA # Ensures that the list position is maintained
2504 M3C_FinPredNames_Dfrm_RV[rv] <- NA # Ensures that the list position is maintained
2505 M3C_FinPredData_Dfrm_RV[rv] <- NA # Ensures that the list position is maintained
2506 M3C_Proj_FinPredData_List_RV[rv] <- NA
2507
2508 } else {## There are significant variables so carry on
2509 ##### NOTE - Key Difference between the M3B and M3C scripts -- RVXY in M3C will contain M3B significant
2510 variables as well. Thus count refers to interactions significant variables only
2511 RVXY <- cbind(M3_N_SWD_RV[rv][1:2],
2512               M3B_FinPredData_List_A[all][rv]) ## Add NI variables here, not inside "GP list" so count of additional
2513               significant variables refers to interactions
2514 RVName <- paste("aVI", "RV", if(rv<10){}, rv,"A", al, sep="")
2515 P_ValueCutoff <- AlphaValues[al] # Alpha value, can be set differently to obtain a less or more stringent test
2516 dir_M3C_GP <- paste(dir_M3C,
2517                    "M3C_",
2518                    RVName,
2519                    "\\.",
2520                    sep="")
2521 dir.create(dir_M3C_GP)
2522
2523 M3Bx_Prod <- M3Bx_Prod_RV[rv]
2524 M3Bx_Proj_Prod <- M3Bx_Proj_Prod_A[all][rv] #
2525
2526 for(lp in seq(length(M3Bx_Prod))) # This is only for Null EVs
2527 {
2528   if(is.na(M3Bx_Prod[lp])==TRUE){M3Bx_Prod[lp]<-NULL
2529     M3Bx_Proj_Prod[lp]<-NULL
2530   }
2531   # End lp ###
2532
2533 SWDList <- M3Bx_Prod
2534 SWDList_Proj <- M3Bx_Proj_Prod_A[all][rv]
2535 RVXY_Proj <- cbind(M1_N_SWD_RV[rv][2:4], # This is the RVXY for all in Module 3
2536                   M3B_Proj_FinPredData_List_A[all][rv]) ## add W1 significant variables, but not inside "GP list"
2537
2538 cat(paste("\n", "Module 3C Checking", length(SWDList), "grouped variables for rv n.", if(rv<10){}, rv, RVName, "Alpha", al,
2539          P_ValueCutoff))
2540
2541 #####
2542 #####
2543 #####
2544 #####
2545 #####
2546 #####
2547 #####
2548 #####
2549 #####
2550 #####
2551 DataGP[rv] <- as.list(SWDList)
2552 DataGPProj[rv] <- as.list(SWDList_Proj)
2553 #_InCode str(DataGP[rv])
2554 #_InCode str(DataGPProj[rv])
2555
2556 #####
2557 #####
2558 #####
2559 #####
2560 #####
2561 #####
2562 #####
2563 #####
2564 #####
2565 #####
2566 #####
2567 #####
2568 #####
2569 #####
2570 #####
2571 #####
2572 #####
2573 #####
2574 #####
2575 #####
2576 #####
2577 #####
2578 #####
2579 #####
2580 #####
2581 #####
2582 #####
2583 #####
2584 #####
2585 #####
2586 #####
2587 #####
2588 #####
2589 #####
2590 #####
2591 #####
2592 #####
2593 #####
2594 #####
2595 #####
2596 #####
2597 #####
2598 #####
2599 #####
2600 #####
2601 #####
2602 #####
2603 #####
2604 #####
2605 #####
2606 #####
2607 #####
2608 #####
2609 #####
2610 #####
2611 #####
2612 #####
2613 #####
2614 #####
2615 #####
2616 #####
2617 #####
2618 #####
2619 #####
2620 #####
2621 #####
2622 #####
2623 #####
2624 #####
2625 #####

```

```

2560 names(DataGP) <- paste("M3C", "_GP", "0", 1, sep="") # This is the first round
2561 names(DataGPProj) <- paste("M3C", "_GPProj", "0", 1, sep=")
2562 #####
2563 #####
2564 #####
2565 #####
2566 ##### Preparing a dataframe in which all key information from each MaxEnt run is stored.
2567 ##### This will be used in many ways.
2568 ##### It will be saved as a list of dataframes
2569 ##### depending on how many levels are considered
2570 #####
2571 ##### MaxResNames #####
2572 MaxResNames <- as.vector(c(
2573   "GP_Name", # Grouped variable
2574   "GP_Num", # Number of grouped variables
2575   "PV_NameGP", # Individual variables used, separated by spaces
2576   "RV_Name",
2577   "GP_Best",
2578   "NPred_Used",
2579   "NPred_Test",
2580   "MaxOut",
2581   "MaxHTMLFile",
2582   "MaxPredFile",
2583   "MaxLambdaFile",
2584   "MaxResultsFile",
2585   "MaxRes_E",
2586   "MaxRes_n",
2587   "MaxRes_N",
2588   "MaxRes_AUC",
2589   "FTVE_PV",
2590   "FTVE_PrevPV",
2591   "FTVE_AddPV",
2592   "dftuPV",
2593   "dftPV",
2594   "FRatioTestPV",
2595   "P_ValuePV", # This will be used as the "previous" in the next round
2596   "P_Value_BestPV",
2597   "Name_BestPV",
2598   "Name_AddPV",
2599   "Sign_RankPV",
2600   "Sign_PVNamePV",
2601   "AlphaValue",
2602   "Regulariz",
2603   "MaxProjFile",
2604   "MaxProjPredFile", # -- The projected prediction
2605   "MaxBatFile"
2606 ))
2607
2608 #####
2609 #####
2610 #####
2611 ##### Pick up from here after you've selected the best and most significant Grouped variables,
2612 ##### (i.e., gp loop)
2613 #####
2614 #####
2615 #####
2616 #####
2617 ##### gp loop - iterations
2618 #####
2619 BestPVNamesGP <- NULL
2620 AddPVNamesGP <- NULL
2621 GLoopExit<-FALSE
2622 while (GLoopExit==FALSE)
2623 {
2624   RVName <- paste("a", "VI", "RV", if(rv<10){}, rv,"A", al, sep="")
2625
2626 #####
2627 #####
2628 #####
2629 #####
2630 #####
2631 #####
2632 #####
2633 #####
2634 #####
2635 #####
2636 #####
2637 #####
2638 #####
2639 #####
2640 #####
2641 #####
2642 #####
2643 #####
2644 #####
2645 #####
2646 #####
2647 #####
2648 #####
2649 #####
2650 #####
2651 #####
2652 #####
2653 #####
2654 #####
2655 #####
2656 #####
2657 #####
2658 #####
2659 #####
2660 #####
2661 #####
2662 #####
2663 #####
2664 #####
2665 #####
2666 #####
2667 #####
2668 #####
2669 #####
2670 #####
2671 #####
2672 #####
2673 #####
2674 #####
2675 #####
2676 #####
2677 #####
2678 #####
2679 #####
2680 #####
2681 #####
2682 #####
2683 #####
2684 #####
2685 #####
2686 #####
2687 #####
2688 #####
2689 #####
2690 #####
2691 #####
2692 #####
2693 #####
2694 #####
2695 #####
2696 #####
2697 #####
2698 #####
2699 #####
2700 #####
2701 #####
2702 #####
2703 #####
2704 #####
2705 #####
2706 #####
2707 #####
2708 #####
2709 #####
2710 #####
2711 #####
2712 #####
2713 #####
2714 #####
2715 #####
2716 #####
2717 #####
2718 #####
2719 #####
2720 #####
2721 #####
2722 #####
2723 #####
2724 #####
2725 #####
2726 #####
2727 #####
2728 #####
2729 #####
2730 #####
2731 #####
2732 #####
2733 #####
2734 #####
2735 #####
2736 #####
2737 #####
2738 #####
2739 #####
2740 #####
2741 #####
2742 #####
2743 #####
2744 #####
2745 #####
2746 #####
2747 #####
2748 #####
2749 #####
2750 #####
2751 #####
2752 #####
2753 #####
2754 #####
2755 #####
2756 #####
2757 #####
2758 #####
2759 #####
2760 #####
2761 #####
2762 #####
2763 #####
2764 #####
2765 #####
2766 #####
2767 #####
2768 #####
2769 #####
2770 #####
2771 #####
2772 #####
2773 #####
2774 #####
2775 #####
2776 #####
2777 #####
2778 #####
2779 #####
2780 #####
2781 #####
2782 #####
2783 #####
2784 #####
2785 #####
2786 #####
2787 #####
2788 #####
2789 #####
2790 #####
2791 #####
2792 #####
2793 #####
2794 #####
2795 #####
2796 #####
2797 #####
2798 #####
2799 #####
2800 #####
2801 #####
2802 #####
2803 #####
2804 #####
2805 #####
2806 #####
2807 #####
2808 #####
2809 #####
2810 #####
2811 #####
2812 #####
2813 #####
2814 #####
2815 #####
2816 #####
2817 #####
2818 #####
2819 #####
2820 #####
2821 #####
2822 #####
2823 #####
2824 #####
2825 #####
2826 #####
2827 #####
2828 #####
2829 #####
2830 #####
2831 #####
2832 #####
2833 #####
2834 #####
2835 #####
2836 #####
2837 #####
2838 #####
2839 #####
2840 #####
2841 #####
2842 #####
2843 #####
2844 #####
2845 #####
2846 #####
2847 #####
2848 #####
2849 #####
2850 #####
2851 #####
2852 #####
2853 #####
2854 #####
2855 #####
2856 #####
2857 #####
2858 #####
2859 #####
2860 #####
2861 #####
2862 #####
2863 #####
2864 #####
2865 #####
2866 #####
2867 #####
2868 #####
2869 #####
2870 #####
2871 #####
2872 #####
2873 #####
2874 #####
2875 #####
2876 #####
2877 #####
2878 #####
2879 #####
2880 #####
2881 #####
2882 #####
2883 #####
2884 #####
2885 #####
2886 #####
2887 #####
2888 #####
2889 #####
2890 #####
2891 #####
2892 #####
2893 #####
2894 #####
2895 #####
2896 #####
2897 #####
2898 #####
2899 #####
2900 #####
2901 #####
2902 #####
2903 #####
2904 #####
2905 #####
2906 #####
2907 #####
2908 #####
2909 #####
2910 #####
2911 #####
2912 #####
2913 #####
2914 #####
2915 #####
2916 #####
2917 #####
2918 #####
2919 #####
2920 #####
2921 #####
2922 #####
2923 #####
2924 #####
2925 #####
2926 #####
2927 #####
2928 #####
2929 #####
2930 #####
2931 #####
2932 #####
2933 #####
2934 #####
2935 #####
2936 #####
2937 #####
2938 #####
2939 #####
2940 #####
2941 #####
2942 #####
2943 #####
2944 #####
2945 #####
2946 #####
2947 #####
2948 #####
2949 #####
2950 #####
2951 #####
2952 #####
2953 #####
2954 #####
2955 #####
2956 #####
2957 #####
2958 #####
2959 #####
2960 #####
2961 #####
2962 #####
2963 #####
2964 #####
2965 #####
2966 #####
2967 #####
2968 #####
2969 #####
2970 #####
2971 #####
2972 #####
2973 #####
2974 #####
2975 #####
2976 #####
2977 #####
2978 #####
2979 #####
2980 #####
2981 #####
2982 #####
2983 #####
2984 #####
2985 #####
2986 #####
2987 #####
2988 #####
2989 #####
2990 #####
2991 #####
2992 #####
2993 #####
2994 #####
2995 #####
2996 #####
2997 #####
2998 #####
2999 #####
3000 #####
3001 #####
3002 #####
3003 #####
3004 #####
3005 #####
3006 #####
3007 #####
3008 #####
3009 #####
3010 #####
3011 #####
3012 #####
3013 #####
3014 #####
3015 #####
3016 #####
3017 #####
3018 #####
3019 #####
3020 #####
3021 #####
3022 #####
3023 #####
3024 #####
3025 #####
3026 #####
3027 #####
3028 #####
3029 #####
3030 #####
3031 #####
3032 #####
3033 #####
3034 #####
3035 #####
3036 #####
3037 #####
3038 #####
3039 #####
3040 #####
3041 #####
3042 #####
3043 #####
3044 #####
3045 #####
3046 #####
3047 #####
3048 #####
3049 #####
3050 #####
3051 #####
3052 #####
3053 #####
3054 #####
3055 #####
3056 #####
3057 #####
3058 #####
3059 #####
3060 #####
3061 #####
3062 #####
3063 #####
3064 #####
3065 #####
3066 #####
3067 #####
3068 #####
3069 #####
3070 #####
3071 #####
3072 #####
3073 #####
3074 #####
3075 #####
3076 #####
3077 #####
3078 #####
3079 #####
3080 #####
3081 #####
3082 #####
3083 #####
3084 #####
3085 #####
3086 #####
3087 #####
3088 #####
3089 #####
3090 #####
3091 #####
3092 #####
3093 #####
3094 #####
3095 #####
3096 #####
3097 #####
3098 #####
3099 #####
3100 #####
3101 #####
3102 #####
3103 #####
3104 #####
3105 #####
3106 #####
3107 #####
3108 #####
3109 #####
3110 #####
3111 #####
3112 #####
3113 #####
3114 #####
3115 #####
3116 #####
3117 #####
3118 #####
3119 #####
3120 #####
3121 #####
3122 #####
3123 #####
3124 #####
3125 #####
3126 #####
3127 #####
3128 #####
3129 #####
3130 #####
3131 #####
3132 #####
3133 #####
3134 #####
3135 #####
3136 #####
3137 #####
3138 #####
3139 #####
3140 #####
3141 #####
3142 #####
3143 #####
3144 #####
3145 #####
3146 #####
3147 #####
3148 #####
3149 #####
3150 #####
3151 #####
3152 #####
3153 #####
3154 #####
3155 #####
3156 #####
3157 #####
3158 #####
3159 #####
3160 #####
3161 #####
3162 #####
3163 #####
3164 #####
3165 #####
3166 #####
3167 #####
3168 #####
3169 #####
3170 #####
3171 #####
3172 #####
3173 #####
3174 #####
3175 #####
3176 #####
3177 #####
3178 #####
3179 #####
3180 #####
3181 #####
3182 #####
3183 #####
3184 #####
3185 #####
3186 #####
3187 #####
3188 #####
3189 #####
3190 #####
3191 #####
3192 #####
3193 #####
3194 #####
3195 #####
3196 #####
3197 #####
3198 #####
3199 #####
3200 #####
3201 #####
3202 #####
3203 #####
3204 #####
3205 #####
3206 #####
3207 #####
3208 #####
3209 #####
3210 #####
3211 #####
3212 #####
3213 #####
3214 #####
3215 #####
3216 #####
3217 #####
3218 #####
3219 #####
3220 #####
3221 #####
3222 #####
3223 #####
3224 #####
3225 #####
3226 #####
3227 #####
3228 #####
3229 #####
3230 #####
3231 #####
3232 #####
3233 #####
3234 #####
3235 #####
3236 #####
3237 #####
3238 #####
3239 #####
3240 #####
3241 #####
3242 #####
3243 #####
3244 #####
3245 #####
3246 #####
3247 #####
3248 #####
3249 #####
3250 #####
3251 #####
3252 #####
3253 #####
3254 #####
3255 #####
3256 #####
3257 #####
3258 #####
3259 #####
3260 #####
3261 #####
3262 #####
3263 #####
3264 #####
3265 #####
3266 #####
3267 #####
3268 #####
3269 #####
3270 #####
3271 #####
3272 #####
3273 #####
3274 #####
3275 #####
3276 #####
3277 #####
3278 #####
3279 #####
3280 #####
3281 #####
3282 #####
3283 #####
3284 #####
3285 #####
3286 #####
3287 #####
3288 #####
3289 #####
3290 #####
3291 #####
3292 #####
3293 #####
3294 #####
3295 #####
3296 #####
3297 #####
3298 #####
3299 #####
3300 #####
3301 #####
3302 #####
3303 #####
3304 #####
3305 #####
3306 #####
3307 #####
3308 #####
3309 #####
3310 #####
3311 #####
3312 #####
3313 #####
3314 #####
3315 #####
3316 #####
3317 #####
3318 #####
3319 #####
3320 #####
3321 #####
3322 #####
3323 #####
3324 #####
3325 #####
3326 #####
3327 #####
3328 #####
3329 #####
3330 #####
3331 #####
3332 #####
3333 #####
3334 #####
3335 #####
3336 #####
3337 #####
3338 #####
3339 #####
3340 #####
3341 #####
3342 #####
3343 #####
3344 #####
3345 #####
3346 #####
3347 #####
3348 #####
3349 #####
3350 #####
3351 #####
3352 #####
3353 #####
3354 #####
3355 #####
3356 #####
3357 #####
3358 #####
3359 #####
3360 #####
3361 #####
3362 #####
3363 #####
3364 #####
3365 #####
3366 #####
3367 #####
3368 #####
3369 #####
3370 #####
3371 #####
3372 #####
3373 #####
3374 #####
3375 #####
3376 #####
3377 #####
3378 #####
3379 #####
3380 #####
3381 #####
3382 #####
3383 #####
3384 #####
3385 #####
3386 #####
3387 #####
3388 #####
3389 #####
3390 #####
3391 #####
3392 #####
3393 #####
3394 #####
3395 #####
3396 #####
3397 #####
3398 #####
3399 #####
3400 #####
3401 #####
3402 #####
3403 #####
3404 #####
3405 #####
3406 #####
3407 #####
3408 #####
3409 #####
3410 #####
3411 #####
3412 #####
3413 #####
3414 #####
3415 #####
3416 #####
3417 #####
3418 #####
3419 #####
3420 #####
3421 #####
3422 #####
3423 #####
3424 #####
3425 #####
3426 #####
3427 #####
3428 #####
3429 #####
3430 #####
3431 #####
3432 #####
3433 #####
3434 #####
3435 #####
3436 #####
3437 #####
3438 #####
3439 #####
3440 #####
3441 #####
3442 #####
3443 #####
3444 #####
3445 #####
3446 #####
3447 #####
3448 #####
3449 #####
3450 #####
3451 #####
3452 #####
3453 #####
3454 #####
3455 #####
3456 #####
3457 #####
3458 #####
3459 #####
3460 #####
3461 #####
3462 #####
3463 #####
3464 #####
3465 #####
3466 #####
3467 #####
3468 #####
3469 #####
3470 #####
3471 #####
3472 #####
3473 #####
3474 #####
3475 #####
3476 #####
3477 #####
3478 #####
3479 #####
3480 #####
3481 #####
3482 #####
3483 #####
3484 #####
3485 #####
3486 #####
3487 #####
3488 #####
3489 #####
3490 #####
3491 #####
3492 #####
3493 #####
3494 #####
3495 #####
3496 #####
3497 #####
3498 #####
3499 #####
3500 #####
3501 #####
3502 #####
3503 #####
3504 #####
3505 #####
3506 #####
3507 #####
3508 #####
3509 #####
3510 #####
3511 #####
3512 #####
3513 #####
3514 #####
3515 #####
3516 #####
3517 #####
3518 #####
3519 #####
3520 #####
3521 #####
3522 #####
3523 #####
3524 #####
3525 #####
3526 #####
3527 #####
3528 #####
3529 #####
3530 #####
3531 #####
3532 #####
3533 #####
3534 #####
3535 #####
3536 #####
3537 #####
3538 #####
3539 #####
3540 #####
3541 #####
3542 #####
3543 #####
3544 #####
3545 #####
3546 #####
3547 #####
3548 #####
3549 #####
3550 #####
3551 #####
3552 #####
3553 #####
3554 #####
3555 #####
3556 #####
3557 #####
3558 #####
3559 #####
3560 #####
3561 #####
3562 #####
3563 #####
3564 #####
3565 #####
3566 #####
3567 #####
3568 #####
3569 #####
3570 #####
3571 #####
3572 #####
3573 #####
3574 #####
3575 #####
3576 #####
3577 #####
3578 #####
3579 #####
3580 #####
3581 #####
3582 #####
3583 #####
3584 #####
3585 #####
3586 #####
3587 #####
3588 #####
3589 #####
3590 #####
3591 #####
3592 #####
3593 #####
3594 #####
3595 #####
3596 #####
3597 #####
3598 #####
3599 #####
3600 #####
3601 #####
3602 #####
3603 #####
3604 #####
3605 #####
3606 #####
3607 #####
3608 #####
3609 #####
3610 #####
3611 #####
3612 #####
3613 #####
3614 #####
3615 #####
3616 #####
3617 #####
3618 #####
3619 #####
3620 #####
3621 #####
3622 #####
3623 #####
3624 #####
3625 #####
3626 #####
3627 #####
3628 #####
3629 #####
3630 #####
3631 #####
3632 #####
3633 #####
3634 #####
3635 #####
3636 #####
3637 #####
3638 #####
3639 #####
3640 #####
3641 #####
3642 #####
3643 #####
3644 #####
3645 #####
3646 #####
3647 #####
3648 #####
3649 #####
3650 #####
3651 #####
3652 #####
3653 #####
3654 #####
3655 #####
3656 #####
3657 #####
3658 #####
3659 #####
3660 #####
3661 #####
3662 #####
3663 #####
3664 #####
3665 #####
3666 #####
3667 #####
3668 #####
3669 #####
3670 #####
3671 #####
3672 #####
3673 #####
3674 #####
3675 #####
3676 #####
3677 #####
3678 #####
3679 #####
3680 #####
3681 #####
3682 #####
3683 #####
3684 #####
3685 #####
3686 #####
3687 #####
3688 #####
3689 #####
3690 #####
3691 #####
3692 #####
3693 #####
3694 #####
3695 #####
3696 #####
3697 #####
3698 #####
3699 #####
3700 #####
3701 #####
3702 #####
3703 #####
3704 #####
3705 #####
3706 #####
3707 #####
3708 #####
3709 #####
3710 #####
3711 #####
3712 #####
3713 #####
3714 #####
3715 #####
3716 #####
3717 #####
3718 #####
3719 #####
3720 #####
3721 #####
3722 #####
3723 #####
3724 #####
3725 #####
3726 #####
3727 #####
3728 #####
3729 #####
3730 #####
3731 #####
3732 #####
3733 #####
3734 #####
3735 #####
3736 #####
3737 #####
3738 #####
3739 #####
3740 #####
3741 #####
3742 #####
3743 #####
3744 #####
3745 #####
3746 #####
3747 #####
3748 #####
3749 #####
3750 #####
3751 #####
3752 #####
3753 #####
3754 #####
3755 #####
3756 #####
3757 #####
3758 #####
3759 #####
3760 #####
3761 #####
3762 #####
3763 #####
3764 #####
3765 #####
3766 #####
3767 #####
3768 #####
3769 #####
3770 #####
3771 #####
3772 #####
3773 #####
3774 #####
3775 #####
3776 #####
3777 #####
3778 #####
3779 #####
3780 #####
3781 #####
3782 #####
3783 #####
3784 #####
3785 #####
3786 #####
3787 #####
3788 #####
3789 #####
3790 #####
3791 #####
3792 #####
3793 #####
3794 #####
3795 #####
3796 #####
3797 #####
3798 #####
3799 #####
3800 #####
3801 #####
3802 #####
3803 #####
3804 #####
3805 #####
3806 #####
3807 #####
3808 #####
3809 #####
3810 #####
3811 #####
3812 #####
3813 #####
3814 #####
3815 #####
3816 #####
3817 #####
3818 #####
3819 #####
3820 #####
3821 #####
3822 #####
3823 #####
3824 #####
3825 #####
3826 #####
3827 #####
3828 #####
3829 #####
3830 #####
3831 #####
3832 #####
3833 #####
3834 #####
3835 #####
3836 #####
3837 #####
3838 #####
3839 #####
3840 #####
3841 #####
3842 #####
3843 #####
3844 #####
3845 #####
3846 #####
3847 #####
3848 #####
3849 #####
3850 #####
3851 #####
3852 #####
3853 #####
3854 #####
3855 #####
3856 #####
3857 #####
3858 #####
3859 #####
3860 #####
3861 #####
3862 #####
3863 #####
3864 #####
3865 #####
3866 #####
3867 #####
3868 #####
3869 #####
3870 #####
3871 #####
3872 #####
3873 #####
3874 #####
3875 #####
3876 #####
3877 #####
3878 #####
3879 #####
3880 #####
3881 #####
3882 #####
3883 #####
3884 #####
3885 #####
3886 #####
3887 #####
3888 #####
3889 #####
3890 #####
3891 #####
3892 #####
3893 #####
3894 #####
3895 #####
3896 #####
3897 #####
3898 #####
3899 #####
3900 #####
3901 #####
3902 #####
3903 #####
3904 #####
3905 #####
3906 #####
3907 #####
3908 #####
3909 #####
3910 #####
3911 #####
3912 #####
3913 #####
3914 #####
3915 #####
3916 #####
3917 #####
3918 #####
3919 #####
3920 #####
3921 #####
3922 #####
3923 #####
3924 #####
3925 #####
3926 #####
3927 #####
3928 #####
3929 #####
3930 #####
3931 #####
3932 #####
3933 #####
3934 #####
3935 #####
3936 #####
3937 #####
3938 #####
3939 #####
3940 #####
3941 #####
3942 #####
3943 #####
3944 #####
3945 #####
3946 #####
3947 #####
3948 #####
3949 #####
3950 #####
3951 #####
3952 #####
3953 #####
3954 #####
3955 #####
3956 #####
3957 #####
3958 #####
3959 #####
3960 #####
3961 #####
3962 #####
3963 #####
3964 #####
3965 #####
3966 #####
3967 #####
3968 #####
3969 #####
3970 #####
3971 #####
3972 #####
3973 #####
3974 #####
3975 #####
3976 #####
3977 #####
3978 #####
3979 #####
3980 #####
3981 #####
3982 #####
3983 #####
3984 #####
3985 #####
3986 #####
3987 #####
3988 #####
3989 #####
3990 #####
3991 #####
3992 #####
3993 #####
3994 #####
3995 #####
3996 #####
3997 #####
3998 #####
3999 #####
4000 #####
4001 #####
4002 #####
4003 #####
4004 #####
4005 #####
4006 #####
4007 #####
4008 #####
4009 #####
4010 #####
4011 #####
4012 #####
4013 #####
4014 #####
4015 #####
4016 #####
4017 #####
4018 #####
4019 #####
4020 #####
4021 #####
4022 #####
4023 #####
4024 #####
4025 #####
4026 #####
4027 #####
4028 #####
4029 #####
4030 #####
4031 #####
4032 #####
4033 #####
4034 #####
4035 #####
4036 #####
4037 #####
4038 #####
4039 #####
4040 #####
4041 #####
4042 #####
4043 #####
4044 #####
4045 #####
4046 #####
4047 #####
4048 #####
4049 #####
4050 #####
4051 #####
4052 #####
4053 #####
4054 #####
4055 #####
4056 #####
4057 #####
4058 #####
4059 #####
4060 #####
4061 #####
4062 #####
4063 #####
4064 #####
4065 #####
4066 #####
4067 #####
4068 #####
4069 #####
4070 #####
4071 #####
4072 #####
4073 #####
4074 #####
4075 #####
4076 #####
4077 #####
4078 #####
4079 #####
4080 #####
4081 #####
4082 #####
4083 #####
4084 #####
4085 #####
4086 #####
4087 #####
4088 #####
4089 #####
4090 #####
4091 #####
4092 #####
4093 #####
4094 #####
4095 #####
4096 #####
4097 #####
4098 #####
4099 #####
4100 #####
4101 #####
4102 #####
4103 #####
4104 #####
4105 #####
4106 #####
4107 #####
4108 #####
4109 #####
4110 #####
4111 #####
4112 #####
4113 #####
4114 #####
4115 #####
4116 #####
4117 #####
4118 #####
4119 #####
4120 #####
4121 #####
4122 #####
4123 #####
4124 #####
4125 #####
4126 #####
4127 #####
4128 #####
4129 #####
4130 #####
4131 #####
4132 #####
4133 #####
4134 #####
4135 #####
4136 #####
4137 #####
4138 #####
4139 #####
4140 #####
4141 #####
4142 #####
4143 #####
4144 #####
4145 #####
4146 #####
4147 #####
4148 #####
4149 #####
4150 #####
4151 #####
4152 #####
4153 #####
4154 #####
4155 #####
4156 #####
4157 #####
4158 #####
4159 #####
4160 #####
4161 #####
4162 #####
4163 #####
4164 #####
4165 #####
4166 #####
4167 #####
4168 #####
4169 #####
4170 #####
4171 #####
4172 #####
4173 #####
4174 #####
4175 #####
4176 #####
4177 #####
4178 #####
4179 #####
4180 #####
4181 #####
4182 #####
4183 #####
4184 #####
4185 #####
4186 #####
4187 #####
4188 #####
4189 #####
4190 #####
4191 #####
4192 #####
4193 #####
4194 #####
4195 #####
4196 #####
4197 #####
4198 #####
4199 #####
4200 #####
4201 #####
4202 #####
4203 #####
4204 #####
4205 #####
4206 #####
4207 #####
4208 #####
4209 #####
4210 #####
4211 #####
4212 #####
4213 #####
4214 #####
4215 #####
4216 #####
4217 #####
4218 #####
4219 #####
4220 #####
4221 #####
4222 #####
4223 #####
4224 #####
4225 #####
4226 #####
4227 #####
4228 #####
4229 #####
4230 #####
4231 #####
4232 #####
4233 #####
4234 #####
4235 #####
4236 #####
4237 #####
4238 #####
4239 #####
4240 #####
4241 #####
4242 #####
4243 #####
4244 #####
4245 #####
4246 #####
4247 #####
4248 #####
4249 #####
4250 #####
4251 #####
4252 #####
4253 #####
4254 #####
4255 #####
4256 #####
4257 #####
4258 #####
4259 #####
4260 #####
4261 #####
4262 #####
4263 #####
4264 #####
4265 #####
4266 #####
4267 #####
4268 #####
4269 #####
4270 #####
4271 #####
4272 #####
4273 #####
4274 #####
4275 #####
4276 #####
4277 #####
4278 #####
4279 #####
4280 #####
4281 #####
4282 #####
4283 #####
42
```



```
2758 Max.Sample<-Max.SVD$DP$respos.1 # Extract only the portion with sample presence
2759 Max.Sample.Name<-paste(MaxRunDir,
2760 " ",
2761 RVName,
2762 ".csv",
2763 sep="")
2764
2765 Max.Sample[,1]<- RVName
2766 names(Max.Sample)[1] <- "Sample"
2767 write.csv(Max.Sample,file=Max.Sample.Name,row.names=FALSE)
2768
2769 MaxHTMLFile <- paste(MaxOut,
2770 RVName,
2771 ".html",
2772 sep="")
2773
2774 MaxPredFile <- paste(MaxOut,
2775 RVName,
2776 ".csv",
2777 sep="")
2778
2779 MaxLambdaFile <- paste(MaxOut,
2780 RVName,
2781 ".lambdas",
2782 sep="")
2783
2784 MaxProjPredFile <- paste(MaxOut, # Projected prediction
2785 RVName,
2786 "_", # Projected prediction file name
2787 MaxRunName, # Projected prediction file name
2788 " Proj", # This is the new name
2789 ".csv",
2790 sep="")
2791
2792 MaxProjFile <- paste(Max.SVD.Proj.Name,
2793 sep="")
2794
2795 #####
2796 ##### Preparing batch file for maxent.jar #####
2797 ##### with the JarFlags Settings chosen in Module 1 #####
2798 #####
2799 #####
2800
2801 MaxRunBat<-paste("java -Xmx512m -jar ",
2802 jarFileFlag,
2803 JarPar$M3C_JarFlags,
2804 " samplesfile=",
2805 Max.Sample.Name,
2806 " outputdirectory=",
2807 MaxOut,
2808 " environmentalayers=",
2809 Max.Bkgnd.SVD.Name, # This is the "Uninformed Background" file that is called
2810 " projectionlayers=", Max.SVD.Proj.Name,
2811 " logfile=",
2812 RVName,
2813 " log",
2814 " visible=FALSE autorun",
2815 sep="")
2816
2817 write(MaxRunBat,paste(MaxRunDir,
2818 ".bat",
2819 sep=""))
2820
2821 system(paste(MaxRunBat),wait=TRUE) # Running maxent.jar from R as specified in MaxRunBat
2822
2823 #####
2824 ##### Building the MaxEnt output results dataframe #####
2825 #####
2826 #####
2827
```

```
2828 MP3R$GP_Name[pv] <- GP_Name #_InsCode names(GP[[gp]])[pv]
2829 MP3R$GP_Num[pv] <- gp
2830 MP3R$PV_NameGP[pv] <- paste("GrPr",if(gp<10){0},gp,"pv",if(pv<10){0},pv,sep="")
2831 MP3R$Name_BestPV[pv] <- BestPVNamesGP$string
2832 MP3R$Name_AddPV[pv] <- PVTtest
2833
2834 MP3R$NPred_Used[pv] <- NPred_Used
2835 MP3R$NPred_Test[pv] <- NPred_Test
2836 MP3R$RV_NameGP[pv] <- RVName
2837 MP3R$MaxOut[pv] <- MaxOut
2838 MP3R$MaxHTMLFile[pv] <- MaxHTMLFile
2839 MP3R$MaxPredFile[pv] <- MaxPredFile
2840 MP3R$MaxLambdaFile[pv] <- MaxLambdaFile
2841 MP3R$MaxResultsFile[pv] <- MaxResFile
2842 MP3R$AlphaValue[pv] <- P_ValueCutoff
2843 MP3R$Regulariz[pv] <- ReguValues[rz]
2844 MP3R$MaxProjFile[pv] <- MaxProjFile
2845 MP3R$MaxProjPredFile[pv]<- MaxProjPredFile
2846 MP3R$MaxBatFile[pv] <- MaxRunBat
2847 } # End of pv loop
2848
2849 cat("\nDone with maxent.jar Run\n")
2850 M3C_MP3RGP[[gp]] <- MP3R
2851 #_InsCode str(M3C_MP3RGP[[gp]])
2852
2853 #####
2854 #####
2855 ##### MaxEnt
2856 ##### Results compiled for analysis
2857 #####
2858 #####
2859 #####
2860
2861 MP3R <- M3C_MP3RGP[[gp]] # This swap makes possible separation of the running of models and analysis
2862 #####
2863 #####
2864 ##### Loop 1
2865 #####
2866 ##### Update the values of the results
2867 #####
2868 #####
2869 #####
2870
2871 for (pv in seq(nrow(M3C_MP3RGP[[gp]]))) # Needed when reading MaxOut results
2872 {
2873   MaxRes<-read.csv(MP3R$MaxResultsFile[pv]) # Keep the results in memory and extract key results
2874   MaxResT <- t(MaxRes) # Transposed for two reasons, readability and better coercing of values
2875   MaxResE<-as.numeric(MaxResT$rownames(MaxResT))== "Entropy") # MaxEnt model's entropy value
2876   MaxResN<-as.numeric(MaxResT$rownames(MaxResT))== "X.Training.samples") # Get the number of samples
2877   MaxResAUC <- as.numeric(MaxResT$rownames(MaxResT))== "X.Background.points") # Get the background pts
2878   FTV<-((log(MaxResN)-MaxResE)/((log(MaxResN)-log(MaxResN))) ## FTV<=((log(N))-E)/((log(N))-log(n)))
2879
2880 if (gp==1) {
2881   PrevBest <- M3B_MP3R$SelGP_Fin[M3B_MP3R$SelGP_Fin$AI_Num==a&M3B_MP3R$SelGP_Fin$RV_Num==rv.]
2882   FTV< PrevBest$FTVE_PV # The best from the previous NI M3B round
2883   FTV< Add <- FTV<-FTVE_Prev
2884   dfu <- MaxResN+MaxResN-2-MP3R$NPred_Used[pv] # Degrees of freedom for the unexplained variation (found in
2885   the denominator) which appear in the numerator (df)
2886   dfe<- MP3R$NPred_Test[pv]-PrevBest$NPred_Used
2887   FRatioTest <- (FTVE_Add*dfu)/((-FTVE)*dfe)
2888 } # if gp=1
2889
2890 if (gp>=2) {FTVE_Prev <- M3C_MP3R$SelGP[[gp-1]]$FTVE_PV[1]
2891   FTV< Add <- (FTVE<-FTVE_Prev)
2892   dfu <-MaxResN+MaxResN-2-MP3R$NPred_Used[pv] # Degrees of freedom for the unexplained variation
2893   (found in the denominator) which appear in the numerator (df)
2894   dfe <- MP3R$NPred_Test[pv] # Degrees of freedom for the explained variation (found in the numerator),
2895   which appear in the denominator (df2)
2896   FRatioTest <- (FTVE_Add*dfu)/((-FTVE)*dfe)
2897 } # if gp>1
2898
2899
2900
2901
2902
2903
2904
```

```
2895 P_Value<-as.numeric(1-pf(FRatioTest.dfe,dfu))
2896 MP3R$MaxRes_E[pv] <- MaxResE
2897 MP3R$MaxRes_n[pv] <- MaxResn
2898 MP3R$MaxRes_N[pv] <- MaxResN
2899 MP3R$MaxRes_AUC[pv] <- MaxResAUC
2900 MP3R$FTVE_PV[pv] <- FTVE
2901 MP3R$FTVE_AddPV[pv] <- FTVE_Add
2902 MP3R$FTVE_PrevPV[pv] <- FTVE_Prev
2903 MP3R$dfuPV[pv] <- dfu
2904 MP3R$dfepV[pv] <- dfe
2905 MP3R$FRatioTestPV[pv] <- FRatioTest
2906 MP3R$p_ValuePV[pv] <- P_Value
2907 MP3R$sign_PVNamePV[pv] <- NA
2908 # AltCode if(is.nan(MP3R$p_ValuePV[pv]))==T){Clean up your data, there are NaN"}
2909 if (gp== &is.nan(MP3R$p_ValuePV[pv]))==F){
2910 #Needed for MaxResAllIGP which gets printed before going through the "M3C_MP3R$el" loop and reranking
2911 if (MP3R$p_ValuePV[pv] <= P_ValueCutoff)
2912 { # Model selection based on PValue of FTtest
2913 MP3R$sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
2914 } # End of model selection if statement
2915 } # End of if gp 1 statement
2916
2917 if (gp>=2){
2918 if (MP3R$FTVE_PV[pv] >= M3C_MP3R$elGP[[gp-1]]$FTVE_PV[1])
2919 {
2920 MP3R$sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
2921 } # End of model selection if statement
2922 } # End of if gp 2 statement
2923 } # End of pv loop
2924 M3C_MP3RGP[[gp]] <- MP3R # with new information -- after calculating Model Selection Criteria
2925 ## NOTE: ABOVE SECTION M3C SPECIFIC, differs between M38 and M3C
2926
2927 #####
2928 ##### Printing all the iterations' MP3R #####
2929 #####
2930 #####
2931 write.csv(M3C_MP3RGP[[gp]],
2932 paste(dir.M3C_GP,
2933 if(DAC>0){DAC,
2934 "M3C_",
2935 "MP3R_",
2936 RVName,
2937 ".csv",
2938 sep="" ),row.names=FALSE)
2939 #####
2940 ##### Select results that meet selection criteria #####
2941 #####
2942 ##### M3C
2943 #####
2944 #####
2945
2946 SignRows <- which((M3C_MP3RGP[[gp]]$P_ValuePV)<=P_ValueCutoff) # Picking only significant rows
2947 if (gp== &length(SignRows)==0){ cat("\n No significant variables found at all, finished with maxent.jar\n")
2948 gpbest <- NA
2949 M3C_MP3RGP[[gp]]$GP_Best <- gpbest # Best number of Grouped variables
2950 BestPVNamesGP <- NULL
2951 AddPVNamesGP <- NULL
2952 M3C_FinPredData_Dftrm <- NA
2953 M3C_FinPredData_List <- NA
2954 M3C_MP3R$elGP[[gp]] <- NA
2955 GLoopExit<-TRUE # No significant improvement or model
2956 }
2957
2958 if (gp>= &length(SignRows)>1){ # Rearrange/rank them first
2959 cat("\n additional significant variables found, ranking and testing again")
2960 # OptCode gpbest <- gp
2961 MP3R$el <- M3C_MP3RGP[[gp]][SignRows,] # Keeping just those that make the PValueCutoff
2962 MP3R$elOrderPValueFRatio <- MP3R$el[order(MP3R$el$p_ValuePV,~MP3R$el$FRatioTestPV),] # First by PValue,
then by F-ratio test
2963 MP3R$elOrderPValueFRatio$Sign_RankPV <- 1:nrow(MP3R$elOrderPValueFRatio) # Assigning order value
```

```
2964 MP3R$elOrderPValueFRatio$p_P_Value_BestPV <-MP3R$elOrderPValueFRatio$p_P_ValuePV[1] # The one that comes
number 1 is the best
2965 M3C_MP3R$elGP [[gp]] <- MP3R$elOrderPValueFRatio
2966 names(M3C_MP3R$elGP)[gp] <- names(M3C_MP3RGP[[gp]) # InsCode names(M3C_MP3R$elGP)
2967 M3C_MP3R$elGP[[gp]]$Name_BestPV <- M3C_MP3R$elGP[[gp]]$GP_Name[1]
2968 if (gp==1){BestPVNamesGP <- unlist(as.vector(na.omit(istrsplit(M3C_MP3R$elGP[[gp]]$Name_AddPV[1,"_"])))
M3C_MP3R$elGP [[gp]]$Name_BestPV <- BestPVNamesGP
2969 } # End of if gp==1 for assigning bestPVname
2970 if (gp>1){BestPVNamesGP <- unlist(as.vector(na.omit(istrsplit(M3C_MP3R$elGP[[gp]]$Name_BestPV[1,"_"])))
2971 } # End of gp >1 for assigning bestPVname -
2972 AddPVNamesGP <- as.vector(na.omit(M3C_MP3R$elGP[[gp]]$Name_AddPV[2:nrow(M3C_MP3R$elGP[[gp]]))]) #
and additional names
2973 #####
2974 ##### Print the MP3R$el_GP as you find them in the SignRow>1 Loop ####
2975 #####
2976 #####
2977 #####
2978 write.csv(M3C_MP3R$elGP[[gp]],
2979 paste(dir.M3C_GP, # Write out when left with 1
2980 if(DAC>0){DAC,
2981 "M3C_",
2982 "MP3R$el_",
2983 RVName,
2984 ".csv",
2985 sep="" ),row.names=FALSE)
2986
2987 #####
2988 ##### Create new datasets for the next iteration ####
2989 #####
2990 ##### gp +1 is picked from the "best"
2991 #####
2992 #####
2993 #####
2994 DataGP[[gp+1]] <- as.list(DataGP[[gp]]c(BestPVNamesGP.AddPVNamesGP))
2995 names(DataGP)[gp+1] <- paste("M3C_", "GP", "0", gp+1, sep="") # This is the first round
2996 DataGPProj[[gp+1]] <- as.list(DataGPProj[[gp]]c(BestPVNamesGP.AddPVNamesGP))
2997 names(DataGPProj)[gp+1] <- paste("M3C_", "GProj", "0", gp+1, sep="")
2998 } # End of if (gp>= &length(SignRows)>1) statement (this is the "Select and rank and prepare new variable datasets to
test loop")
2999
3000 if (gp>= &length(SignRows)==0){ cat("\n No additional significant variables found, finished with maxent.jar\n")
BestPVNamesGP <- unlist(as.vector(na.omit(istrsplit(M3C_MP3R$elGP[[gp-1]]$Name_BestPV[1,"_"]
3001 )))
AddPVNamesGP <- NULL # No additional significant variables were picked - none improved the
model
3002 M3C_FinPredData_Dftrm <- as.data.frame(DataGP[[gp]]c(BestPVNamesGP.AddPVNamesGP)) #
No additional improvement
3003 M3C_FinPredData_List <- as.list(DataGP[[gp]]c(BestPVNamesGP.AddPVNamesGP)) # Only the
first one of the previous is the best, as the others didn't contribute
3004 ### None are significant in the gp iteration - so the first of the previous is the "best"
3005 gpbest <- gp-1
3006 M3C_MP3R$elGP[[gpbest]]$GP_Best[1] <- gpbest
3007 RVName <- M3C_MP3R$elGP[[gpbest]]$RV_Name[1]
3008 GLoopExit<-TRUE # No significant improvement or model - exit the gp loop
3009 }
3010
3011 if (gp>= &length(SignRows)==1){ cat("\n Last significant variable found, finished with maxent.jar\n")
3012 gpbest <- gp
3013 M3C_MP3R$elGP[[gp]]$GP_Best <- gpbest # Number of best grouped variables
3014
3015 M3C_MP3R$elGP[[gp]] <- M3C_MP3RGP[[gp]][SignRows,]
3016 #####
3017 #####
3018 ##### Printing this last MP3R which is updated ####
3019 #####
3020 #####
3021 write.csv(M3C_MP3RGP[[gp]],
3022 paste(dir.M3C_GP, # Write it out when left with 1
```

```
3023 if(DAC>0){DAC}.
3024 "M3C_",
3025 "MP3RSel_",
3026 RVName,
3027 ".csv",
3028 sep=""),row.names=FALSE)
3029 write.csv(M3C_MBP3RselGP[[gp]][1,], # It is just 1 row
3030 paste(dir,M3C_GP,
3031 if(DAC>0){DAC},
3032 "M3C_",
3033 "MP3RSel_",
3034 RVName,
3035 ".csv",
3036 sep=""),row.names=FALSE)
3037
3038
3039
3040
3041 AddPVNamesGP <- as.vector(na.omit(M3C_MBP3RselGP[[gp]]$Name_AddPV[2:nrow(
3042 M3C_MBP3RselGP[[gp]]))) # and additional names
3043 if (gp==1&length(SignRows)==1){
3044 AddPVNamesGP <- NULL
3045 } ## End of if (gp==1&length(SignRows)==1)
3046
3047 M3C_FinPredData_Dfrfm <- as.data.frame(DataGP[[gp]][c(BestPVNamesGP,AddPVNamesGP)])
3048 M3C_FinPredData_List <- as.list(DataGP[[gp]][c(BestPVNamesGP,AddPVNamesGP)]) # as list keeps
3049 the "grouped" names
3050 } # End of if (gp==1&length(SignRows)==1) statement -- Goes out of GP Loop!
3051
3052 #####
3053 ##### Above is the gp loop: grouped variables iteration loop
3054 #####
3055 } # end of main gp loop - iterations: back up to gp loop if while loop is satisfied
3056 #####
3057 ##### After best gp number has been found
3058 ##### Print last SignGP,
3059 ##### but only line 1 as this takes care of SignifRow==1 or 2
3060 #####
3061 #####
3062
3063 ## Only if gpbest exists
3064
3065 if(!is.na(gpbest))==FALSE)
3066 {
3067 # OptCode cat("\n Printing ModParTable.csv only if gpbest exists")
3068 write.csv(M3C_MBP3RselGP[[gpbest]][1,],
3069 paste(dir,M3C_GP,
3070 if(DAC>0){DAC},
3071 "M3C_",
3072 "MP3RSel_",
3073 "Fin_",
3074 RVName,
3075 ".csv",
3076 sep=""),row.names=FALSE)
3077
3078 write.csv(M3C_MBP3RselGP[[gpbest]][1,], # A copy of the same as above in the "collated results"
3079 paste(dir,M3C_res,
3080 if(DAC>0){DAC},
3081 "M3C_",
3082 "MP3RSel_",
3083 "Fin_",
3084 RVName,
3085 ".csv",
3086 sep=""),row.names=FALSE)
3087
3088 write.csv((M3C_MBP3RselGP[[gpbest]][1,],
3089 paste(dir,M3C_res,
```

```
3090 if(DAC>0){DAC},
3091 "M3C_",
3092 "MP3RSel_",
3093 "Fin_",
3094 RVName,
3095 "Trans",
3096 ".csv",
3097 sep=""),row.names=TRUE)
3098 } #End of gpbest present if statement and csv printing
3099
3100 #####
3101 ### M3C Saving the names of the best set of variables ###
3102 ### in 2 ways:
3103 ### 1. Simply the 2-level names list, where first level is the EV, second is the DV
3104 ### 2. By creating a 2-column dataframe easier to read and count up
3105 ###
3106 #####
3107 M3C_FinPredNames_List <- list()
3108 M3C_FinPredNames_Dfrfm <- data.frame()
3109
3110 ev<-lp ## lp for list position, or ev for environment variable. Keep both for backward compatibility with other modules
3111 if(is.na(M3C_FinPredData_Dfrfm[1][1]))==FALSE)
3112 {
3113 for (ev in seq(length(M3C_FinPredData_List)))
3114 {
3115 M3C_FinPredNames_List[[ev]] <- names(M3C_FinPredData_List[[ev]]) # The DV names (second level)
3116 names(M3C_FinPredNames_List)[ev] <- names(M3C_FinPredData_List[ev]) # The EV names (first level)
3117
3118 M3C_FinPred_EVName <- rep(as.vector(names(M3C_FinPredData_List[ev])),length(M3C_FinPredData_List[[ev]])) ##
3119 Preparing this for a 2-column dataframe which will repeat the EV as long as needed
3120 M3C_FinPred_DVName <- as.vector(rnames(M3C_FinPredData_List[[ev]]))
3121 Dfrfm <- as.data.frame(cbind(M3C_FinPred_EVName,M3C_FinPred_DVName))
3122 M3C_FinPredNames_Dfrfm <- rbind(M3C_FinPredNames_Dfrfm,Dfrfm)
3123 } # End of ev loop
3124 rm(Dfrfm,M3C_FinPred_DVName,M3C_FinPred_EVName)
3125 } # end of if is.na(M3C_FinPredData_Dfrfm) condition
3126
3127
3128 #####
3129 #####
3130 ##### Adding projection lists ###
3131 #####
3132 #####
3133 #####
3134
3135 gpbest <- length(M3C_FinPredData_List) # Final number of grouped variables - or gpbest
3136 # InsCode names(M3C_FinPredData_List)
3137 M3C_Proj_FinPredData_List <- as.list(DataGPProj[[gpbest]][names(M3C_FinPredData_List)])
3138 # InsCode str(M3C_Proj_FinPredData_List)
3139 # InsCode names(M3C_FinPredData_List)
3140 # InsCode names(M3C_Proj_FinPredData_List)
3141
3142 #####
3143 #####
3144 ##### Assigning results to rv lists ###
3145 #####
3146
3147 DataGP_RV[[rv]] <- DataGP
3148 GP_RV[[rv]] <- GP
3149 M3C_MBP3RGP_RV[[rv]] <- M3C_MBP3RGP
3150 M3C_MBP3RselGP_RV[[rv]] <- M3C_MBP3RselGP
```

```

3151 # InvCode str(M3C_FinPredData_List)
3152 M3C_FinPredData_List_RV[[rv]] <- M3C_FinPredData_List
3153 M3C_FinPredNames_List RV[[rv]] <- M3C_FinPredNames_List
3154 M3C_FinPredNames_Difrm RV[[rv]] <- M3C_FinPredNames_Difrm
3155 M3C_FinPredData_Difrm RV[[rv]] <- M3C_FinPredData_Difrm
3156 M3C_Proj_FinPredData_List RV[[rv]] <- M3C_Proj_FinPredData_List
3157
3158 #####
3159 #####
3160 if(is.na(M3C_FinPredData_Difrm[[]][[]]))==FALSE_
3161 {
3162   # cat("print csv only if gpbest exists")
3163
3164   write.csv(M3C_FinPredNames_Difrm_RV[[rv]],
3165             paste(dir,M3C.res,
3166                   "M3C_FinPredNames".",
3167                   "n",
3168                   RVName,
3169                   ".csv",
3170                   sep=",",row.names=FALSE)
3171
3172   write.csv(M3C_FinPredNames_Difrm_RV[[rv]],
3173             paste(dir,M3C_CP,
3174                   "M3C_FinPredNames".",
3175                   "n",
3176                   RVName,
3177                   ".csv",
3178                   sep=",",row.names=FALSE)
3179
3180   write.csv(M3C_FinPredData_Difrm RV[[rv]],
3181             paste(dir,M3C.res,
3182                   "M3C_FinPredData_Difrm".",
3183                   "n",
3184                   RVName,
3185                   ".csv",
3186                   sep=",",row.names=FALSE)
3187   } # End of if is.na(M3C_FinPredData_Difrm) condition
3188
3189 #####
3190 #####
3191 #####
3192 ##### Tidying -- removing extra bits #####
3193 #####
3194 #####
3195 rm(AddPvNamesGP,BestPvNamesGP,BestPvNamesGPsring,DataGP, DataGP_RV,DataGPProj,dfe,dfr,FRatioTest,FTVE,
3196     FTV Add,FTVE_Prev,
3197     gp.GP,GPProj,CPList,CPListProj, GPLoopExit,CP_Name,CPNames,Ip,
3198     Max.Sample,Max.Sample.Name,Max.SWD,Max.Bkgnd.SWD,Max.Bkgnd.SWD.Name, Max.SWD.Proj,
3199     Max.SWD.Proj.Name,
3200     MaxOut_MaxRes,MP3R_MaxResAUC,MaxResE,MaxResFile,MaxResN_MaxResN_MaxResNames,MaxResT_MaxRunBat,
3201     MaxRunDir,MaxHTMLFile,
3202     MaxLambdaFile,MaxProjFile,MaxProjFile_MaxProjPredFile,
3203     Prespos,RVXY,RVXY_Proj_SignRows,SWDList,SWDList_Proj,
3204     MP3Rsel,MP3RselOrderPValueFRatio,
3205     M3C_MP3RC,P,M3C_MP3RselGP,M3C_FinPredData_Difrm,M3C_FinPredData_List, M3C_Proj_FinPredData_List,
3206     M3C_FinPredNames_Difrm,M3C_FinPredNames_List,
3207     PVNames,pVTtest,PVUsed,P_Value)
3208 rm(PrevBest) # For M3C only
3209 } # End of if "no variables at all"
3210 } # End of rv loop
3211 #####
3212 #####

```

```

3210 #####
3211 ##### Passing over RV to al data
3212 #####
3213 #####
3214 #####
3215 M3C_MP3RGP_A[[a]] <- M3C_MP3RGP_RV
3216 M3C_MP3RselGP_A[[a]] <- M3C_MP3RselGP_RV
3217 M3C_FinPredData_List_A[[a]] <- M3C_FinPredData_List_RV
3218 M3C_FinPredNames_List_A[[a]] <- M3C_FinPredNames_List_RV
3219 M3C_FinPredNames_Dfrim_A[[a]] <- M3C_FinPredNames_Dfrim_RV
3220 M3C_FinPredData_Dfrim_A[[a]] <- M3C_FinPredData_Dfrim_RV
3221 M3C_Proj_FinPredData_List_A[[a]] <- M3C_Proj_FinPredData_List_RV
3222 #####
3223 #####
3224 ##### Removing some objects and tidying #####
3225 #####
3226 #####
3227 rm(M3C_MP3RGP_RV,M3C_MP3RselGP_RV,
3228 M3C_FinPredData_List_RV,M3C_Proj_FinPredData_List_RV,
3229 M3C_FinPredNames_List_RV,M3C_FinPredData_Dfrim_RV,M3C_FinPredNames_Dfrim_RV,
3230 GP_RV.P_ValueCutOff)
3231
3232 if (SaveTempFlag==TRUE) {save.image(paste(dir.data,
3233 "Temp",
3234 ScriptNameCode,
3235 " _end_Al",".RData",sep=""))
3236 } # End of SaveTempFlag
3237 } # End of al loop
3238
3239
3240
3241 save.image(paste(dir.data,
3242 if(DAC>0){DAC,
3243 ScriptNameCode,
3244 " _aWl_EndAIRV_Temp",
3245 ".RData",sep=""))
3246
3247 #####
3248 ##### Extracting some results #####
3249 #####
3250 #####
3251 #####
3252 al<-1
3253 st(M3C_MP3RselGP_A[[a]])
3254
3255 length(M3C_MP3RselGP_A)
3256
3257 rv<-1
3258 length(M3C_MP3RselGP_A[[a]])
3259 length(M3C_MP3RselGP_A[[a]][!rv])
3260 mi<-1
3261
3262 #####
3263 ##### Picking out the first line out of all gp iterations that are significant
3264 #####
3265 ##### Regardless of RVs and Al, rowbind all together
3266 #####
3267 #####
3268
3269 rv<-1
3270 al<-1
3271 mi<-1 # model iteration
3272 M3C_MP3RselGP_1st<-data.frame()
3273 NumModAl <-length(M3C_MP3RselGP_A)
3274 for(al in seq(NumModAl))
3275 {
3276 NumModRV <- length(M3C_MP3RselGP_A[[a]])
3277 for(rv in seq(NumModRV))

```



```
3278 {
3279   NumIter<- length(M3C_MP3RselGP_Al[AI][rv])
3280   mi<-1 # model iteration
3281   if(is.na(M3C_MP3RselGP_Al[AI][rv])){NumIter}[rv]==FALSE}
3282   for(mi in seq(NumIter))
3283   {
3284     ModParTable <- data.frame(Al_Num'=1,rv_Num'=1,FTCode=asXJ,"MI_Num'=i)
3285     ModParTable <- cbind(ModParTable,M3C_MP3RselGP_Al[AI][rv][mi])
3286     ModParTable$AI_Num <- al
3287     ModParTable$RV_Num <- rv
3288     ModParTable$FTCode <- substr(ModParTable$RV_Name,1,3)
3289     ModParTable$MI_Num <- mi
3290     M3C_MP3RselGP_1st <- rbind(M3C_MP3RselGP_1st,ModParTable[,]) # Add the first ones ... with the new columns!
3291   } # End of mi
3292   rm(ModParTable,NumIter)
3293 } # End of if NA is false statement
3294 } # End of rv
3295 } # End of al
3296
3297 al
3298 dim(M3C_MP3RselGP_1st)
3299 head(M3C_MP3RselGP_1st[,3:7])
3300 M3C_MP3RselGP_1st[M3C_MP3RselGP_1st$RV_Num==1,1:5]
3301 M3C_MP3RselGP_1st[M3C_MP3RselGP_1st$RV_Num==2,1:5]
3302
3303 M3C_MP3RselGP_1st[M3C_MP3RselGP_1st$AI_Num==1,1:5]
3304 M3C_MP3RselGP_1st[M3C_MP3RselGP_1st$AI_Num==2,1:5]
3305 ##### From here, pick any and all "best"/final models
3306 M3C_MP3RselGP_FinPos <- which(M3C_MP3RselGP_1st$GP_Best>0)
3307 M3C_MP3RselGP_Fin <- M3C_MP3RselGP_1st[M3C_MP3RselGP_FinPos[,1:length(M3C_MP3RselGP_FinPos)],]
3308 rm(M3C_MP3RselGP_FinPos)
3309 dim(M3C_MP3RselGP_Fin)
3310 head(M3C_MP3RselGP_Fin)
3311
3312 #####
3313 #####
3314 ##### The following batch files (.bat) are extra but provide a useful way to copy and group results
3315 ##### and make them more accessible outside R. ####
3316 #####
3317 #####
3318 names(M3C_MP3RselGP_1st)
3319 (M3C_MP3RselGP_Fin$MaxProjPredFile)
3320
3321 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxHTMLFile, ",dir.html,M3C_MP3RselGP_1st$RV_Name,
3322   ".html", sep="")
3323 write(TmpCopyText,paste(dir.M3C,
3324   "M3C_AIBest_",
3325   "HTML",
3326   ".bat",sep="))
3327
3328 M3C1st_HTML.bat <- paste(dir.M3C,
3329   "M3C_AIBest_",
3330   "HTML",
3331   ".bat", sep="")
3332 system(paste(M3C1st_HTML.bat),wait=TRUE,
3333   show.output.on.console=TRUE)
3334
3335 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxOut,"plots",",dir.html,
3336   "plots", sep="")
3337 write(TmpCopyText,paste(dir.M3C,
3338   "M3C_AIBest_",
3339   "HTMLPlots",
3340   ".bat",sep="))
3341 M3C1st_HTMLPlots.bat <- paste(dir.M3C,
3342   "M3C_AIBest_",
3343   "HTMLPlots",
3344   ".bat",sep=")
3345 system(paste(M3C1st_HTMLPlots.bat),wait=TRUE,show.output.on.console=TRUE)
```

```
3346
3347 #### Gives access to the lambda file here so that it can be read via the html link already built
3348 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxLambdaFile, ",dir.html,M3C_MP3RselGP_1st$RV_Name,
3349   "lambdas", sep="")
3350 write(TmpCopyText,paste(dir.M3C,
3351   "M3C_AIBest_",
3352   "LambHtml",
3353   ".bat", sep="))
3354
3355 M3C1st_LambHtml.bat <- paste(dir.M3C,
3356   "M3C_AIBest_",
3357   "LambHtml",
3358   ".bat", sep=")
3359
3360 system(paste(M3C1st_LambHtml.bat),wait=TRUE,
3361   show.output.on.console=TRUE)
3362 #####
3363 #####
3364 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxLambdaFile, ",dir.lamdba,M3C_MP3RselGP_1st$RV_Name,
3365   "lam.csv", sep="")
3366 write(TmpCopyText,paste(dir.M3C,
3367   "M3C_AIBest_",
3368   "Lambda",
3369   ".bat", sep="))
3370 M3C1st_Lambda.bat <- paste(dir.M3C,
3371   "M3C_AIBest_",
3372   "Lambda",
3373   ".bat", sep="")
3374 system(paste(M3C1st_Lambda.bat ),wait=TRUE,show.output.on.console=TRUE)
3375
3376 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxPredFile, ",dir.Pred CSV,M3C_MP3RselGP_1st$RV_Name,
3377   "Pred.csv", sep="")
3378 write(TmpCopyText,paste(dir.M3C,
3379   "M3C_AIBest_",
3380   "MaxPred",
3381   ".bat", sep="))
3382 M3C1st_MaxPred.bat <- paste(dir.M3C,
3383   "M3C_AIBest_",
3384   "MaxPred",
3385   ".bat", sep="")
3386 system(paste(M3C1st_MaxPred.bat),wait=TRUE,show.output.on.console=TRUE)
3387
3388 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxResultsFile, ",dir.res,M3C_MP3RselGP_1st$RV_Name,
3389   "Res.csv", sep="")
3390 write(TmpCopyText,paste(dir.M3C,
3391   "M3C_AIBest_",
3392   "MaxRes",
3393   ".bat", sep="))
3394 M3C1st_MaxRes.bat <- paste(dir.M3C,
3395   "M3C_AIBest_",
3396   "MaxRes",
3397   ".bat", sep="")
3398 system(M3C1st_MaxRes.bat,wait=TRUE,show.output.on.console=TRUE)
3399
3400 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxProjPredFile, ",dir.Proj_CSV,M3C_MP3RselGP_1st$RV_Name,
3401   "ProjPred.csv", sep="")
3402 write(TmpCopyText,paste(dir.M3C,
3403   "M3C_AIBest_",
3404   "MaxProj",
3405   ".bat", sep="))
3406 M3C1st_MaxProj.bat <- paste(dir.M3C,
3407   "M3C_AIBest_",
3408   "MaxProj",
3409   ".bat", sep=")
3410 system(paste(M3C1st_MaxProj.bat),wait=TRUE,show.output.on.console=TRUE)
3411
3412 ## Added extracting projection files. Note that these are the data files, not the predictions
3413 TmpCopyText <- paste("copy ", M3C_MP3RselGP_1st$MaxProjFile, ",dir.ProjData_SWD,M3C_MP3RselGP_1st$RV_Name,
3414   "Proj.csv", sep="")
```



```
3415 write(TmpCopyText,paste(dir.M3C,
3416 "M3C_AlIBest_",
3417 "MaxProjData",
3418 ".bat",sep=""))
3419 M3C1st_MaxProjData.bat <- paste(dir.M3C,
3420 "M3C_AlIBest_",
3421 "MaxProjData",
3422 ".bat",sep="")
3423 system(paste(M3C1st_MaxProjData.bat),wait=TRUE,show.output.on.console=TRUE)
3424 #####
3425 #####
3426 #####
3427 Obj2Save <- objects(pattern="M3C")
3428 Obj2Save # These are all the M3C objects created in this module
3429 #####
3430 save(list=Obj2Save,
3431 file=paste(dir.rdata,
3432 "Obj_", # These are all the M3A objects created in this module
3433 if(DAC>0){DAC},
3434 ScriptNameCode,"_",
3435 "End_All","RData",sep=""))
3436 #####
3437 rm(TmpCopyText,Obj2Save)
3438 save.image(paste(dir.rdata,
3439 if(DAC>0){DAC},
3440 ScriptNameCode,
3441 " aWI_End_All_Alphas_RV",
3442 "RData",sep=""))
3443 #####
3444 #####
3445 ## Next script ###
3446 #####
3447 #####
```



```
1 #####
2 #####
3 #####
4 ##### MIA7 Module 4 - Model Extraction and Post Processing #####
5 ##### Selecting Models and Creating Alternative Output #####
6 #####
7 ##### M4a_SelMods_AM.r #####
8 #####
9 ##### Author: Sabrina Mazzoni #####
10 #####
11 #####
12 #####
13 #####
14 ## This module requires considerable user decisions and control, like choosing which models (if not all) to analyse
15 ## Once this is done, then the raw predictions will be extracted, converted to the PRO output format and prepared for
    further evaluation
16 ## Other model parameters will also be extracted and collated into additional parameter tracking tables.
17 ## For example, opening all relevant lambda files created by maxent.jar, then counting the type and number of
18 ## variables used in the final models.
19
20 memory.size0
21 memory.limit0
22 memory.limit(8000000)
23
24 ScriptNameCode <- "M4a"
25
26 ##### Next section optional #####
27 ##### Some data clean up at the start #####
28 ##### Choose objects for deletion, especially if datasets are large
29 ## These don't have to be kept for M4, but may be kept if desired
30 DelObjFlag<-FALSE # Switch this flag then run
31
32 M1_Patt<- objects(pattern="M1")
33 M1_Patt
34 save(list=M1_Patt,
35      file=paste(dir.rdata,
36                  "Obj_",
37                  DAC,
38                  "M1_Patt",
39                  ".RData",sep=""))
40 # If too big, can be deleted
41 if (DelObjFlag==TRUE) {rm(list=M1_Patt)
42   rm(M1_Patt)}
43 #####
44 M2_Patt<- objects(pattern="M2")
45 M2_Patt
46 save(list=M2_Patt,
47      file=paste(dir.rdata,
48                  "Obj_",
49                  DAC,
50                  "M2_Patt",
51                  ".RData",sep=""))
52 if (DelObjFlag==TRUE) {
53   rm(list=M2_Patt)
54   rm(M2_Patt)
55 }
56
57 #####
58 M2h_Patt<- objects(pattern="M2h")
59 M2h_Patt
60 save(list=M2h_Patt,
61      file=paste(dir.rdata,
62                  "Obj_",
63                  DAC,
64                  "M2h_Patt",
65                  ".RData",sep=""))
66
67 if (DelObjFlag==TRUE) {
68   rm(list=M2h_Patt)
69   rm(M2h_Patt)
```

```
70 }
71 #####
72 #####
73 save(M2_N_SWD_RV,
74      file=paste(dir.rdata,
75                  "Obj_",
76                  DAC,
77                  "M2_N_SWD_RV",
78                  ".RData",sep=""))
79 # rm(M2_N_SWD_RV)
80 #####
81 Proj_Patt<- objects(pattern="Proj")
82 Proj_Patt
83
84 save(list=Proj_Patt,
85      file=paste(dir.rdata,
86                  "Obj_",
87                  DAC,
88                  "Proj_Patt",
89                  ".RData",sep=""))
90
91 if (DelObjFlag==TRUE) {
92   rm(list=Proj_Patt)
93   rm(Proj_Patt)
94 }
95
96 #####
97 #object.size(M3_N_SWD_RV)
98
99 M3_Patt<- objects(pattern="M3")
100 M3_Patt
101 save(list=M3_Patt,
102      file=paste(dir.rdata,
103                  "Obj_",
104                  DAC,
105                  "M3_Patt",
106                  ".RData",sep=""))
107 #####
108 M3A_Patt<- objects(pattern="M3A")
109 M3A_Patt
110 save(list=M3A_Patt,
111      file=paste(dir.rdata,
112                  "Obj_",
113                  DAC,
114                  "M3A_Patt",
115                  ".RData",sep=""))
116 if (DelObjFlag==TRUE) {
117   rm(list=M3A_Patt) # If too big, can be deleted
118   rm(M3A_Patt)
119 }
120
121 #####
122 #####
123 ##### Setting directories,
124 ##### including some that will be used to make MaxEnt's output easier to access
125 #####
126 #####
127 dir.M4 <- paste(dir.max,"M4_All","\",sep="")
128 # AltCode dir.M4 <- choose.dir() ## use this to choose directory by hand
129 # AltCode dir.M4 <- paste(dir.M4,"\",sep="")
130 dir.create(dir.M4)
131
132 dir.html <- paste(dir.M4,
133                  "HTML", # MaxEnt's HTML file
134                  "\",sep="") #
135 dir.create(dir.html)
136 dir.create(paste(dir.html,"plots",sep="")) #MaxEnt's HTML's plots folder
137
138 dir.lambda <- paste(dir.M4,
139                    "Lambda", # MaxEnt's lambda file
```

```

140  "\\\" , sep="" ) #
141  dir.create(dir.lambda)
142
143  dir.ModPar <- paste(dir.M4,
144    "ModPar", # The new lambda files, ModParameter Tables created here
145    "\\\" , sep="" ) #
146  dir.create(dir.ModPar)
147
148
149  dir.Pred_CSV <- paste(dir.M4,
150    "Pred_CSV", # Raw predictions
151    "\\\" , sep="" ) #
152  dir.create(dir.Pred_CSV)
153
154  dir.Pred_ASC <- paste(dir.M4,
155    "Pred_ASC", # Raw predictions
156    "\\\" , sep="" ) #
157  dir.create(dir.Pred_ASC)
158
159  dir.Pred_ASCPPO <- paste(dir.M4,
160    "Pred_ASCPPO", # PRO predictions
161    "\\\" , sep="" ) #
162  dir.create(dir.Pred_ASCPPO)
163
164
165  dir.res <- paste(dir.M4,
166    "Res_Max", # MaxEnt Results file
167    "\\\" , sep="" ) #
168  dir.create(dir.res)
169  dir.Proj_CSV <- paste(dir.M4, # Adding projection
170    "Proj_CSV",
171    "\\\" , sep="" )
172  dir.create(dir.Proj_CSV)
173
174  dir.ProjData_SVD <- paste(dir.M4, # Adding projection data file
175    "ProjData_SVD",
176    "\\\" , sep="" )
177
178  dir.create(dir.ProjData_SVD)
179
180
181  #####
182  #####
183  ##### Selecting models to post-process -
184  Alternative MaxEnt (AM Models)
185  ##### (maximum likelihood interpretation options)
186  #####
187  ## The next lines of code are just to show which data are available
188  #####
189  # Dataframe MP3Rsel
190  al <-1
191  rv <-1
192  gp <-1
193  str(M3B_MP3RselGP_AI) # List of 3 levels: Alphas, RV, GP (iterations),
194  class(M3B_MP3RselGP_AI[[dir.village]])
195  str(M3C_MP3RselGP_AI) # List of 3 levels: Alphas, RV, GP (iterations),
196
197
198  M3B_MP3RselGP_1st$MaxProjFile
199  M3B_MP3RselGP_1st$MaxProjPredFile
200
201
202  M3B_MP3RselGP_1st[c(1:4,8:11),1:6]
203  names(M3B_MP3RselGP_1st)
204
205  M3C_MP3RselGP_1st[c(1:4,8:11),1:6]
206  names(M3C_MP3RselGP_1st)
207
208  M3C_MP3RselGP_Fin[,1:7]
209  dim(M3C_MP3RselGP_Fin)

```

```

210
211  ##### After inspecting the results, models to be comparing/collating/extracting etc. are selected
212  ## In this example, the simplest "four" models, i.e., those with 1 to 4 EVs from the M3B module (no interactions) are selected
213  ## plus the "best/final" no-interactions models (M3B)
214  ## and the "best/final" most complex models, with interactions (M3C)
215  Pos_EV4 <- which(M3B_MP3RselGP_1st$GP_Num==4) # This command captures the first 4 for any al or rv
216
217  M3B_MP3RselGP_Fin[,1:7]
218
219  M3C_MP3RselGP_Fin[,1:7]
220
221
222  M4a_MP3RselMods_AM <- rbind(M3B_MP3RselGP_1st[c(Pos_EV4),], # Place this first
223    M3B_MP3RselGP_Fin, # Next in terms of complexity
224    M3C_MP3RselGP_Fin) # Most complex model
225
226  as.data.frame(M4a_MP3RselMods_AM[,c(1:4,7,8)])
227
228  #####
229  #####
230  ##### Preparation of the M4 core list to which most results will be saved #####
231  #####
232
233  str(M4a_MP3RselMods_AM) # These are just for the alternative MaxEnt procedure, similar scripts for the standard procedure
234  dim(M4a_MP3RselMods_AM)
235  M4a_AMNames <- paste(DAC,
236    # #"AM",
237    M4a_MP3RselMods_AM$RV_Name,
238    sep="" ) # or names of M4a_SelMods #
239  as.data.frame(M4a_AMNames)
240  as.data.frame(cbind(M4a_MP3RselMods_AM$RV_Name,
241    M4a_MP3RselMods_AM$RV_Num,
242    M4a_MP3RselMods_AM$AI_Num,
243    M4a_MP3RselMods_AM$MI_Num))
244
245  #####
246  #####
247  #####
248
249  ## Extracting Key MaxEnt outputs for selected models,
250  ## using the ModSelectionTracking table
251
252  copyfiles <- paste("copy ", M4a_MP3RselMods_AM$MaxHTMLFile," ", dir.html,M4a_AMNames,
253    #" ",M4a_MP3RselMods_AM$PV_NameGP, # To make sense when RVName fails to hold this info
254    " ",html", sep="" )
255  write(copyfiles,paste(dir.M4,
256    "M4_AllBest_",
257    "HTML",
258    ".bat", sep="" ))
259
260  M4_HTML.bat <- paste(dir.M4,
261    "M4_AllBest_",
262    "HTML",
263    ".bat", sep="" )
264  system(paste(M4_HTML.bat),wait=TRUE)
265  show.output.on.console=TRUE)
266
267  copyfiles <- paste("copy ", M4a_MP3RselMods_AM$MaxOut,"plots",",", dir.html,
268    "plots", sep="" )
269  write(copyfiles,paste(dir.M4,
270    "M4_AllBest_",
271    "HTMLPlots",
272    ".bat", sep="" ))
273  M4_HTMLPlots.bat <- paste(dir.M4,
274    "M4_AllBest_",
275    "HTMLPlots",
276    ".bat", sep="" )
277  system(paste(M4_HTMLPlots.bat),wait=TRUE)show.output.on.console=TRUE)
278  TempCopyText <- paste("copy ", M4a_MP3RselMods_AM$MaxLambdaFile," ", dir.html,M4a_AMNames,
279    "lambdas", sep="" )

```

```
280 write(TmpCopyText,paste(dir.M4,
281 "M4_AllBest_",
282 "LambHtml",
283 " ",sep=""))
284 M4_LambHtml.bat <- paste(dir.M4,
285 "M4_AllBest_",
286 "LambHtml",
287 " ",sep="")
288
289 system(paste(M4_LambHtml.bat),wait=TRUE.
290 show.output.on.console=TRUE)
291 #####
292 copyfiles <- paste("copy ", M4a_MP3RselMods_AM$MaxLambdafile, ",
293 dir.lambda,M4a_AMNames,
294 #", "M4a_MP3RselMods_AM$pv_NameGP,
295 "Lam.csv", sep="")
296 write(copyfiles,paste(dir.M4,
297 "M4_AllBest_",
298 "Lambda",
299 " ",sep=""))
300 M4_Lambda.bat <- paste(dir.M4,
301 "M4_AllBest_",
302 "Lambda",
303 " ",sep="")
304
305 system(paste(M4_Lambda.bat),wait=TRUE.show.output.on.console=TRUE)
306 #####
307 copyfiles <- paste("copy ", M4a_MP3RselMods_AM$MaxPredFile, ",
308 dir.Pred CSV,M4a_AMNames,
309 #", "M4a_MP3RselMods_AM$pv_NameGP,
310 "Pred.csv", sep="")
311 write(copyfiles,paste(dir.M4,
312 "M4_AllBest_",
313 "MaxMod",
314 " ",sep=""))
315 M4_MaxMod.bat <- paste(dir.M4,
316 "M4_AllBest_",
317 "MaxMod",
318 " ",sep="")
319 system(paste(M4_MaxMod.bat),wait=TRUE.show.output.on.console=TRUE) # there's a bug ... if I have
320 show.output.on.console=FALSE it does not run!
321
322 copyfiles <- paste("copy ", M4a_MP3RselMods_AM$MaxResultsFile, ", dir.res,M4a_AMNames,
323 #", "M4a_MP3RselMods_AM$pv_NameGP,
324 "Res.csv", sep="")
325 write(copyfiles,paste(dir.M4,
326 "M4_AllBest_",
327 "MaxRes",
328 " ",sep=""))
329
330 M4_MaxRes.bat <- paste(dir.M4,
331 "M4_AllBest_",
332 "MaxRes",
333 " ",sep="")
334 system(M4_MaxRes.bat,wait=TRUE.show.output.on.console=TRUE)
335 #####
336 #####
337 #####
338 ##### Extracting parameters for the projected predictions
339 ##### MaxEnt models are trained on 10,000+training points
340 ##### but Projected geographically to a larger/full extent
341 #####
342
343 TmpCopyText <- paste("copy ", M4a_MP3RselMods_AM$MaxProjPredFile, ", dir.Proj_CSV,M4a_MP3RselMods_AM$
344 RV_Name,
345 "ProjPred.csv", sep="")
346 write(TmpCopyText,paste(dir.M3B,
```

```
346 "M3B_AllBest_",
347 "MaxProj",
348 " ",sep="")
349 M3B1st_MaxProj.bat <- paste(dir.M3B,
350 "M3B_AllBest_",
351 "MaxProj",
352 " ",sep="")
353 system(paste(M3B1st_MaxProj.bat),wait=TRUE.show.output.on.console=TRUE)
354
355 ### Te following files are the data files, not the predictions
356
357 TmpCopyText <- paste("copy ", M4a_MP3RselMods_AM$MaxProjFile, ", dir.ProjData_SVD,M4a_MP3RselMods_AM$
358 RV_Name,
359 "Proj.csv", sep="")
360 write(TmpCopyText,paste(dir.M3B,
361 "M3B_AllBest_",
362 "MaxProjData",
363 " ",sep=""))
364 M3B1st_MaxProjData.bat <- paste(dir.M3B,
365 "M3B_AllBest_",
366 "MaxProjData",
367 " ",sep="")
368 system(paste(M3B1st_MaxProjData.bat),wait=TRUE.show.output.on.console=TRUE)
369
370 head(M4a_MP3RselMods_AM)
371 nrow(M4a_MP3RselMods_AM)
372
373 write.csv(M4a_MP3RselMods_AM,
374 paste(dir.M4, #
375 if(DAC>0){DAC},
376 ScriptNameCode,
377 "AM$el_MP3R_Dtfm",
378 ".csv",
379 sep="").row.names=FALSE) #
380
381 save.image(paste(dir.rdata,
382 if(DAC>0){DAC},
383 ScriptNameCode,
384 "_SelMod_AM",
385 ".RData",sep=""))
386
387 ##### End of selection of models to inspect
388 #####
389 #####
390 ##### Next script #####
391 #####
392 #####
393 #####
394 #####
395 #####
396 #####
397 ##### MIA7 Module 4 - Model extraction and post processing #####
398 ##### Selecting model and creating alternative output #####
399 #####
400 ##### M4b_ModParCoefPredLam.R
401 ##### Opening maxent.jar lambda files
402 ##### Extracting key details of the final model
403 ##### Names of variables, respective coefficients, and model parameters
404 #####
405 ##### Note: This component/script started just as "simple counter of DV's from lambda file
406 ##### but has been extended to aim at "collecting all key model parameters by looking not just at the lambda files
407 ##### Key elements to this script
408 ##### 1. extracting the names of the EV's and DV . counting each,
409 ##### 2. extracting maxent.jar's coefficients and constants and bringing back other
410 ##### 3. key parameters from the M4a_MP3RselMods_AM tracking table
411 #####
412 #####
413 #####
414 #####
```



```
415 ## Set up directory
416
417 ScriptNameCode <- "M4b"
418 require(pyr.MIAPar$dir.Rlib) # Used for joining data
419
420 ## Option 1: pick by hand
421 # OptCode   dir.lambda <-choose.dir("Choose data directory --") # or pick from your objects
422 # _OptCode   dir.lambda <-paste(dir.lambda, "\\", sep="") # adding \\ to all directory objects consistently
423
424 ## Option 2: pick from previous objects
425 str(dir.lambda)
426 setwd(dir.lambda) # Necessary, or it will use your default
427 getwd()
428
429 DirFilesVect <- as.vector(list.files(dir.lambda)) ##### testing: CSV.file.list[51]
430
431
432
433 str(DirFilesVect)
434 DirFilesVect
435 DirFilesVect[]
436
437 str(M4a_MP3RSelMods_AM)
438
439
440 names(M4a_MP3RSelMods_AM)
441 length(M4a_MP3RSelMods_AM)
442 M4a_AMNames
443
444 M4a_MP3RSelMods_AM$GP_Name
445 M4a_MP3RSelMods_AM$SRV_Name
446
447 names(M4a_MP3RSelMods_AM)
448
449
450 #####
451 #####
452 #####
453 ##### Counting variables
454 ##### for AM lambda files
455 #####
456 #####
457 #####
458
459 LamOrigList_AM <- as.list(1:length(M4a_AMNames))
460
461 M4b_MaxCoeffList_AM <- as.list(1:length(M4a_AMNames)) # Holds all individual variables (and their coefficients) from the
lambda files
462 M4b_MaxCoeffList_AM <- as.list(1:length(M4a_AMNames)) # Holds the "last 4 values", from the lambda files (so called
"constants")
463
464 names(M4b_MaxConstList_AM) <- M4a_AMNames
465 M4b_MaxModPar_AM <- as.list(1:length(M4a_AMNames)) # Keeps all (almost all) MaxEnt model parameters
466
467
468
469 for (mod in seq(length(M4a_AMNames))) #
{ # AltCode LamBestList[mod]]
<-as.data.frame(read.csv(Sys.glob(paste(dir.P4.Lam, DirFilesVect[mod], sep="")), as.is=TRUE, header=FALSE))
LamOrigList_AM[[mod]] <-as.data.frame(read.csv(Sys.glob(
paste(dir.lambda,
M4a_AMNames[mod], DirFilesVect[mod],
"Lam.csv",
sep="")),
as.is=TRUE,
header=FALSE))
470
471
472
473
474
475
476
477
```

```
478
479 # str(LamOrigList_AM[[mod]])
480 # head(LamOrigList_AM[[mod]])
481 LamRowNum <- nrow(LamOrigList_AM[[mod]])$V1== "linearPredictorNormalizer") # Pick the first
PosMaxConst <- which(LamOrigList_AM[[mod]]$V1== "linearPredictorNormalizer") # These are the model
M4b_MaxConstList_AM[[mod]] <- LamOrigList_AM[[mod]]$PosMaxConstLamRowNum,1:2] # These are the model
parameters that need to be kept and separated
names(M4b_MaxCoeffList_AM)[mod] <- paste("ModPar_", M4a_AMNames[mod], sep="") #The same as above but No. 1
indicates column - as it is a vector
CoefAll <- LamOrigList_AM[[mod]][1:(PosMaxConst-1)] # Temporarily, consider all first then just pulls out non-zeros
CoefMaxPos <- which(as.numeric(CoefAll[,2])!=0) # Only takes non-zero values into consideration (### Makes sure
coercion as numeric)
486
487
488 TotMaxDV <- length(CoefMaxPos) # Number of derived variables which are used in the final MaxEnt Model
489
490 M4b_MaxCoeffList_AM [[mod]] <- LamOrigList_AM[[mod]][CoefMaxPos,] # Bring all 4 columns
CountEVDV <- nrow(M4b_MaxCoeffList_AM[[mod]])
491
492 if (CountEVDV!=0) { # Only to be run if there are ANY variables at all ... some exceptionally poor models have none
picked.
ModName <- as.vector(rep(M4a_AMNames[mod], CountEVDV)) #vector(rep(M4a_AMNames[mod,1], CountEVDV))
ModEVNames <- as.vector(1:CountEVDV) # Just setting up
ModDVNames <- as.vector(1:CountEVDV) # Just setting up
NumEV <- as.vector(1:CountEVDV)
NumDV <- as.vector(1:CountEVDV) ## Add this so it can be in the file and ev dv as well
EVDVNames <- strsplit(as.vector(M4b_MaxCoeffList_AM[[mod]][1:1])
",\\",
#dv<-1
for(dv in seq(length(EVDVNames)))
{
if(length(EVDVNames[[dv]]==2) { # If both EV and DV got split properly
ModEVNames[dv] <- EVDVNames[[dv]][1] # The first part is the EVName ## in order not to treat them as factors
ModDVNames[dv] <- EVDVNames[[dv]][2] # The second part is the Derived variables (DV) name
} else {# when the original EVNames did not get carried over, when 'flattening' list originally
TempSplit <- strsplit(as.vector(EVDVNames[[dv]]
",\\",
# Splitting EVNames from the DVNames
ModEVNames[dv] <- TempSplit[[1]] # Only the first portion (EVName)
ModDVNames[dv] <- EVDVNames[[dv]] # The DV names
} # End of extracting EVs and DVs, which are split in 2 parts: the if else statement
}
} # End of extracting model's individual EV and DV Names
ModEVNames<-as.character(ModEVNames)
#str(ModDVNames)
XtraCols <- cbind(ModName, ModEVNames, ModDVNames, NumEV, NumDV) # in order not to treat them as factors
#class(XtraCols[1])
M4b_MaxCoeffList_AM[[mod]] <- cbind(M4b_MaxCoeffList_AM[[mod]], XtraCols)
for (col in 5:7)
{M4b_MaxCoeffList_AM[[mod]][,col] <- as.character(M4b_MaxCoeffList_AM[[mod]][,col])}
# str(M4b_MaxCoeffList_AM[[mod]])
M4b_MaxCoeffList_AM[[mod]]$NumEV <- length(unique(ModEVNames))
M4b_MaxCoeffList_AM[[mod]]$NumDV <- length(unique(ModDVNames))
names(M4b_MaxCoeffList_AM)[mod] <- paste(names(M4b_MaxCoeffList_AM[mod]), " EV",
M4b_MaxCoeffList_AM[[mod]]$NumEV[1],
"DV", M4b_MaxCoeffList_AM[[mod]]$NumDV[1], sep="") # Metadata embedding
names(M4b_MaxCoeffList_AM[[mod]][1] <- "EVDVName" # Adding name to maxent.jar's lambda file 1 column
names(M4b_MaxCoeffList_AM[[mod]][2] <- "DVCoeff" # Adding name to maxent.jar's lambda column
names(M4b_MaxCoeffList_AM[[mod]][3] <- "EVDVMin" # Adding name to maxent.jar's lambda file 1 column
names(M4b_MaxCoeffList_AM[[mod]][4] <- "EVDVMax" # Adding name to maxent.jar's lambda file 1 column
#####
#####
##### Part 2: Extracting more parameters #####
##### It is useful to separate what maxent.jar calls the coefficients from the constants,
##### bring them back into the same table, assigning each to a row,
##### in order to then use them in the MaxEnt formula directly
##### Also may add AUC and other key info from the MP3R table
### into 1 single table with all model parameters, if desired
543
```

```
609 ##### (user can decide not to use)
610 ###
611 #####
612 #####
613 str(M4b_MaxModPar_AM)
614 length(M4b_MaxModPar_AM)
615 dim(M4b_MaxModPar_AM)[mod]]
616 head(M4b_MaxModPar_AM[[mod]])
617 tail(M4b_MaxModPar_AM[[mod]])
618 (M4b_MaxModPar_AM[[mod]][1:7])
619 (M4b_MaxModPar_AM[[1:7]])
620 head(M4b_MaxModPar_AM[[1]])
621
622
623 names(M4b_MaxModPar_AM[[mod]])
624 M4b_MaxModPar_AM[[mod]][1]
625
626 M4b_MaxModPar_AM_Dfrfm <- M4b_MaxModPar_AM[[1]]
627 for (mod in 2:length(M4b_MaxModPar_AM))
628 {
629     M4b_MaxModPar_AM_Dfrfm <- rbind(M4b_MaxModPar_AM_Dfrfm, M4b_MaxModPar_AM[[mod]])
630     # OptCode c(LamCountList_AM_Dfrfm, LamCountList_AM[[mod]]) ##
631 }
632
633 str(M4b_MaxModPar_AM_Dfrfm)
634 head(M4b_MaxModPar_AM_Dfrfm)
635
636 write.csv(M4b_MaxModPar_AM_Dfrfm, # Prints all
637           paste(dir.M4,
638                 DAC,
639                 # OptCode "\\",
640                 # OptCode "Mod",
641                 "M4b_MaxModPar_AM_Dfrfm_All", ".csv", sep=""), row.names=FALSE)
642
643 write.csv(M4b_MaxModPar_AM_Dfrfm[c(1,13,9,5,7)], # Prints just the three columns with ModName, ModEVName and
644           ModDVName, add RV and MI and NI for easy sorting
645           paste(dir.M4,
646                 DAC,
647                 # OptCode "\\",
648                 # _OptCode "Mod",
649                 "M4b_MaxModPar_AM_Mod_EVName_DVName", ".csv", sep=""), row.names=FALSE)
650 #####
651 ###
652 ### Collapsing into one, to count things up and extract specific columns needed
653 ###
654 #####
655 ModName <- unique(M4b_MaxModPar_AM_Dfrfm$ModName)
656 length(ModName)
657 CountModName <- count(M4b_MaxModPar_AM_Dfrfm$ModName)
658 class(CountModName)
659 CountModName[,2]
660
661 mod<-1
662 M4b_MaxModPar_AM_ModName1st <- M4b_MaxModPar_AM_Dfrfm[,1]
663 for(mod in seq(length(ModName)))
664 {
665     print(paste("Mod ", mod, " ", ModName[mod]))
666     ModNamePos <- which(M4b_MaxModPar_AM_Dfrfm$ModName==ModName[mod])
667     # Just pick the "first one" always
668     M4b_MaxModPar_AM_ModName1st[mod] <- M4b_MaxModPar_AM_Dfrfm[ModNamePos[1],]
669 }
670
671
672 dim(M4b_MaxModPar_AM_ModName1st)
673 dim(M4a_MP3RSeIMods_AM)
674 tail(M4a_MP3RSeIMods_AM[-3:7])
675
676 head(M4b_MaxModPar_AM_ModName1st)
677 names(M4b_MaxModPar_AM_ModName1st[c(5,17,9,10,16)])
```

```
544 ConstNames <- (M4b_MaxConstList_AM[[mod]][1]) # Use first column for column names
545 ConstVal<- as.data.frame((M4b_MaxConstList_AM[[mod]][2:D]) # Use second column for values, etc.
546 names(ConstVal) <- ConstNames
547 RV_Num <- M4a_MP3RSeIMods_AM$RV_Num[mod] # Useful for "grouping results later"
548 SelMeth <- substr(M4a_MP3RSeIMods_AM$RV_Name[mod],1,1) # This is very specific to RV_Name starting with a or s
for alternative or standard MaxEnt
549 Lambda <- 1
550 Alpha <- M4a_MP3RSeIMods_AM$AlphaValue[mod] # Useful for "grouping results later"
551 FTCCode<- M4a_MP3RSeIMods_AM$FTCode[mod] # Useful for "grouping results later"
552
553 DVs<-M4b_MaxCoefList_AM[[mod]]$NumDV # Copies for final table
554 EVs <- M4b_MaxCoefList_AM[[mod]]$NumEV # Total EV
555 IEV <- 0 # Gives a zero, unless it is interactions
556 if(M4a_MP3RSeIMods_AM$FTCode[mod]== "a\N") { IEV <- M4a_MP3RSeIMods_AM$FTCode=="a\N"} ## Only interactions EVs
557 IEVPos<- which(M4a_MP3RSeIMods_AM$FTCode=="a\N")
558
559 FTVE_PV <- M4a_MP3RSeIMods_AM$FTVE_PV[mod]
560 MaxRes_AUC <- M4a_MP3RSeIMods_AM$MaxRes_AUC[mod]
561 Test_AUC<-0 #as.numeric()
562 GPNName <- M4a_MP3RSeIMods_AM$GP_Name[mod]
563
564 MaxRes_n <- M4a_MP3RSeIMods_AM$MaxRes_n[mod]
565
566 M4b_MaxModPar_AM[[mod]]<- cbind(M4b_MaxCoefList_AM[[mod]],
567                                DAC,
568                                RV_Num,
569                                SelMeth,
570                                Lambda,
571                                Alpha,
572                                FTCCode,
573                                DVs,
574                                EVs,
575                                IEV,
576                                FTVE_PV,
577                                MaxRes_AUC,
578                                Test_AUC, # Make a space for this now, for the eventual case that they are calculated later
579                                GPNName,
580                                MaxRes_n,
581                                ConstVal[c(3,4,1,2)]
582                                )
583
584 names(M4b_MaxModPar_AM)[mod]<- names(M4b_MaxCoefList_AM)[mod]
585 #####
586 #####
587 ##### Print each separate file, with Aall columns (could choose less, like 5:9 only) #####
588 #####
589 write.csv(M4b_MaxModPar_AM[[mod]], #[5:9] #
590           paste(dir.ModPar,
591                 DAC,
592                 # OptCode "\\", # "key", "Mod",
593                 names(M4b_MaxModPar_AM[[mod]], ".csv", sep=""), row.names=FALSE) # Keep the row names as this is the
ID!
594 } # End of "if CountEVDV= 0" statement
595
596 #_InsCode str( M4b_MaxCoefList_AM[[18]])
597 #####
598 ##### Usual tidying #####
599
600 rm(CoefAll, CoefMaxPos, ConstNames, ConstVal, CountEVDV, EVDVNames, LamRowNum,
601    MaxRes_n, ModEVNames, ModDVNames, ModName, NumEV, NumDV, PosMaxConst, RV_Num, SelMeth,
602    Lambda, Alpha, FTCCode, DVs, EVs, IEV, FTVE_PV, MaxRes_AUC, Test_AUC, GPNName, TotMaxDV, XtraCols)
603 } # end of mod loop
604
605 #####
606 #####
607 ##### All Individual MaxCoef, or M4b_MaxModPar_AM Files coerced into 1 single dataframe #####
608 ##### This saves on files needed to be opened by hand
```

```
678 as.data.frame(names(M4a_MP3RselMods_AM))
679 head(M4a_MP3RselMods_AM)
680 names(M4a_MP3RselMods_AM)[2:6,1:33,10:5,7,17:20,8)])
681 names(M4b_MaxModPar_AM_ModName1st[,c(5:17,9,10:16)])
682 names(M4b_MaxModPar_AM_ModName1st
683 names(M4b_MaxModPar_AM_ModName1st
684 names(M4a_MP3RselMods_AM)
685
686 (M4b_MaxModPar_AM_ModName1st[,c(5:6,7)])
687 (M4b_MaxModPar_AM[,10:11],c(5:6,7)))
688
689 dim(M4b_MaxModPar_AM_ModName1st[,c(5:6,7)])
690 dim(M4b_MaxModPar_AM[,10:11],c(5:6,7)))
691
692
693
694 write.csv(M4b_MaxModPar_AM_ModName1st, # Print only "unique" by model
695 paste(dir.M4,
696 # OptCode "\\",
697 # OptCode "Mod",
698 DAC,
699 "M4b_MaxModPar_AM_ModName1st", ".csv", sep=""), row.names=FALSE)
700
701 names(M4b_MaxModPar_AM_ModName1st)
702 IDAColumns <- c(5, 10:21,23:24,22) # Add also Nn to make sure AM and SM are comparable # -- choose these by hand
703
704 write.csv(M4b_MaxModPar_AM_ModName1st[IDAColumns], # Print just the columns wanted
705 paste(dir.M4,
706 # OptCode "\\",
707 # OptCode "Mod",
708 DAC,
709 "M4b_MaxModPar_AM_ModName1st_SelCol_IDA", ".csv", sep=""), row.names=FALSE)
710
711 ## and another copy in AM to save time
712
713 write.csv(M4b_MaxModPar_AM_ModName1st[IDAColumns], # Print just the columns wanted
714 paste(dir.max,
715 # OptCode "\\",
716 # OptCode "Mod",
717 DAC,
718 "M4b_MaxModPar_AM_ModName1st_SelCol_IDA", ".csv", sep=""), row.names=FALSE)
719
720 save.image(paste(dir.rdata,
721 #IDAC> 0)IDAC),
722 ScriptNameCode,
723 "ModParAll",
724 "End",
725 "RData", sep="")) #
726
727 #####
728 Next Script #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 ##### MIA7 Module 4 - Model extraction and post processing #####
735 ##### Selecting models and creating alternative output #####
736 ##### Reading lambda files for standard MaxEnt (SM) #####
737 ##### Optional #####
738 #####
739
740 ScriptNameCode <- "M4sm" # Counting lambdas for SM
741 ScriptNameCode
742 dir.M5 <- paste(dir.max, "M5_All", "\\", sep="") # In case it had not been done already
743
744 dir.lam.sm <- paste(dir.M5,
745 "Pred_Lam\\",
746 sep="")
747 DirFilesVect <- as.vector(list.files(dir.lam.sm)) #### testing: CSV file list[[5]]
```

```
748 DirFilesVect
749 EVNamesASCIi <-sub("-",asc","", basename(ev.asc))
750
751
752 ### Counting DVs in SM models
753
754 #####
755 #####
756 #####
757 ##### Counting lambdas #####
758 ##### for SM lambda files #####
759 #####
760 #####
761 #####
762 #####
763 M4a_SMNames <- unlist(strsplit(DirFilesVect, "_lam.csv"))
764
765 LamOrigList_SM <- as.list(1:length(M4a_SMNames))
766
767
768 M4b_MaxCoeffList_SM <- list() # Holds all individual variables (and their coefficients) from the lambda files
769 M4b_MaxConstList_SM <- list() # Holds the "last 4 values", from the lambda files (so called "constants")
770
771 M4b_MaxModPar_SM <- list() # To keep all (almost all) MaxEnt model parameters
772 M4b_MaxModPar_SM_Dftrm <- data.frame()
773 M4b_MaxModPar_SM_Dftrm_1st<- data.frame()
774
775 mod<-1
776
777 for (mod in seq(length(M4a_SMNames))) # mod Pred List .. can also just be a position list counter
778 {LamOrigList_SM[[mod]] <-as.data.frame(read.csv(sys.globb(
779 paste(dir.lam.sm,
780 M4a_SMNames[mod],# _OptCode DirFilesVect[mod],
781 "lam.csv", #
782 sep=""))).
783 as.is=TRUE.
784 header=FALSE)}
785
786 # InsCode str(LamOrigList_SM[[mod]])
787 # InsCode head(LamOrigList_SM[[mod]])
788 LamRowNum <- nrow(LamOrigList_SM[[mod]]) # The total number of rows from the original lambda File
789 PosMaxConst <- which(LamOrigList_SM[[mod]]$V1== "linearPredictorNormalizer") # Pick the first
790 M4b_MaxConstList_SM[[mod]] <- LamOrigList_SM[[mod]][PosMaxConst:LamRowNum,1:2] # These are the constants that
791 need to be kept and separated
792
793 CoefAll <- LamOrigList_SM[[mod]][1:(PosMaxConst-1),] # Temporarily, consider all first then just pulls out non-zeros
794 CoefMaxPos <- which(as.numeric(CoefAll[,2])!= 0) # Only takes non-zero values into consideration (### Makes sure
795 coercion as numeric)
796
797 TotMaxDV <- length(CoefMaxPos) # Number of derived variables which are used in the final MaxEnt Model
798
799 M4b_MaxCoeffList_SM [[mod]] <- LamOrigList_SM[[mod]][CoefMaxPos,] # Bring all 4 columns.
799 names(M4b_MaxCoeffList_SM[mod]) <- paste("ModPar_",M4a_SMNames[mod], sep="") #Same as above but No. 1 indicating
800 column - as it is a vector
801
802 CountEVDV <- nrow(M4b_MaxCoeffList_SM[[mod]])
803 EVUsed_SM<-as.character()
804 EVDV_Used<-data.frame()
805 EVUsed<-as.character()
806
807 ev<-1
808 for (ev in seq(length(EVNamesASCIi)))
809 { # InspCode EVNamesASCIi[ev]
809 EVCoeffTemp <- strsplit(M4b_MaxCoeffList_SM[[mod]][,1],fixed=TRUE, # 1 is column 1 where the Name of the feature
809 used is from lambda file
```

```
810 EVNames$ASCI[evj]
811 IntTemp<-strsplit(M4b_MaxCoeffList_SM[[mod]][1:11],"\\") # This one counts products/interactions
812 EVDVNum<-0
813 EVUsed<-as.character(0)
814 IEVNum<-0
815 cnt<-1
816 for(cnt in seq(length(EVCoeffTemp)))
817 {
818   ##### Note: length of EVCoeffTemp[cnt] is not enough, because the portion of EVName can be in more than 1 EV, thus
   must first split then match
819   if(length(EVCoeffTemp[cnt])>1) {EVDVNum<-EVDVNum+1
820   EVUsed <- c(EVUsed,EVNames$ASCI[evj])
821   } # End of if EVCoeffTemp[cnt]]>1 condition means that this EV got used
822
823   if(length(IntTemp[cnt])> 1){print("found interactions")}
824   print(cnt)
825   IEVNum<-IEVNum+1}
826   } # End of cnt -- inspecting EVTemp
827   EVUsed_EVDVNum <- rep(EVDVNum,length(EVUsed)) # Repeat it to match the number of rows for each EV
828   EVUsed_IEVNum <- rep(IEVNum,length(EVUsed)) # Repeat it to match the number of rows for each EV
829   # InsCode class(EVDVTemp)
830   class(EVDVNum)
831   EVDVTemp <- as.data.frame(cbind(EVUsed,EVUsed_EVDVNum,EVUsed_IEVNum))
832   EVDV_Used<- rbind(EVDV_Used,EVDVTemp)
833   } # end of ev
834
835   EVUsed <-unique(EVDV_Used)
836   class(EVUsed)
837   dim(EVUsed)
838
839
840   EVNum SM <- length(EVUsed[1:11])
841   DVNum_SM <- CountEVDV
842   M4b_MaxModPar_SM[[modj]]<- cbind(M4a_SMNames[modj],
843   EVDV_Used,
844   EVNum_SM,
845   DVNum_SM)
846   head(M4b_MaxModPar_SM[[modj]])
847
848   M4b_MaxModPar_SM_Dfrfm_1st <- rbind(M4b_MaxModPar_SM_Dfrfm_1st, M4b_MaxModPar_SM[[modj]][,c(1:6,5:4)]) #
   Just the first line and 3 columns of each list gives the counts of EV and DV totals
849   M4b_MaxModPar_SM_Dfrfm <- rbind(M4b_MaxModPar_SM_Dfrfm, M4b_MaxModPar_SM[[modj]) # This has it all !
850   } # End of mod model
851
852
853   head(M4b_MaxModPar_SM_Dfrfm_1st )
854   write.csv(M4b_MaxModPar_SM_Dfrfm_1st,paste(dir,max,
855   if(length(DAC)>0){DAC},
856   ScriptNameCode,
857   "-", #.row.names=FALSE
858   "MaxModPar_SM_Dfrfm_1st",csv",sep="")) #.row.names=FALSE)
859
860   write.csv(M4b_MaxModPar_SM_Dfrfm,paste(dir,M5,
861   if(length(DAC)>0){DAC},
862   # OptCode "All ",
863   ScriptNameCode,
864   "-",
865   "MaxModPar_SM",
866   "RData",sep=""))
867   save.image(paste(dir,rdata,
868   if(DAC>0){DAC},
869   ScriptNameCode,
870   "MaxModPar_SM",
871   "RData",sep=""))
872
873   #####
874   ##### Next Script #####
875   #####
876   #####
877   #####
```

```
878
879 #####
880 #####
881 #####
882 ## Module 4c - Probability ratio output #####
883 ## Separately for projections options
884 #####
885 #####
886
887 ScriptNameCode <- "M4c"
888
889 #####
890 #####
891 ##### Consolidating all raw values and turning them into PRO values
892 ##### Does this for the SelMods selected
893 ##### Starting with alternative MaxEnt
894 #####
895 #####
896
897 str(M4a_MP3RSelMods_AM)
898 str(M4b_MaxModPar_AM)
899 names(M4a_MP3RSelMods_AM)
900
901
902 str(M4b_MaxModPar_AM)
903 length(M4b_MaxModPar_AM) # One list with one dataframe for each model
904
905 dim(M4b_MaxModPar_AM[1:11]) # The dimensions of the dataframe relate to the number of variables used
906
907 str(M4b_MaxModPar_AM[3:11])
908
909 dim(M4b_MaxModPar_AM_Dfrfm)
910 names(M4b_MaxModPar_AM_Dfrfm)
911 ByModName <-M4b_MaxModPar_AM_Dfrfm[unique(M4b_MaxModPar_AM_Dfrfm$ModName),] # or like this
912 dim(ByModName)
913 head(M4b_MaxModPar_AM_ModName1st)
914 tail(M4b_MaxModPar_AM_ModName1st) # The first line for each model
915
916
917 tail(M4a_MP3RSelMods_AM$MaxPredFile)
918 tail(M4a_MP3RSelMods_AM$MaxProjPredFile)
919 List_MaxPredFile <- M4a_MP3RSelMods_AM$MaxProjPredFile[1:16]] # the full projected predictions
920
921 # Option 2 # From a choice selected copies you did with M4 batch process
922 # List_MaxPredFile <- choose.files("please choose the MaxModPredFiles You want to analyse further",dir.M4)
923
924
925 List_MaxPredFile
926
927 ## ModelNames
928 unique(as.data.frame(M4a_AMNames))
929 # or from M4b
930 # InsCode M4b_MaxModPar_AM_ModName1st$ModName
931
932
933 M4c_AMRaw <- list(0 # MaxEnt raw as "vector" type (swd) rather than raster
934 M4c_AMPRO <- list(0 # Probability Ratio Output as "vector" type (swd) rather than raster
935 M4c_AMRaw_RAS <- list(0 # MaxEnt raw as raster
936 M4c_AMPRO_RAS <- list(0 # Probability Ratio Output as raster
937 str(M4c_AMRaw)
938 ModNames <- vector(0 # This is filled in as the predictions from the MaxEnt models are read, it will be the "response
   variable" with a name that reflects model parameters
939
940 # Choose which N to multiply with
941 ### Option 1: Same value for all
942 # AltCode PRO_N <- as.numeric(readline("Type which N to multiply with ..."))
943 ### Option 2: Count the dataframe by row number of model
944
945 mod<-1
```



```
946 for (mod in seq(length(List_MaxPredFile))) # mod variable list
947 {
948   M4c_AMRaw[mod] <- list(read.csv(List_MaxPredFile[mod])) # Reading MaxEnt's raw predictions
949   # InsCode str(M4c_AMRaw[mod])
950   # _InsCode str(M4c_AMRaw[mod]) #str(M4c_AMRaw[mod]][3])
951   PRO_N <- as.numeric(length(M4c_AMRaw[mod])) # Read by file
952   PROTemp<-M4c_AMRaw[[mod]][3]*as.numeric(PRO_N)
953   M4c_AMPRO[mod] <- list(cbind(M4c_AMRaw[[mod]][1:2],PROTemp))
954   # str(M4c_AMPRO[mod])
955   # str(M4c_AMPRO[[mod]])
956
957   ModNames[mod] <- as.character(strsplit(names(M4c_AMRaw[[mod]][3]),"raw.values")) # Remove the default column name
958   #### User can add this to distinguish more quickly
959   names(M4c_AMRaw[[mod]][1:2]) <- paste(ModNames[mod], " RAW",sep="")#
960   names(M4c_AMPRO[[mod]][1:2]) <- paste(ModNames[mod], "_PRO",sep="")#
961   ## Assign names to list and to the columns accordingly
962   names(M4c_AMRaw[[mod]][3]) <- ModNames[mod]#
963 }
964 str(M4c_AMRaw)
965 str(M4c_AMPRO)
966 mod<-1
967 # InsCode sum(M4c_AMRaw[[mod]][3]) # should be close to 1
968 # InsCode sum(M4c_AMPRO[[mod]][3])
969 str(M4c_AMPRO[mod])
970 names(M4c_AMPRO)
971 str(M4c_AMPRO[[1])
972
973 #####
974 ## The user can choose to save raster files in lists (stacks) or not
975 ## The user can choose to convert them the ASCII files or not
976 ## This option of saving them as ASCII files is useful on many levels,
977 ## including taking advantage of the maxent.jar overlay extraction function
978 ## Options to save the raw or the PRO format data,
979 ## the MaxEnt models only, or the full projected models
980 ##
981
982 require("raster")
983 #require("PresenceAbsence")
984
985 RAW_ASCII_Flag<FALSE # Create the raw raster if TRUE
986 PRO_ASCII_Flag<TRUE # Create the Probability Ratio Output raster if TRUE
987
988 if(RAW_ASCII_Flag==TRUE){
989   for (mod in seq(length(M4c_AMRaw))) #
990   { #####
991     ##### Converting back to ASCII -- useful for overlaying with test data using maxent.jar
992     M4c_AMRaw_RAS[[mod]]<-rasterFromXYZ(M4c_AMRaw[[mod]])
993     writeRaster(M4c_AMRaw_RAS[[mod]],
994               filename=paste(dir.Pred_ASC,"_OptCode_DAC",
995                             "Mod",
996                             if(mod<10){},
997                             " ",
998                             "_",
999                             names(M4c_AMRaw[[mod]][3]),".asc",sep=""),
1000               format="ascii",overwrite=TRUE)
1001     } #End of mod for making rasters
1002   } # RAW_ASCII_Flag
1003
1004   if(PRO_ASCII_Flag==TRUE){
1005     for (mod in seq(length(M4c_AMPRO))) # mod Pred List
1006     { #####
1007       ##### Converting back to ASCII -- useful for overlaying with test data using maxent.jar
1008       M4c_AMPRO_RAS[[mod]] <-rasterFromXYZ(M4c_AMPRO[[mod]])# 1 bracket maintains names better
1009       # str(M4c_AMRaw_RAS[[mod]])
1010       # str(M4c_AMPRO_RAS[[mod]])
1011       writeRaster(M4c_AMPRO_RAS[[mod]],
1012                 filename=paste(dir.Pred_ASCPRO,DAC,names(M4c_AMPRO[[mod]][3]),".asc",sep=""),
1013                 format="ascii",overwrite=TRUE)
1014     } #End of mod for making rasters
1015   } # PRO_ASCII_Flag optional
```

```
1016
1017 str(RVNames)
1018 str(TrainSVD)
1019 str(TTest.Data)
1020 str(TTestFile)
1021
1022 # _OptCode PRO_Stak <- stack(M4c_AMPRO_RAS[1:length(M4c_AMPRO_RAS)])
1023
1024 dim(M3B_MP3RselGP_1st)
1025 (M3B_MP3RselGP_1st$GP_Num==1)
1026 Pos_1<-which(M3B_MP3RselGP_A1$GP_Num==1)
1027 (M3B_MP3RselGP_A1$M3B_MP3RselGP_All$M3B_MP3RselGP_RV1)
1028 # [Pos_1,1:8]
1029 #####
1030 ##
1031 #####
1032 #####
1033 #####
1034 save.image(paste(dir.rdata,
1035
1036
1037 if(DAC>0){DAC,
1038   ScriptNameCode,
1039   #"reduced",
1040   "_SelMod$ _PRO_AM",
1041   ".RData",sep=""})
1042
1043 #####
1044 ## Next script #####
1045 #####
1046 #####
1047 #####
1048 #####
1049 #####
1050 ##### MIA7 Module 4 - Model extraction and post processing #####
1051 ##### Selecting models and creating alternative output #####
1052 #####
1053 ##### M4d_ResPlot.R #####
1054 ##### 2015 #####
1055 ##### Author: Sabrina Mazzoni #####
1056 #####
1057 #####
1058 ## This requires more project specific user input so use with caution
1059 ## Prepare data: what's needed is EV Linear and Quantile Data and PRO
1060
1061 ScriptNameCode <- "M2d" # just a "short bit of ScriptName
1062
1063
1064 M4d_EVLin_AM <- list()
1065 for(rv in seq(length(M2_CoreLists_RV)))
1066 {M4d_EVLin_AM[[rv]] <- M2_CoreLists_RV[[rv]]$Lin}
1067
1068 mod<-1
1069 str(M4d_EVLin_AM[[mod]])
1070
1071 names(M2_CoreLists_RV[[rv]])
1072 names(M2_CoreLists_RV[[rv]][c(1,2,4,25,10)])
1073
1074 M4d_EVQnt <- list() # e
1075 for(rv in length(RVNames))
1076 { M4d_EVQnt[[rv]] <- M2_CoreLists_RV[[rv]][c(1,2,4,25,10)]
1077 }
1078
1079 rv<-1
1080 names(M4d_EVQnt[[rv]])
1081 head(M4d_EVQnt[[rv]][1])
1082
1083
```



```

1084 #####
1085 #####
1086 ## M4 d after quantile, calculate some means
1087 M4d_PRO_EVQnt <- list0
1088 mod<-1
1089
1090 for(mod in seq(length(M4d_PRO)))
1091 {
1092   KeyEv<- as.data.frame(M4d_EV[[mod]])
1093   #head( M4d_EV[[mod]])
1094   #str(M4d_PRO[[mod]])
1095   ModEV <- cbind(M4d_PRO[[mod],keyEv])
1096   str(ModEV)
1097   str(QntClsFr)
1098
1099   QntClsFr <- M4d_EVQnt[[mod]]$QntClsFr
1100   QntClsTo <- M4d_EVQnt[[mod]]$QntClsTo
1101
1102   PRO_EVQnt <- data.frame0
1103   ev<-1
1104   for (ev in seq(length(KeyEV)))
1105   {
1106     # mod<-1 ## this is where we have 1 model per set of ev's always in position 1
1107     i=1
1108     for (i in isqn.meth) ## the quantile class
1109     {
1110       WhichPos<- which((ModEV[,ev+1]> QntClsFr[i,ev]&ModEV[,ev+1]<=QntClsTo[i,ev]) # sum of RV pres x quantile
1111       class
1112       PRO_EVQnt[i,ev]<- mean(ModEV[WhichPos, i]) # The model will always be in position 1 here
1113     } # end of i quantile classes
1114     names(PRO_EVQnt)[ev] <- paste("PRO ",
1115     "Mod",if (mod<0) {},mod,
1116     "_QntEV",ev, sep="")
1117   } #end of ev
1118   str(PRO_EVQnt)
1119   M4d_PRO_EVQnt[[mod]] <- PRO_EVQnt
1120   } # end of mod (which is same as rv in this case)
1121 #####
1122 #####
1123 ### Plotting response curve #####
1124 ### for each model against the original EV's
1125 ## with rectangles xy lim
1127 #####
1128
1129 str(M4d_EV) # x points
1130 str(M4d_EVQnt) # x line
1131 str(M4d_PRO) # y points #list
1132 str(M4d_PRO_EVQnt) # y line
1133 #####
1134 length(M4d_EV) # x points
1135 length(M4d_EVQnt) # x line
1136 length(M4d_PRO) # y points #list
1137 length(M4d_PRO_EVQnt) # y line
1138
1139 M4d_EVQnt[1,rv]$QntClsFr[i,ev]
1140
1141
1142 #mod<-2
1143 str(M4d_EV[[mod]])
1144
1145 dir.M4.resp <- paste(dir.M4, "RespCurv\\", sep="") # Response curve
1146
1147 dir.create(dir.M4.resp)
1148
1149 dir.create(paste(dir.M4.resp, "[N\\", sep=""))
1150
1151
1152

```

```

1153 #####
1154 ##### Note Users must decide how best to group their plots, whether by model numbers or rv
1155 ## Thus the current grouping by rv may need to be changed by users
1156
1157 #for (rv in seq(length(M2_CoreLists_RV)))
1158 {
1159
1160   ### This below is the "scatter of all data points"
1161   LOGFlag<-FALSE #TRUE# FALSE
1162
1163   mod
1164   for(mod in seq(length(M4d_PRO_EVQnt)))
1165   {
1166     PRO_EVQnt <- M4d_PRO_EVQnt[[mod]]
1167     #str(PRO_EVQnt)
1168     EVQnt <- M4d_EVQnt [[mod]] # str(EVQnt)
1169     QntCls <- EVQnt$QntClsMid # Quantile Class or pick 4 for mid point
1170     # QntCls <- EVQnt[[4]] # Quantile Class or pick 4 for mid point
1171
1172     #PLOT ALONE
1173     XLine <- QntCls
1174     # head(XLine)
1175     YLine <- PRO_EVQnt
1176     # head(YLine)
1177     ## Against the full 10,000 pts
1178
1179     # rv<-3 #ev<-1
1180     #_InsCode str(M3_SWD_RV[rv][[ev]][4]) #XPlot <- M3_SWD_RV[rv][[ev]][4]
1181     #_InsCode str(M4b_SelMods_PRO[rv]) #YLine <- M4b_SelMods_PRO[rv]
1182     XPnts <- M4d_EV[[mod]]
1183     YPnts <- as.data.frame(M4d_PRO[[mod]])
1184     #_InsCode str(XPnts)
1185     #_InsCode print(range(XPnts))
1186     #_InsCode print(range(YPnts))
1187     #_InsCode head(YPnts)
1188     #_InsCode class(YPnts)
1189     #_InsCode dim(YPnts)
1190
1191     # y <- YPnts[[]]# for use in ln plot
1192     # str(y)# for use in ln plot
1193     ev<-1
1194
1195     xMax<- 1 ## This is the upper ylim value ... to be hand picked
1196     yMax<- 6 ## This is the upper ylim value ... to be hand picked
1197
1198     for(ev in seq(length(XPnts)))
1199     {
1200       #cat(paste("Plotting Model",mod,"EV",ev))
1201       plot(XPnts[ev],
1202       YPnts[,1], # This will always be 1 column (1 model)
1203       xlim=c(0,xMax), # Easier to remember both the x and y lim for both plot and the rectangle
1204       # OptCode ylim=c(0.5),
1205       ylim= c(0,yMax), #range(YPnts[,1]),
1206       pch=20, cex=3, col="grey", # was 19
1207       xlab=names(XPnts[ev]),
1208       ylab=paste(names(YPnts),sep="")) # Probability Ratio Output (x) # pch= default is empty dot
1209       lines(XLine[ev],YLine[ev],col="black",lwd=2)
1210       rect(0,0,M4d_EVQnt[[mod]]$QntClsFr[i,ev],yMax,density=20,angle=45,col=16,border=NA)
1211       rect(M4d_EVQnt[[mod]]$QntClsTo[0,ev],0,xMax,yMax,density=20,angle=45,col=16,border=NA)
1212       dev.print(win.metafile,filename=paste(dir.M4.resp,
1213       "M4d_",
1214       "EV",ev,"",
1215       names(M4d_PRO[[mod]],
1216       #_OptCode "obj", obj,PageNum, "of",NPages,
1217       ".wmf", sep=""), res=300)
1218       graphics.off0
1219     } # End of ev plots
1220
1221     if(LOGFlag==TRUE){
1222

```

```
1223 #####
1224 #### Testing log option #####
1225 for(ev in seq(length(XPnts)))
1226 { # OptCode cat(paste("Plotting Model",mod,"EV",ev)
1227   plot(XPnts[,ev],
1228     YPnts[,1],
1229     log="y",
1230     # OptCode ylim=c(0.5), # Set limits if want to make better comparisons
1231     ylim= range(YPnts[,1]),
1232     pch=20, cex=3, col="grey", # was 19 #???
1233     xlab=names(XPnts[ev]),
1234     ylab=paste(names(YPnts),sep="")) # Probability Ratio Output (x) # pch= default is empty dot
1235     lines(QntCts[ev],PRO_EVQnt[,ev],col="black",lwd=2)
1236     dev.print(win.metafile.filename=paste(dir.M4,resp,
1237       "LN\\",
1238       "M4d_",
1239       "EV",ev,"_",
1240       "Ln_",
1241       names(M4d_PRO[mod]),
1242       # OptCode "obj", obj,PageNum, "of",NPages,
1243       ".wmf", sep=""), res=300)
1244     graphics.off()
1245   } # End of ev plots
1246 } # Only do this if (LOGFlag==TRUE)
1247 } # End of mod
1248
1249 #####
```



```

128 names(M3A_FinPredData_Dfrfm_AI) <- paste("M3A_FinPredData_Dfrfm_AI",:length(AlphaValues),sep="")
129 names(M3A_FinPredNames_Dfrfm_AI) <- paste("M3A_FinPredNames_Dfrfm_AI",:length(AlphaValues),sep="")
130
131 #####
132 #
133 #      Start of al loop      #####
134 #
135 #####
136 al <-1 # or user can decide
137 rv<-1
138
139 for (al in seq(length(AlphaValues)))
140 {
141   P_ValueCutOff <- MIAPar$AlphaValues[al] # alpha value ... can also be set different to a less or more stringent test # Oct 10
142   ...
143   cat(paste("\nChecking Alpha",al,P_ValueCutOff))
144   #####
145   #####
146   MP3R_RV <- as.list(:length(M3_N_SWD_RV))
147   names(MP3R_RV) <- paste("MP3Rsel_RV",:length(M3_N_SWD_RV),sep="")
148
149   MP3Rsel_RV <- as.list(:length(M3_N_SWD_RV))
150   names(MP3Rsel_RV) <- paste("MP3Rsel_RV",:length(M3_N_SWD_RV),sep="")
151
152   M3A_MP3Rsel_Fin_RV <- as.list(:length(M3_N_SWD_RV))
153   names(M3A_MP3Rsel_Fin_RV) <- paste("M3A_MP3Rsel_Fin_RV",:length(M3_N_SWD_RV),sep="") # That's always the
name of the species
154
155   M3A_MP3Rsel_1st_RV <- as.list(:length(M3_N_SWD_RV)) # In case you want to trace the "best" predictor, the 1st line, for
each individual EV, (and corresponding RV, and AI)
156   names(M3A_MP3Rsel_1st_RV) <- paste("M3A_MP3Rsel_1st_RV",:length(M3_N_SWD_RV),sep="")# That's always the
name of the species
157
158   M3A_FinPredData_List_RV <- as.list(:length(M3_N_SWD_RV))
159   names(M3A_FinPredData_List_RV) <- paste("M3A_FinPredData_List_RV",:length(M3_N_SWD_RV),sep="")
160
161   M3A_FinPredNames_List_RV <- as.list(:length(M3_N_SWD_RV)) ## this only contains the EV chosen (replaces
M3A_FinPredNames_List_RV)
162
163   names(M3A_FinPredNames_List_RV) <-paste("M3A_FinPredNames_List_RV",:length(M3_N_SWD_RV),sep="")
164
165   M3A_FinPredNames_Dfrfm_RV <- as.list(:length(M3_N_SWD_RV)) ## this only contains the EV chosen (replaces
M3A_FinPredNames_List_RV)
166   names(M3A_FinPredNames_Dfrfm_RV) <- paste("M3A_FinPredNames_Dfrfm_RV",:length(M3_N_SWD_RV),sep="")
167
168   M3A_FinPredData_Dfrfm_RV <- as.list(:length(M3_N_SWD_RV)) # as dataframe duplicate of the list above but better for
printing in col names
169   names(M3A_FinPredData_Dfrfm_RV) <- paste("M3A_FinPredData_Dfrfm_RV",:length(M3_N_SWD_RV),sep="")
170
171   #####
172   ##      Start of rv loop      #####
173   rv<-1 # decide where to start
174
175   for (rv in seq(length(M3_N_SWD_RV)))
176   {
177     SWDList <- M3_N_SWD_RV [rv] # with the 3 extra columns again, This is the data with combined rng, zsk, dev, step
4 select pred, and categorical bin data
178     RVXY <- M3_N_SWD_RV [rv,III][2:-1] # THIS is the RVXY for all in Module 3 not just A
179     # unique(RVXY[III])
180     # RVName <- names(RVXY[II]) # original
181     RVName <- paste("RV",rv,"AI",sep=")
182
183     cat(paste("\nModule 3a: Checking",length(SWDList), "grouped variables for rv n.",rv,RVName, "Alpha",al, MIAPar$
P_ValueCutOff))
184
185     dir.M3A_CP <- paste(dir.M3A,
186                       "M3A_al",
187                       al,
188                       "_",

```

```

189   "rv",
190   rv,
191   "\\",
192   sep="") # Single Predictor Analysis for Dataset being used
193   dir.create(dir.M3A_CP)
194
195   MP3R_GPList <- list()
196   MP3Rsel_GPList <- list()
197   M3A_MP3Rsel_Fin<-list()
198   MP3Rsel_1st_EV <- list()
199   M3A_FinPredData_List<- list()
200
201   #####
202   ##### preparing for the maxent output results and picking the best pv combination
203   ##### A dataframe which will store all key information from each MaxEnt run.
204   ##### This will be used in many ways.
205   ##### it will be saved as a List of Dataframes
206   ##### depending on how many levels are considered
207   #####
208   #####
209   #####
210   ##### MaxResNames #####
211   MaxResNames <- as.vector(c("EV_Name",
212                             "GP_Name", # Grouped Predictor
213                             "GP_Num", # number of grouped predictors
214                             "PV_Col", # could be 1 or more ... to see
215                             "RV_NameGP", # individual predictors used, separated by space
216                             "RV_Name", # the refined name which will store model parameter info (will be ModName)
217                             "GP_Best",
218                             "NPred_Used", # not used in 3A but in 3B and 3C -- as it is the same as PV_Used
219                             "NPred_Test", # not used in 3A as it is always 1
220                             "MaxTMLFile",
221                             "MaxPredFile",
222                             "MaxLambdaFile",
223                             "MaxResultsFile",
224                             "MaxRes_E",
225                             "MaxRes_n",
226                             "MaxRes_N",
227                             "MaxRes_AUC",
228                             "FIVE_PV",
229                             "FIVE_PrevPV",
230                             "FIVE_AddPV",
231                             "dtuPV",
232                             "dfePV",
233                             "FRatioTestPV",
234                             "P_ValuePV",
235                             "P_Value_BestPV",
236                             "Name_BestPV",
237                             "Name_AddPV",
238                             "Sign_RankPV",
239                             "Sign_PVNamePV",
240                             "AlphaValue",
241                             "Regulariz")
242   ))
243
244   BestCPNames<-""
245
246   #####
247   #####
248   ##### Running Maxent for individual EV's first #####
249   #####
250   #####
251
252   ip<-1
253
254   for (ip in seq(length(SWDList))) # This creates the new Data for step 2, which is to pick the first (best),

```

```
255 {
256   DataGP <- list()
257   GP <- list()
258   MP3R_GP <- list()
259   MP3Rsel_GP <- list()
260
261   #####
262
263
264   DataGP[[1]] <- as.list(SWDList[[p]][1:length(SWDList[[p])])
265   length(DataGP)
266   str(DataGP[[1]]) # this is the first component of what will be as many gp iterations as needed
267   #length(DataGP[[1]])
268   #str(DataGP[[1]]) # this is the actual contents of the first gp component, which is a list of components itself
269
270   EVName <- names(SWDList[[p])
271   names(DataGP[[1]] <- paste("EV", if(p<10){},lp,"_", names(SWDList[[p]), "_GP", "O", "1, sep=") # this is the first round
272   # str(DataGP)
273
274   #####
275   #####
276   ##### Directories
277   #
278
279   dir.max.ev <- paste(dir.M3A_GP,
280     "EV",
281     if(p<10){},
282     lp,
283     "\\ ",
284     sep=")
285
286   dir.create(dir.max.ev)
287
288   #####
289   #####
290   #####
291   ##### Derived Variables grouped by each Environmental Variable Name
292   ##### Pick up from here after you've selected the best and most significant Grouped Predictors,
293   ##### gp (grouped predictors) loop in other words
294   ##### gp also stands for "iteration" within each Environmental Variable being considered
295   #####
296   #####
297   #####
298   #####
299   #####
300   #####
301   ##### gp loop #####
302   gp<-0
303   CPLoopExit<-FALSE
304   while (CPLoopExit==FALSE)
305   {
306     gp<-gp+1
307
308     RVName <- paste("RV",n,"A",lp,"EV",if(p<10){},lp,
309       "P",if(gp<10){},gp, sep=")
310
311     cat(paste("\nChecking EV",lp,names(SWDList[[p]), "with", (length(SWDList[[p]])-4), "derived variable:")
312
313
314     # cat(paste("\nChecking Alpha",aLP_ValueCutOff, "EV",lp,names(SWDList[[p]]))
```

```
315 # cat("Number of GroupedPredictors to test:",length(DataGP[gp]))
316
317   GP[[gp]] <- as.list(1:length(DataGP[gp])-(gp-1)) # to be used in maxent, will combine predictors as specified by gp run
318   names(GP[gp]) <- paste("EV",if(p<10){},lp,"_", names(SWDList[[p]), "_", "GP", if(gp<10){},gp, sep=")
319   #str(GP)
320
321   for (pv in seq(length(GP[[gp]])) # This creates the new Data for step 2, which is to pick the first (best),
322   {
323     if (gp==1) {GP[[gp]] <- DataGP[[gp]] # this is for the loop version
324     # str(GP)
325
326     if (gp>=2) {
327
328       GP[[gp]][[pv]] <- as.data.frame(cbind(DataGP[[gp]][BestPVNamesGP],DataGP[[gp]][pv+(gp-1)])
329       #head(GP[[gp]][[pv]])
330
331       names(GP[[gp]][pv]) <- paste("EV",if(p<10){},lp,"_", # name of this list
332         "GP",if(gp<10){},gp,"_",
333         "PV",1,"_",if(pv<9){},pv+1, sep=") # decide numbers or names
334
335       } # end of gp>= 2 statement
336
337     } # end of pv in building GP dataset
338     # str(GP)[[gp]]
339     #str(GP)
340
341     #####
342     #####
343     ### preparing the maxent results holder
344     # MP3R<-MP3R GP[gp]
345
346     MP3R <- as.data.frame(array(dim=c(length(GP[[gp]]),length(MaxResNames)))
347     colnames(MP3R) <- MaxResNames
348
349
350     if (gp==1) {ownames(MP3R) <- names(DataGP[[gp]])}
351     if (gp>=2) {ownames(MP3R) <- names(DataGP[[gp]][gp:length(DataGP[gp])])
352     #str(MP3R)
353     #str(MP3R, GP)
354     #str(GP[[gp]])
355     #names(GP[[gp]][pv])
356     #str(GP[[gp]][[pv]][gp]) # THIS is the first column of the first pvcombination in the gp iteration of EV!!!
357     #names(GP[[gp]][[pv]][1:2])
358     #names(GP[[gp]][pv][1:2])
359     #names(GP[[gp]])[[pv]]
360
361     CPName <- names(GP[gp]) # name of list which is a list (or dataframe) of Within Group Predictors (1 or more) that
362     will be tested
363     GPPVNames <- names(GP[[gp]]) # name of individual Predictors that will be tested (gp=1, 1; gp =2, 2 etc), names of the
364     objects inside the list which are your PV (1 or more)
365
366     # names(GP[[gp]]) # oct 8
367
368     # names(GP[[gp]][pv]) # oct 8
369     #names(GP[[gp]][[pv]]) # position pv of the list in position gp
370     #names(GP[[gp]][[pv]][2]) # second position within the pv position of the list in position gp
371     #names(GP[[gp]][[pv]][gp]) # THIS is the third level, which for a list of 1 list, will show the first record
372
373     #str(GP[[gp]])
374
375     #####
376     #####
377
378     #####
379     #####
380     #####
381     #####
382     #####
383     #####
384
385     #####
386     #####
387     #####
388     #####
389     #####
390     #####
391     #####
392     #####
393     #####
394     #####
395     #####
396     #####
397     #####
398     #####
399     #####
400     #####
401     #####
402     #####
403     #####
404     #####
405     #####
406     #####
407     #####
408     #####
409     #####
410     #####
411     #####
412     #####
413     #####
414     #####
415     #####
416     #####
417     #####
418     #####
419     #####
420     #####
421     #####
422     #####
423     #####
424     #####
425     #####
426     #####
427     #####
428     #####
429     #####
430     #####
431     #####
432     #####
433     #####
434     #####
435     #####
436     #####
437     #####
438     #####
439     #####
440     #####
441     #####
442     #####
443     #####
444     #####
445     #####
446     #####
447     #####
448     #####
449     #####
450     #####
451     #####
452     #####
453     #####
454     #####
455     #####
456     #####
457     #####
458     #####
459     #####
460     #####
461     #####
462     #####
463     #####
464     #####
465     #####
466     #####
467     #####
468     #####
469     #####
470     #####
471     #####
472     #####
473     #####
474     #####
475     #####
476     #####
477     #####
478     #####
479     #####
480     #####
481     #####
482     #####
483     #####
484     #####
485     #####
486     #####
487     #####
488     #####
489     #####
490     #####
491     #####
492     #####
493     #####
494     #####
495     #####
496     #####
497     #####
498     #####
499     #####
500     #####
501     #####
502     #####
503     #####
504     #####
505     #####
506     #####
507     #####
508     #####
509     #####
510     #####
511     #####
512     #####
513     #####
514     #####
515     #####
516     #####
517     #####
518     #####
519     #####
520     #####
521     #####
522     #####
523     #####
524     #####
525     #####
526     #####
527     #####
528     #####
529     #####
530     #####
531     #####
532     #####
533     #####
534     #####
535     #####
536     #####
537     #####
538     #####
539     #####
540     #####
541     #####
542     #####
543     #####
544     #####
545     #####
546     #####
547     #####
548     #####
549     #####
550     #####
551     #####
552     #####
553     #####
554     #####
555     #####
556     #####
557     #####
558     #####
559     #####
560     #####
561     #####
562     #####
563     #####
564     #####
565     #####
566     #####
567     #####
568     #####
569     #####
570     #####
571     #####
572     #####
573     #####
574     #####
575     #####
576     #####
577     #####
578     #####
579     #####
580     #####
581     #####
582     #####
583     #####
584     #####
585     #####
586     #####
587     #####
588     #####
589     #####
590     #####
591     #####
592     #####
593     #####
594     #####
595     #####
596     #####
597     #####
598     #####
599     #####
600     #####
601     #####
602     #####
603     #####
604     #####
605     #####
606     #####
607     #####
608     #####
609     #####
610     #####
611     #####
612     #####
613     #####
614     #####
615     #####
616     #####
617     #####
618     #####
619     #####
620     #####
621     #####
622     #####
623     #####
624     #####
625     #####
626     #####
627     #####
628     #####
629     #####
630     #####
631     #####
632     #####
633     #####
634     #####
635     #####
636     #####
637     #####
638     #####
639     #####
640     #####
641     #####
642     #####
643     #####
644     #####
645     #####
646     #####
647     #####
648     #####
649     #####
650     #####
651     #####
652     #####
653     #####
654     #####
655     #####
656     #####
657     #####
658     #####
659     #####
660     #####
661     #####
662     #####
663     #####
664     #####
665     #####
666     #####
667     #####
668     #####
669     #####
670     #####
671     #####
672     #####
673     #####
674     #####
675     #####
676     #####
677     #####
678     #####
679     #####
680     #####
681     #####
682     #####
683     #####
684     #####
685     #####
686     #####
687     #####
688     #####
689     #####
690     #####
691     #####
692     #####
693     #####
694     #####
695     #####
696     #####
697     #####
698     #####
699     #####
700     #####
701     #####
702     #####
703     #####
704     #####
705     #####
706     #####
707     #####
708     #####
709     #####
710     #####
711     #####
712     #####
713     #####
714     #####
715     #####
716     #####
717     #####
718     #####
719     #####
720     #####
721     #####
722     #####
723     #####
724     #####
725     #####
726     #####
727     #####
728     #####
729     #####
730     #####
731     #####
732     #####
733     #####
734     #####
735     #####
736     #####
737     #####
738     #####
739     #####
740     #####
741     #####
742     #####
743     #####
744     #####
745     #####
746     #####
747     #####
748     #####
749     #####
750     #####
751     #####
752     #####
753     #####
754     #####
755     #####
756     #####
757     #####
758     #####
759     #####
760     #####
761     #####
762     #####
763     #####
764     #####
765     #####
766     #####
767     #####
768     #####
769     #####
770     #####
771     #####
772     #####
773     #####
774     #####
775     #####
776     #####
777     #####
778     #####
779     #####
780     #####
781     #####
782     #####
783     #####
784     #####
785     #####
786     #####
787     #####
788     #####
789     #####
790     #####
791     #####
792     #####
793     #####
794     #####
795     #####
796     #####
797     #####
798     #####
799     #####
800     #####
801     #####
802     #####
803     #####
804     #####
805     #####
806     #####
807     #####
808     #####
809     #####
810     #####
811     #####
812     #####
813     #####
814     #####
815     #####
816     #####
817     #####
818     #####
819     #####
820     #####
821     #####
822     #####
823     #####
824     #####
825     #####
826     #####
827     #####
828     #####
829     #####
830     #####
831     #####
832     #####
833     #####
834     #####
835     #####
836     #####
837     #####
838     #####
839     #####
840     #####
841     #####
842     #####
843     #####
844     #####
845     #####
846     #####
847     #####
848     #####
849     #####
850     #####
851     #####
852     #####
853     #####
854     #####
855     #####
856     #####
857     #####
858     #####
859     #####
860     #####
861     #####
862     #####
863     #####
864     #####
865     #####
866     #####
867     #####
868     #####
869     #####
870     #####
871     #####
872     #####
873     #####
874     #####
875     #####
876     #####
877     #####
878     #####
879     #####
880     #####
881     #####
882     #####
883     #####
884     #####
885     #####
886     #####
887     #####
888     #####
889     #####
890     #####
891     #####
892     #####
893     #####
894     #####
895     #####
896     #####
897     #####
898     #####
899     #####
900     #####
901     #####
902     #####
903     #####
904     #####
905     #####
906     #####
907     #####
908     #####
909     #####
910     #####
911     #####
912     #####
913     #####
914     #####
915     #####
916     #####
917     #####
918     #####
919     #####
920     #####
921     #####
922     #####
923     #####
924     #####
925     #####
926     #####
927     #####
928     #####
929     #####
930     #####
931     #####
932     #####
933     #####
934     #####
935     #####
936     #####
937     #####
938     #####
939     #####
940     #####
941     #####
942     #####
943     #####
944     #####
945     #####
946     #####
947     #####
948     #####
949     #####
950     #####
951     #####
952     #####
953     #####
954     #####
955     #####
956     #####
957     #####
958     #####
959     #####
960     #####
961     #####
962     #####
963     #####
964     #####
965     #####
966     #####
967     #####
968     #####
969     #####
970     #####
971     #####
972     #####
973     #####
974     #####
975     #####
976     #####
977     #####
978     #####
979     #####
980     #####
981     #####
982     #####
983     #####
984     #####
985     #####
986     #####
987     #####
988     #####
989     #####
990     #####
991     #####
992     #####
993     #####
994     #####
995     #####
996     #####
997     #####
998     #####
999     #####
1000    #####
```

```
377 ##### Running Maxent
378 ##### grouped predictors (number of derived variables per EV)
379
380
381 dir.max.gp<-paste(dir.max.ev,
382 "GP", #Stands for grouped predictors (number of derived variables per EV) or iterations etc (old SPA1 2 etc)
383 if(gp<10){}, # this helps with the sorting
384 gp,
385 "\",
386 sep="") ## EV or Dat for datasets depending
387 dir.create(dir.max.gp)
388
389
390 #####
391 #####
392 ##### Running Maxent
393 #####
394 #####
395
396 # 2013 pv<-1
397
398 for (pv in seq(length(CP[gp]))) # This creates the new Data for step 2, which is to pick the first (best),
399 {
400   MaxRunDir <- paste(dir.max.gp,
401     names(CP[gp])[pv],
402     sep="")
403   MaxOut <- paste(MaxRunDir,
404     "\\",
405     sep="")
406
407   dir.create(MaxOut)
408   # names(Max.SWD) head(Max.SWD)
409   if (gp == 1) {
410     Max.SWD<-as.data.frame(cbind(RVXY,GP[gp][pv])) # Sep 19 2013 ## THIS IS BIG N, the bgnd in my setup
411     includes training (n)!
412     PVUsed <- ncol(Max.SWD)-3
413     PVNames <- names(GP[gp][pv])
414     PVBest <- NA
415     BestGPNames <- PVBest
416     PVTest <- names(GP[gp][pv])
417     NPred_Test <- length(names(GP[gp][pv][gp]))
418   }
419
420   if (gp >= 2) {
421     Max.SWD<-as.data.frame(cbind(RVXY,GP[gp][pv])) # str(Max.SWD)
422     PVUsed <- ncol(Max.SWD)-3
423     # names(Max.SWD)
424     # names(Max.SWD[length(Max.SWD)-gp])
425     # names(Max.SWD[length(Max.SWD)-gp])
426     PVBest <- MP3R$el_CP[gp-1]$Name_BestPV[pv]
427     BestGPNames<-PVBest
428     PVTest <- names(GP[gp][pv][gp])
429     PVNames<-paste(BestGPNames,PVTest)
430     NPred_Test<- length(names(GP[gp][pv][gp]))
431     # end of if gp loop
432     NPred_Used <- PVUsed # keep for flexibility
433     #####
434     ##### Starting Max.Bkgnd.SWD.Name
435     #####
436
437     cat("\n Iter: ", gp, ", MaxEnt Model: ", RVName, "\n Derived Variables Used: ", na.omit(BestGPNames), PVTest, sep="
438 ")
439
440     ## Sep 19 2013 was: Max.Bkgnd.SWD <- Max.SWD[Max.SWD[1] == "UB",] # Temporary change to have Max.SWD
441     only have UB, not Training as well
442     Max.Bkgnd.SWD <- Max.SWD # the big N, which includes n in this case
443     # str(Max.Bkgnd.SWD)
444     Max.Bkgnd.SWD.Name <- paste(MaxRunDir,
445       "CSV",
```

```
441   sep="")
442   write.csv(Max.Bkgnd.SWD.file==Max.Bkgnd.SWD.Name,row.names=FALSE)
443
444   MaxResFile<-paste(MaxOut,
445     RVName,
446     "Results.csv",
447     sep="") # Assign ModName to Max Results file
448
449   Prespos <-which(NMax.SWD[1]=="UB")
450   Max.Sample<- (Max.SWD[Prespos,1]) # extract only the portion of EV where the RV (always col 1 for SWD)
451   Max.Sample.Name<-paste(MaxRunDir,
452     "\\",
453     RVName,
454     "CSV",
455     sep="")
456   Max.Sample[,1]<-RVName
457   # names(Max.Sample)[1] <- "Sample"
458   write.csv(Max.Sample.file==Max.Sample.Name,row.names=FALSE)
459
460   #####
461   #####
462   ##### Saving parameters in a traceable form so that can be extracted later #####
463   ##### saving results #####
464   MaxHTMLFile <- paste(MaxOut,
465     RVName,
466     ".html",
467     sep="")
468
469   MaxPredFile <- paste(MaxOut,
470     RVName,
471     "CSV",
472     sep="")
473
474   MaxLambdaFile <- paste(MaxOut,
475     RVName,
476     ".lambdas",
477     sep="")
478
479   #####
480   ##### Preparing the Maxent Batch File #####
481   ## with the JarFlags Settings chosen in Module 1
482   MaxRunBat<-paste("java -mx1200m -jar ",
483     jarFileFlag,
484     M3A_JarFlags,
485     " samplesfile=",
486     Max.Sample.Name,
487     " outputdirectory=",
488     MaxOut,
489     " environmentalayers=",
490     Max.Bkgnd.SWD.Name, # this is the "Uninformed Background" file it calls
491     "# not yet " projectionlayers=" ,dir.ev.cnt, # Sep 19 2013 ... incorporate projection files at a later stage
492     " logfile=",
493     RVName,
494     ".log",
495     " autorun",
496     sep="")
497   write(MaxRunBat,paste(MaxRunDir,
498     "Bat",
499     sep=""))
500
501   system(paste(MaxRunBat,wait=TRUE) # running maxent.jar bat file
502
503   #####
504   #####
505   ## Building the MaxOutputsResults Dataframe
506   #####
507   ##### cp #####
508   #####
509   MP3R$EV_Name[pv] <- EVName
510
```



```
511 MP3R$GP_Num[pv] <- gp # iteration number which will tell you how many predictors (number of derived variables
    per EV) are picked
512 MP3R$GP_Name[pv] <- names(CP[[gp]])[pv]
513 MP3R$PV_Coll[pv] <- pv
514 MP3R$PV_NameGP[pv] <- PVNames
515 MP3R$Name_BestPV[pv] <- BestCPNames
516 MP3R$Name_AddPV[pv] <- PVTest
517 MP3R$NPred_Use[dv] <- NPred_Use[dv]
518 MP3R$NPred_Test[pv] <- NPred_Test
519 MP3R$RV_Name[pv] <- RVName
520 MP3R$MaxResultsFile[pv] <- MaxResFile
521 MP3R$MaxHTMLFile[pv] <- MaxHTMLFile
522 MP3R$MaxPredFile[pv] <- MaxPredFile
523 MP3R$MaxLambdaFile[pv] <- MaxLambdaFile
524 MP3R$MaxResultsFile[pv] <- MaxResFile
525 MP3R$AlphaValue[pv] <- P_ValueCutoff
526 MP3R$Regulariz[pv] <- ReguValues[rz]
527
528 } # end of pv loop # inner loop, by column
529
530 cat("\nDone with Maxent Runs\n")
531
532 MP3R_CP[[gp]] <- MP3R
533 names(MP3R_CP[[gp]]) <- names(GP)[gp]
534 # } # end of outer loop, lp, by list position
535 #####
536 #####
537 #####
538 #####
539 #####
540 #####
541 #####
542 #####
543 #####
544 #####
545 #for (lp in seq(length(EVPVList))) # Loop2
546 #{
547
548 MP3R <- MP3R_CP[[gp]]
549
550 #####
551 #####
552 #####
553 # Loop 1
554 # pv <- 2
555 #####
556 #####
557 #####
558 #####
559 #####
560 #####
561 #####
562 #####
563 #####
564 #####
565 #####
566 #####
567 #####
568 #####
569 #####
570 #####
571 #####
572 #####
573 #####
574 #####
575 #####
```

```
576 }
577
578 if (gp>=2) {FTVE_Prev <- MP3R$sel_GP[[gp-1]]$FTVE_PV[1]
579
580 FTVE_Add <- (FTVE-FTVE_Prev)
581
582 dfu <-MaxResN-MaxResN-(PVUsed+2) # Degrees of freedom of the unexplained variation (denominator)
583 # but appear in the numerator(df1)
584 dfe <- PVUsed - (gp-1) # Degrees of freedom of the explained variation(numerator), which appear in
    the denominator (df2)
585 FRatioTest <- ((FTVE_Add*dfu)/((1-FTVE)*dfe)
586
587 }
588
589 P_Value<-as.numeric(1-p(f(RatioTest,dfe,dfu)))
590
591 MP3R$MaxRes_E[pv] <- MaxResE
592 MP3R$MaxRes_n[pv] <- MaxResn
593 MP3R$MaxRes_N[pv] <- MaxResN
594 MP3R$MaxRes_AUC[pv] <- MaxResAUC
595 MP3R$FTVE_PV[pv] <- FTVE
596 if (gp>=2) {MP3R$FTVE_AddPV[pv] <- FTVE_Add
597 MP3R$FTVE_PrevPV[pv] <- FTVE_Prev}
598 MP3R$dfuPV[pv] <- dfu
599 MP3R$dfePV[pv] <- dfe
600 MP3R$FRatioTestPV[pv] <- FRatioTest
601 MP3R$P_ValuePV[pv] <- P_Value
602 MP3R$AlphaValue[pv] <- AlphaValues[a]
603 MP3R$Regulariz[pv] <- ReguValues[rz]
604
605 if (gp==1) {
606 if (MP3R$P_ValuePV[pv] <= P_ValueCutoff)
607 { # Model Selection based on PValue of FTtest
608 MP3R$Sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
609 } # end of model selection if statement
610 } # end of if gp 1 statement
611
612 if (gp>=2) {
613 MP3R$Sign_PVNamePV[pv] <- NA
614 if (MP3R$FTVE_PV[pv] >= MP3R$sel_GP[[gp-1]]$FTVE_PV[1])
615 {
616 MP3R$Sign_PVNamePV[pv] <- MP3R$Name_AddPV[pv]
617 } # end of model selection if statement
618 } # end of if gp 2 statement
619 } # end of pv loop
620
621 #####
622 #####
623 #####
624 #####
625 #####
626 #####
627 #####
628
629 MP3R_CP[[gp]] <- MP3R
630 write.csv(MP3R_CP[[gp]],
631 paste(dir,M3A_GP,
632 "MP3R",
633 # paste("EV".if(lp<10){0},lp," ",names(SWDLList[p]),_", "GP".if(gp<10){0},gp,sep=""),
634 # gp, # "EV",
635 # ",
636 # names(MP3R_CP[[gp]]),
637 # ".csv",
638 sep=""),row.names=F) #
639
```



```
640 SignRows <- which((MP3R_GP[[gp]]$P_ValuePV)<=P_ValueCutOff)
641
642 if (gp== &length(SignRows)==0) {BestPVNamesGP <- NULL
643   AddPVNamesGP <-NULL
644   GLoopExit<-TRUE # no significant improvement or model
645 }
646
647 ## str(MP3RSEL_RV[[2]][1:2]) # significant results gp=2, lp= 12
648 if (gp> &length(SignRows)==0){ BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(MP3Rsel_GP[[gp-1]]$Name_BestPV
  [1,""])))
649   AddPVNamesGP <-NULL
650   MP3Rsel_GP[[gp]] <- MP3Rsel_GP[[gp-1]][,1]
651   names(MP3Rsel_GP[[gp]] <- names(MP3R_GP[[gp])
652 }
653
654 write.csv(MP3Rsel_GP[[gp]].
655   paste(dir.M3A_GP,
656     "_",
657     names(MP3Rsel_GP[[gp]]),
658     ".csv",
659     sep="").row.names=F)
660
661
662 M3A_FinPredData_List[[length(M3A_FinPredData_List)+1]] <- as.list(DataGP[[gp]][c(
663   BestPVNamesGP,AddPVNamesGP)])
664   names(M3A_FinPredData_List)[length(M3A_FinPredData_List)] <- names(SWDLst[[lp])
665   M3A_MP3Rsel_Fin[[length(M3A_MP3Rsel_Fin)+1]] <- MP3Rsel_GP[[gp-1]][,1]
666   GLoopExit<-TRUE # no significant improvement or model
667 }
668
669 if (gp>= &length(SignRows)==1){MP3Rsel_GP[[gp]] <- MP3R_GP[[gp]][SignRows,]
670   names(MP3Rsel_GP[[gp]] <- names(MP3R_GP[[gp])
671 }
672
673 BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(MP3Rsel_GP[[gp]]$PV_NameGP[1,""])))
674   AddPVNamesGP <- NULL
675   M3A_FinPredData_List[[length(M3A_FinPredData_List)+1]] <- as.list(DataGP[[gp]][c(
676     BestPVNamesGP,AddPVNamesGP)])
677   names(M3A_FinPredData_List)[length(M3A_FinPredData_List)] <- names(SWDLst[[lp])
678   M3A_MP3Rsel_Fin[[length(M3A_MP3Rsel_Fin)+1]] <- MP3Rsel_GP[[gp]][,1]
679 }
680
681 write.csv(MP3Rsel_GP[[gp]].
682   paste(dir.M3A_GP,
683     "_",
684     names(MP3Rsel_GP[[gp]]),
685     ".csv",
686     sep="").row.names=F) #
687   GLoopExit<-TRUE # no significant improvement or model
688 }
689
690 if (gp>= &length(SignRows)>1) {# rearrange them first
691   MP3Rsel <- MP3R_GP[[gp]][SignRows,]
692   MP3Rsel_OrderPValueRatio <- MP3Rsel[order(MP3Rsel$P_ValuePV,-MP3Rsel$FRatioTestPV)] # first by PValue,
  then by FratioTest
693   MP3Rsel_OrderPValueRatio$Sign_RankPV <- 1:nrow(MP3Rsel_OrderPValueRatio) # assigning order value
694   MP3Rsel_OrderPValueRatio$P_Value_BestPV <-MP3Rsel_OrderPValueRatio$P_ValuePV[[1]] # the one that come
  number 1 is the best ...no need to do min or max
695   MP3Rsel_OrderPValueRatio$Name_BestPV<-MP3Rsel_OrderPValueRatio$PV_NameGP[[1]
696   MP3Rsel_GP[[gp]] <- MP3Rsel_OrderPValueRatio
697   names(MP3Rsel_GP[[gp]] <- names(MP3R_GP[[gp])
698 }
699
700 BestPVNamesGP <- unlist(as.vector(na.omit(strsplit(MP3Rsel_GP[[gp]]$Name_BestPV[1,""])))
701   AddPVNamesGP <- as.vector(na.omit(MP3Rsel_GP[[gp]]$Name_AddPV[2:nrow(MP3Rsel_GP[[gp])]))
702 }
703
704 #####
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
741 #####
742 #####
743 #####
744 #####
745 #####
746 #####
747 #####
748 #####
749 #####
750 #####
751 #####
752 #####
753 #####
754 #####
755 #####
756 #####
757 #####
758 #####
759 #####
760 #####
```

```
701 #####
702 #####
703 ##### Now Create new datasets for the next iteration
704 ##### gp +1 is picked from the "best"
705 #####
706 #####
707 #####
708 #####
709 #####
710 #####
711 #####
712 #####
713 #####
714 #####
715 #####
716 #####
717 #####
718 #####
719 #####
720 #####
721 #####
722 #####
723 #####
724 #####
725 #####
726 #####
727 #####
728 #####
729 #####
730 #####
731 #####
732 #####
733 #####
734 #####
735 #####
736 #####
737 #####
738 #####
739 #####
740 #####
741 #####
742 #####
743 #####
744 #####
745 #####
746 #####
747 #####
748 #####
749 #####
750 #####
751 #####
752 #####
753 #####
754 #####
755 #####
756 #####
757 #####
758 #####
759 #####
760 #####
761 #####
762 #####
763 #####
764 #####
765 #####
766 #####
767 #####
768 #####
769 #####
770 #####
771 #####
772 #####
773 #####
774 #####
775 #####
776 #####
777 #####
778 #####
779 #####
780 #####
781 #####
782 #####
783 #####
784 #####
785 #####
786 #####
787 #####
788 #####
789 #####
790 #####
791 #####
792 #####
793 #####
794 #####
795 #####
796 #####
797 #####
798 #####
799 #####
800 #####
801 #####
802 #####
803 #####
804 #####
805 #####
806 #####
807 #####
808 #####
809 #####
810 #####
811 #####
812 #####
813 #####
814 #####
815 #####
816 #####
817 #####
818 #####
819 #####
820 #####
821 #####
822 #####
823 #####
824 #####
825 #####
826 #####
827 #####
828 #####
829 #####
830 #####
831 #####
832 #####
833 #####
834 #####
835 #####
836 #####
837 #####
838 #####
839 #####
840 #####
841 #####
842 #####
843 #####
844 #####
845 #####
846 #####
847 #####
848 #####
849 #####
850 #####
851 #####
852 #####
853 #####
854 #####
855 #####
856 #####
857 #####
858 #####
859 #####
860 #####
861 #####
862 #####
863 #####
864 #####
865 #####
866 #####
867 #####
868 #####
869 #####
870 #####
871 #####
872 #####
873 #####
874 #####
875 #####
876 #####
877 #####
878 #####
879 #####
880 #####
881 #####
882 #####
883 #####
884 #####
885 #####
886 #####
887 #####
888 #####
889 #####
890 #####
891 #####
892 #####
893 #####
894 #####
895 #####
896 #####
897 #####
898 #####
899 #####
900 #####
901 #####
902 #####
903 #####
904 #####
905 #####
906 #####
907 #####
908 #####
909 #####
910 #####
911 #####
912 #####
913 #####
914 #####
915 #####
916 #####
917 #####
918 #####
919 #####
920 #####
921 #####
922 #####
923 #####
924 #####
925 #####
926 #####
927 #####
928 #####
929 #####
930 #####
931 #####
932 #####
933 #####
934 #####
935 #####
936 #####
937 #####
938 #####
939 #####
940 #####
941 #####
942 #####
943 #####
944 #####
945 #####
946 #####
947 #####
948 #####
949 #####
950 #####
951 #####
952 #####
953 #####
954 #####
955 #####
956 #####
957 #####
958 #####
959 #####
960 #####
961 #####
962 #####
963 #####
964 #####
965 #####
966 #####
967 #####
968 #####
969 #####
970 #####
971 #####
972 #####
973 #####
974 #####
975 #####
976 #####
977 #####
978 #####
979 #####
980 #####
981 #####
982 #####
983 #####
984 #####
985 #####
986 #####
987 #####
988 #####
989 #####
990 #####
991 #####
992 #####
993 #####
994 #####
995 #####
996 #####
997 #####
998 #####
999 #####
```

```

761 M3A_FinPredData_Dffrm <- cbind(M3A_FinPredData_Dffrm,as.data.frame(M3A_FinPredData_List[[ev]])) # Stick all DV
    predictors together into 1 dataframe

762 M3A_MP3Rsel_1st_GP <- MP3Rsel_GPList[[ev]][1][1]
763 for {gpl in seq(length(MP3Rsel_GPList[[ev]]))}
764 {
765     M3A_MP3Rsel_1st_GPL[gpl] <- MP3Rsel_GPList[[ev]][[gpl]][1]
766 } # end of M3A_MP3Rsel_1st_GP
767
768 #head(MM3A_MP3Rsel_1st_GP) #head(MP3Rsel_GPList[[ev]])
769 M3A_MP3Rsel_1st_EV[[ev]] <- unique(M3A_MP3Rsel_1st_GP)
770 # names(M3A_MP3Rsel_1st_EV)[ev] <- names(M3A_FinPredData_List[ev])
771
772 } # end of ev loop, assigning results
773
774 # str(M3A_FinPredData_Dffrm)
775 M3A_FinPredData_Dffrm <- M3A_FinPredData_Dffrm[-1]
776
777 } # end of if no predictors are found don't go in here
778 ##### Saving stuff to RV
779
780 MP3R_RV[[rv]] <- MP3R_GPList
781 MP3Rsel_RV[[rv]] <- MP3Rsel_GPList
782
783 M3A_MP3Rsel_Fin_RV[[rv]] <- M3A_MP3Rsel_Fin
784 M3A_MP3Rsel_1st_RV[[rv]] <- M3A_MP3Rsel_1st_EV
785
786 M3A_FinPredData_List_RV[[rv]] <- M3A_FinPredData_List
787 M3A_FinPredNames_List_RV[[rv]] <- M3A_FinPredNames_List
788 M3A_FinPredNames_Dffrm_RV[[rv]] <- M3A_FinPredNames_Dffrm
789 M3A_FinPredData_Dffrm_RV[[rv]] <- M3A_FinPredData_Dffrm
790
791 write.csv(M3A_FinPredNames_Dffrm_RV[[rv]],
792          paste(dir,M3A_
793                "M3A_FinPredNames",
794                "n_
795                "al",
796                al,
797                RVName,
798                ".csv",
799                sep="").row.names=FALSE)
800
801 write.csv(M3A_FinPredData_Dffrm_RV[[rv]],
802          paste(dir,M3A_
803                "M3A_FinPred_Dffrm",
804                "n_
805                "al",
806                al,
807                RVName,
808                ".csv",
809                sep="").row.names=FALSE) #
810 #####
811 ##### Tidying up in rv loop ...
812 #####
813 #####
814 #####
815
816 if (saveTempFlag==TRUE) {save.image(paste(dir,rdata,
817                                           "tmp_",
818                                           ScriptNameCode,"_",
819                                           "A",al,"RV",rv,"RData",sep=""))}
820
821 rm(AddPVNamesGP, BestGPNames, BestPVNamesGP, DataGP, dfe,dfu,
822     EVName,SVDList,
823     FRatioTest, FTVe, FTVe_Add, FTVe_Prev,
824     gp, GP_GPLoopExit, GPName, GPPVNames, lp, MaxSample, MaxSampleName, Max.SWD,Max.Bkgnd.SWD,
825     Max.Bkgnd.SWD.Name,
826     MaxOut, MaxRes, MP3R, MP3R_GP, MP3R_GPList, MaxResAUC, MaxResE, MaxResFile, MaxResN,
827     MaxResNames, MaxResT,
828     MaxRunDir, MP3Rsel_GP, MP3Rsel_GPList, MP3Rsel_OrderPValueFRatio, SignRows,
829     PVBest, PVNames, PVTest, PVUced, P_Value,

```

```

828 RVXY,M3A_FinPred_DVName,M3A_FinPred_EVName,
829 MP3Rsel_1st_EV,M3A_MP3Rsel_1st_GP,
830 M3A_FinPredData_List,M3A_FinPredNames_List,M3A_FinPredNames_Dffrm,M3A_FinPredData_Dffrm)
831 } ##### end of rv loop #####
832
833 MP3R_AI[[al]] <- MP3R_RV
834 MP3Rsel_AI[[al]] <- MP3Rsel_RV
835 M3A_MP3Rsel_Fin_AI[[al]] <- M3A_MP3Rsel_Fin_RV
836 M3A_MP3Rsel_1st_AI[[al]] <- M3A_MP3Rsel_1st_RV
837
838 M3A_FinPredData_List_AI[[al]] <- M3A_FinPredData_List_RV
839 M3A_FinPredNames_List_AI[[al]] <- M3A_FinPredNames_List_RV
840
841 M3A_FinPredData_Dffrm_AI[[al]] <- M3A_FinPredData_Dffrm_RV
842 M3A_FinPredNames_Dffrm_AI[[al]] <- M3A_FinPredNames_Dffrm_RV
843 #####
844 ##### Tidying up in al loop ...
845 #####
846 #####
847 rm(M3A_FinPredData_List_RV,M3A_FinPredNames_List_RV,
848     M3A_FinPredData_Dffrm_RV,M3A_FinPredNames_Dffrm_RV,
849     M3A_MP3Rsel_Fin_RV,P_ValueCutOff,
850     MP3R_RV,MP3Rsel_RV)
851 } # al loop oct 23 pm
852
853 M3A_MP3R_AI <- MP3R_AI # keep all MP3R_ Considered for Step 5
854 M3A_MP3Rsel_AI <- MP3Rsel_AI # keep all SignResAllConsidered for Step 5
855 rm(MP3R_AI,MP3Rsel_AI,Dffrm,M3A_MP3Rsel_Fin,Presop)
856
857 names(M3A_MP3R_AI) <- paste("M3A_MP3R_AI",1:length(AlphaValues),sep="")
858 names(M3A_MP3Rsel_AI) <- paste("M3A_MP3Rsel_AI",1:length(AlphaValues),sep="")
859 names(M3A_MP3Rsel_Fin_AI) <- paste("M3A_MP3Rsel_Fin_AI",1:length(AlphaValues),sep="")
860 names(M3A_MP3Rsel_1st_AI) <- paste("M3A_MP3Rsel_1st_AI",1:length(AlphaValues),sep="")
861
862 names(M3A_FinPredData_List_AI) <- paste("M3A_FinPredData_List_AI",1:length(AlphaValues),sep="")
863 names(M3A_FinPredNames_List_AI) <- paste("M3A_FinPredNames_List_AI",1:length(AlphaValues),sep="")
864
865 save.image(paste(dir,rdata,
866               "if(DAC>0){DAC",
867               ScriptNameCode,"_",
868               "End_AI",RData",sep=""))

```

MaxEnt Software Java Flags in Alphabetical Order and other Scripts Tips

A combination of scripting tips and reorganising of the MaxEnt Software help file (quite simply in alphabetical order) for a more effective and clear use the software in "batch mode", compiled by Sabrina Mazzoni. Contact me (sabgis@yahoo.com) for further questions.

Flag	Abb	Type	Default	Meaning	BAT COMMAND LINE
addallsamplestobackground		boolean	FALSE	Add all samples to the background, even if they have combinations of environmental values that are already present in the background	addallsamplestobackground=FALSE
addsamplestobackground	d	boolean	TRUE	Add to the background any sample for which has a combination of environmental values that isn't already present in the background	addsamplestobackground=TRUE
adjustsampleradius		integer	0	Add this number of pixels to the radius of white/purple dots for samples on pictures of predictions. Negative values reduce size of dots.	adjustsampleradius=0
allowpartialdata		boolean	FALSE	During model training, allow use of samples that have nodata values for one or more environmental variables.	allowpartialdata=FALSE
appendtoresultsfile		boolean	FALSE	If false, maxentResults.csv file is reinitialized before each run	appendtoresultsfile=FALSE
applythresholdrule		string		Apply a threshold rule, generating a binary output grid in addition to the regular prediction grid. Use the full name of the threshold rule in Maxent's html output as the argument. For example, 'applyThresholdRule=Fixed_cumulative value 1'.	applythresholdrule=
askoverwrite	r	boolean	TRUE	If output files already exist for a species being modeled, pop up a window asking whether to overwrite or skip. Default is to overwrite.	askoverwrite=TRUE
autofeature	A	boolean	TRUE	Automatically select which feature classes to use, based on number of training samples	autofeature=TRUE
autorun	a	boolean	FALSE	Start running as soon as the the program starts up	autorun=FALSE
beta_categorical		double	-1	Regularization parameter to be applied to all categorical features; negative value enables automatic setting	beta_categorical=-1
beta_hinge		double	-1	Regularization parameter to be applied to all hinge features; negative value enables automatic setting	beta_hinge=-1
beta_lqp		double	-1	Regularization parameter to be applied to all linear, quadratic and product features; negative value enables automatic setting	beta_lqp=-1
beta_threshold		double	-1	Regularization parameter to be applied to all threshold features; negative value enables automatic setting	beta_threshold=-1
betamultiplier	b	double	1	Multiply all automatic regularization parameters by this number. A higher number gives a more spread-out distribution.	betamultiplier=1
biasfile		file		Sampling is assumed to be biased according to the sampling distribution given in this grid file. Values in this file must not be zero or negative. MaxEnt will factor out the bias.	biasfile=
				Requires environmental data to be in grids, rather than a SWD format file	=
cache		boolean	TRUE	Make a .mxe cached version of ascii files, for faster access	cache=TRUE
convergencethreshold	c	double	0.00001	Stop training when the drop in log loss per iteration drops below this number	convergencethreshold=0.00001
defaultprevalence		double	0.5	Default prevalence of the species: probability of presence at ordinary occurrence points. See Elith et al., Diversity and Distributions, 2011 for details.	defaultprevalence=0.5
					=
doclamp		boolean	TRUE	Apply clamping when projecting	doclamp=TRUE
environmentallayers	e	file/directory		Environmental variables can be in a directory containing one file per variable, or all together in a .csv file in SWD format. Please enter a directory name or file name.	environmentallayers=
extrapolate		boolean	TRUE	Predict to regions of environmental space outside the limits encountered during training	extrapolate=TRUE
fadebyclamping		boolean	FALSE	Reduce prediction at each point in projections by the difference clamped and non-clamped output at that point	fadebyclamping=FALSE
					=
hinge	h	boolean	TRUE	Allow hinge features to be used	hinge=TRUE
hingethreshold		integer	15	Number of samples at which hinge features start being used	hingethreshold=15
jackknife	J	boolean	FALSE	Measure importance of each environmental variable by training with each environmental variable first omitted, then used in isolation	jackknife=FALSE
l2lqthreshold		integer	10	Number of samples at which quadratic features start being used	l2lqthreshold=10
linear	l	boolean	TRUE	Allow linear features to be used	linear=TRUE
logfile		string	maxent.log	File name to be used for writing debugging information about a run in output directory	logfile=maxent.log
logscale		boolean	TRUE	If selected, all pictures of models will use a logarithmic scale for color-coding.	logscale=TRUE
lq2lqptthreshold		integer	80	Number of samples at which product and threshold features start being used	lq2lqptthreshold=80
maximumbackground	MB	integer	10000	If the number of background points / grid cells is larger than this number, then this number of cells is chosen randomly for background	maximumbackground=10000
maximumiterations	m	integer	500	Stop training after this many iterations of the optimization algorithm	maximumiterations=500
nodata	n	integer	-9999	Value to be interpreted as nodata values in SWD sample data	nodata=-9999
outputdirectory	o	directory		Directory where outputs will be written. This should be different from the environmental layers directory.	outputdirectory=
outputfiletype		string	asc	File format used for writing output grids	outputfiletype=asc
outputformat		string	logistic	Representation of probabilities used in writing output grids. See Help for details	outputformat=logistic
outputgrids	x	boolean	TRUE	Write output grids. Turning this off when doing replicate runs causes only the summary grids (average, std deviation etc.) to be written, not those for the individual runs.	outputgrids=TRUE
perspeciesresults		boolean	FALSE	Write separate maxentResults file for each species	perspeciesresults=FALSE
pictures		boolean	TRUE	Create a .png image for each output grid	pictures=TRUE
plots		boolean	TRUE	Write various plots for inclusion in .html output	plots=TRUE

prefixes		boolean	TRUE	When toggling samples or layers selected or layer types, allow toggle string to be a prefix rather than an exact match.		prefixes=TRUE
product	p	boolean	TRUE	Allow product features to be used		product=TRUE
projectionlayers	j	file/directory		Location of an alternate set of environmental variables. Maxent models will be projected onto these variables.		projectionlayers=
				Can be a .csv file (in SWD format) or a directory containing one file per variable.		=
				Multiple projection files/directories can be separated by commas.		=
quadratic	q	boolean	TRUE	Allow quadratic features to be used		quadratic=TRUE
randomseed		boolean	FALSE	If selected, a different random seed will be used for each run, so a different random test/train partition		randomseed=FALSE
				will be made and a different random subset of the background will be used, if applicable.		=
randomtestpoints	X	integer	0	Percentage of presence localities to be randomly set aside as test points, used to compute AUC, omission etc.		randomtestpoints=0
removeduplicates		boolean	TRUE	Remove duplicate presence records.		removeduplicates=TRUE
				If environmental data are in grids, duplicates are records in the same grid cell.		=
				Otherwise, duplicates are records with identical coordinates.		=
replicates		integer	1	Number of replicate runs to do when cross-validating, bootstrapping or doing sampling with replacement runs		replicates=1
replicatetype		string	crossvalidate	If replicates > 1, do multiple runs of this type:		replicatetype=crossvalidate
				Crossvalidate: samples divided into replicates folds; each fold in turn used for test data.		=
				Bootstrap: replicate sample sets chosen by sampling with replacement.		=
				Subsample: replicate sample sets chosen by removing random test percentage without replacement to be used for evaluation.		=
responsecurves	P	boolean	FALSE	Create graphs showing how predicted relative probability of occurrence depends on the value of each environmental variable		responsecurves=FALSE
responsecurvesexponent		boolean	FALSE	Instead of showing the logistic value for the y axis in response curves, show the exponent (a linear combination of features)		responsecurvesexponent=FALSE
samplesfile	s	file		Please enter the name of a file containing presence locations for one or more species.		samplesfile=
skipifexists	S	boolean	FALSE	If output files already exist for a species being modeled,		skipifexists=FALSE
				skip the species without remaking the model.		=
testsamplesfile	T	file		Use the presence localities in this file to compute statistics (AUC, omission etc.) The file can contain different localities for different		testsamplesfile=
				It takes precedence over the random test percentage.		=
threads		integer	1	Number of processor threads to use. Matching this number to the number of cores on your computer speeds up some operations, especially variable jackknifing.		threads=1
threshold		boolean	TRUE	Allow threshold features to be used		threshold=TRUE
togglelayerselected	N	string		Toggle selection of environmental layers whose names begin with this prefix (default: all selected)		togglelayerselected=
togglelayertype	t	string		Toggle continuous/categorical for environmental layers whose names begin with this prefix (default: all continuous)		togglelayertype=
togglespeciesselected	E	string		Toggle selection of species whose names begin with this prefix (default: all selected)		togglespeciesselected=
tooltips		boolean	TRUE	Show messages that explain various parts of the interface, like this message		tooltips=TRUE
verbose	v	boolean	FALSE	Gived detailed diagnostics for debugging		verbose=FALSE
visible	z	boolean	TRUE	Make the Maxent user interface visible		visible=TRUE
warnings		boolean	TRUE	Pop up windows to warn about potential problems with input data.		warnings=TRUE
				Regardless of this setting, warnings are always printed to the log file.		=
writebackgroundpredictions		boolean	FALSE	Write .csv file with predictions at background points		writebackgroundpredictions=FALSE
writelampgrid		boolean	TRUE	Write a grid that shows the spatial distribution of clamping.		writelampgrid=TRUE
				At each point, the value is the absolute difference between prediction values with and without clamping.		=
writemess		boolean	TRUE	A multidimensional environmental similarity surface (MESS) shows where novel climate conditions exist in the projection layers.		writemess=TRUE
				The analysis shows both the degree of novelty and the variable that is most out of range at each point.		=
writeplotdata		boolean	FALSE	Write output files containing the data used to make response curves, for import into external plotting software		writeplotdata=FALSE

List of Papers

PAPER 1. Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt.

Rune Halvorsen, Sabrina Mazzoni, Anders Bryn, and Vegar Bakkestuen.
Ecography (2015), **38**, 172-183. DOI: 10.1111/ecog.00565

PAPER 2. MIAT: Modular R-wrappers for flexible implementation of MaxEnt distribution modelling.

Sabrina Mazzoni, Rune Halvorsen and Vegar Bakkestuen.
Ecological Informatics (2015), **30**, 215-221.
Open Access: <http://dx.doi.org/10.1016/j.ecoinf.2015.07.001>

PAPER 3. Sampling bias in presence-only data used for species distribution modelling: Assessment and effects on models.

Bente Støa, Rune Halvorsen, Sabrina Mazzoni, and Vladimir I. Gusarov.
Sommerfeltia. (In press)

PAPER 4. 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?

Mika Bendiksby, Sabrina Mazzoni, Marte Holten Jørgensen, Rune Halvorsen and Håkon Holien.
Journal of Biogeography (2014), **41**, 2020–2031. DOI: 10.1111/jbi.12347

PAPER 5. How important are choice of model selection method and spatial autocorrelation of presence data for distribution modelling by MaxEnt?

Rune Halvorsen, Sabrina Mazzoni, John W. Dirksen, Erik Næsset, Terje Gobakken and Mikael Ohlson.
Ecological Modelling (2016), **328**, 108-118.
DOI: dx.doi.org/10.1016/j.ecolmodel.2016.02.021

PAPER 6. Optimal MaxEnt model selection strategy evaluated by use of independent test data.

Sabrina Mazzoni, Rune Halvorsen, Vegar Bakkestuen, Inger Auestad, Trine Bekkby, Johannes Breidenbach, Desalegn Chala, John W. Dirksen, Anette Edvardsen, Dag Endresen, Lars Erikstad, Hege Gundersen, Einar Heegaard, Knut Anders Hovstad, Eli Rinde and Anders K. Wollan.
(Manuscript currently under review at *Global Ecology and Biogeography*)

PAPER 1. Opportunities for improved distribution modelling practice via a strict maximum likelihood interpretation of MaxEnt.

Rune Halvorsen, Sabrina Mazzoni, Anders Bryn, and Vegar Bakkestuen.

ECOGRAPHY, 38, 172-183 (2015). doi: 10.1111/ecog.00565

PAPER 2. MIAT: Modular R-wrappers for flexible implementation of MaxEnt distribution modelling.

Sabrina Mazzoni, Rune Halvorsen and Vegar Bakkestuen.

Ecological Informatics (2015). Open Access: [dx.doi.org/10.1016/j.ecoinf.2015.07.001](https://doi.org/10.1016/j.ecoinf.2015.07.001)

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 16 July 2015

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.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

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-Aroita 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 ENMTTools (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).

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

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 MIA 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 1Descriptive 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 Bendiksbj 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
 Dtfm = Dataframe or List
 WI = With Interactions
 RV1= Response Variable 1
 A12= Model selection criteria (alpha 2)
 P03= Number of predictors

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).

Appendix 2. Detailed description of the main components of the MIAT, with “vignettes” (screen shots from practical runs).

Module 1 sets up and specifies the DM procedure by, defining and preparing derived variables (Module 2) and by providing input for parameterization of MaxEnt models by forward subset selection in accordance with the maximum likelihood interpretation of MaxEnt (Module 3). Computationally, modules 2 and 3 use and re-use several standard procedures, among which the most important is the iterative procedure for forward stepwise variable selection (Halvorsen et al. 2015: Fig. 1). These core procedures, which may occur in one or more modules, are described below under headings B and C. Two other set of core procedures, filed under headings A and D, respectively, comprise the generic procedures for input of data and specifications of the modelling procedure (A) and for compilation of key results (D). The key elements of the core procedures are also exemplified by screen-shot ‘vignettes’, a selection of which are numbered and collated in this appendix to illustrate the main features of the MIAT toolbox and are grouped as follows:

- A. Parameter set-up and initialization section (module M1)
 - a. Prepares integration of maxent.jar directly into the scripts, and keep things as simple, similar and descriptive as possible (*vignette 1*).
 - b. ‘Initialization’ of the input parameters provided both in the MIAPar, as well as according to the parameters of the input data (*vignette 2*).
 - c. Interactive lines of code asking users to either type or select with the mouse the necessary files (*vignettes 3, 4*).
 - d. Preparation of R-specific objects (lists) to be used throughout the modelling process; to store parameters and data that belong together (*vignette 5*).
- B. Iteration process to prepare a set of derived variables (DVs) (*vignettes 14, 15*) to represent each environmental variable (EV) (Module 2), or to select EVs for the final model (Module 3)
 - a. The first iteration cycle tests the full set of DVs for each given EV individually (*top halves of vignettes 16, 17*).
 - b. The next iteration cycle tests all individually significant DVs for explained variation additional to that explained by the best DV, etc (*bottom halves of vignettes 16, 17*).
 - c. A similar iteration procedure is applied to test and select EVs (represented by DVs) in Module 3b (*vignettes 18-19,25*), and to test and select interactions between EVs in Module 3c (*vignettes 20-21*).
- C. For each model iteration in modules 2–3, the script:
 - a. creates the appropriate ‘model run’ directory for each iteration; (*vignettes 7,15*)
 - b. writes out, as comma-separated value (.csv) files, the selected DV and response variable (RV) needed to parameterise a MaxEnt model in this working directory; (*top right corner of vignette 12*) and creates a results output folder; (*vignette 14*)
 - c. saves all this information in the form of a Windows batch file (.bat) in accordance with Phillips (2011), for access outside R software (*bottom right corner of vignette 12*) and runs the maxent.jar from inside R (*left side of vignette 12*);
 - d. the concept of embedding metadata is used in the names of each file and folder and also extended to a selection of key maxent.jar standard outputs (maxentResults.csv, *.lambdas, and *.html, where * represents the name of the response variable) so as to improve traceability and interpretability; (seen throughout, including *vignette 14*).
 - e. This information is also stored internally and externally to R (*vignettes 9-13, 26*).
- D. Key information for each MaxEnt model produced in modules 2 and 3 is compiled as:
 - a. A ‘model parameter tracking table’, provided in the form of a dataframe inside R and a .csv table outside R; the dataframe contains the starting parameters, new information from the modelling results, and the results of applying the model performance assessment criteria (*vignettes 22-24*).
 - b. Appropriate results selected and saved in an R list previously initialized, as well as collated for easier access, after being having been selected for further inspection (*vignettes 9, 10, 25, 26*).
 - c. Series of code that cleans up unnecessary objects and saves the entire session or just portions of it in files in .RData format, which can be used as input for the next stages (*vignettes 27*).

The full set of *vignettes*, one *vignette* in each file, is available from ftp://norbif.uio.no/pub/outgoing/_2015_PubAcc/MIAT/ (accessible by copying and pasting the path in any web browser), and are organised in folders by the nested modular structure of the toolbox (modules M1, ... M5 with components; see Table 1; also see Fig. 2). File names contain module and component numbers, a running number (starting at 1 for each component, and some keywords indicating their content). *Vignettes* exemplifying general issues, such as organisation of results etc., are also filed directly under the root directory.

Vignettes 1–2

Script code excerpts showing internal documentation and guidance through the process

```
#####
##
##   Setting the maxent.jar "settings", parameters, flags, options
##
#####

## Breaking down into a "series of simple text objects" that will be "pasted together"
## to make up the "command file" that will define the maxent.jar settings
## Simple set up for now is ability to choose certain objects or not
## Users should consult the maxent.jar help (or the alphabetically sorted table provided here)
## for more info on these maxent.jar flags
## Depending on the script, they are selected and pasted together (with or without a space)
## Make sure there's a space (and no more than 1) at the appropriate place

jarFileFlag <- paste(dir.in, "maxent.jar", sep="")
## checking if maxent can be run (normally not part of your script)
if (file.exists(jarFileFlag)==FALSE){cat("!!!! WARNING !!!! \nPlease place a copy of maxent.jar in your dir.in directory\n!!!! WARNING !!!!")}

jarFileFlag

## this is now its own Parameter Table/object for Maxent.jar flags, but may simplify this when functions are made
##

JarPar <- list()

## to integrate this if desired later: M2g_MaxSetting <- "maxent.jar threads=6 pictures=FALSE writebackgroundpredictions=FALSE autofeature=FALSE

JarPar$JarFlag_AW1 <- " autofeature=FALSE beta_categorical=0 beta_hinge=0 beta_lqp=0 betamultiplier=0 beta_threshold=0 linear=TRUE hinge=FALSE p
JarPar$JarFlag_AW2 <- " addallsamplestobackground=FALSE addsamplestobackground=FALSE maximumbackground=200000" # has to be false as I includ n i
JarPar$JarFlag3 <- " doclamp=FALSE extrapolate=FALSE jackknife=TRUE responsecurves=FALSE perspeciesresults=TRUE randomtestpoints=0 responsecurve
JarPar$JarFlag4 <- " adjustsmpleradius=1 outputformat=raw outputgrids=TRUE writebackgroundpredictions=TRUE writeplotdata=TRUE"
JarPar$JarFlag5 <- " nowarnings redoifexists=TRUE removeduplicates=TRUE skipifexists=FALSE threads=6" # Stuck visible at the end, easier to fin

JarPar$JarFlag_SM1a <- " addallsamplestobackground=TRUE addsamplestobackground=TRUE maximumbackground=10000" ## Ensures N is Backgrnd+n, when
JarPar$JarFlag_SM1b <- " addallsamplestobackground=FALSE addsamplestobackground=TRUE maximumbackground=10000" ## Ensures N is no greater than B
## Currently the rest is set in the "very flexible" M3_SM module
## Like this
JarPar$FTCode <- c("NI", "WI") # Feature Types chosen ... set ... NI for No Interactions, WI for With Interactions, set these 2 for now by hand,
JarPar$MaxSet_SM_FT <- c(" autofeature=FALSE product=FALSE", " autofeature=TRUE") ## just the bit that specifies the feature type, matches the

## users can choose which ones to put together
## very rudimentary example, more options to be developed

JarPar$M3A_JarFlags <- paste(JarPar$JarFlag_AW1, JarPar$JarFlag_AW2, JarPar$JarFlag3, JarPar$JarFlag4, JarPar$JarFlag5, sep="")
JarPar$M3B_JarFlags <- paste(JarPar$JarFlag_AW1, JarPar$JarFlag_AW2, JarPar$JarFlag3, JarPar$JarFlag4, JarPar$JarFlag5, sep="")
JarPar$M3C_JarFlags <- paste(JarPar$JarFlag_AW1, JarPar$JarFlag_AW2, JarPar$JarFlag3, JarPar$JarFlag4, JarPar$JarFlag5, sep="")
```

```
ScriptNameCode <- "M1a" # script name code used to identify where you are in the process
####
#####

# All key parameters in one object
# and by parameters, we actually have several types of parameters:
# 1. Parameters of the tools/scripts
# 2. Parameters of the Transformations
# 3. Parameterisation of the statistical model
# 4. Parameters of the "Maxent runs" and results(from each new model)

#####
####
####   M1a ParNames - some to start with #####
####
#####

M1aParNames <- c("dir.Rlib", # R library location to load packages
"dir.Rlib.num", # Rlibrary location choice (a numeric value the user chooses after all libraries are listed)
"DAC", # Dataset Abbreviation Code, will be used as a prefix to results, (handy if running tools on multiple datasets)
"dir.in", # Starting directory, where to locate the Input Data, and subsequently place all results files and folders
"IntType", # this version works with ASCII and csv files as starting input, other options are possible
# user is asked to split the Explanatory Variables in Continuous and Categorical and identify folder
"dir.ev.cat", # Categorical Environmental Data Folder Name/location
"EVCatNames", # Names of CONTINUOUS Variables Only
"EVCatNames", # Names of CATEGORICAL Variables Only
"EVA1Names", # Name of all Environmental Variables
"TrainFile", # user is asked for input training file to be chosen by list
"TestFile", # user is asked for input test data file if present, otherwise 0
"RVNames", # Response Variable Names
"SubSampleSize", # used in the subsampling of data if records more than 500,000. User can change default
"BTGFile", # Backgroun Target Group files
"MxBkgNum", # Maximum number of background points to be selected to build MaxEnt model with - 10,000 maxent.jar default
"qn.meth", # quantile classes to build deviation predictors - 10, 20, 100 etc
"DevExp", # deviation exponents: default: c(5,1,2)
"num.smoothing", # number of smoothing iterations (3 default)
"knots", # num knots (iterations) for spline objects - readline input the number of knots to inspect
"KValues", # KValue at knot position i # a dataframe with both the n.iterations (the knot), and the value at the knot ()
"AlphaValues", # Internal model assessment/improvement criteria: significance value cut off
"ReguIValues" # regularization value for Lasso's penalty model assessment criteria
)

M1aPar <- as.list(1:length(M1aParNames))
names(M1aPar) <- M1aParNames
```

Vignettes 3-5

Interactive excerpt showing what users will see on the console and requesting them to identify data files/folders locations:

```
This is a set of exploratory data and modelling tools:MIAT.

It will allow you to load raw explanatory variables
(either in raster or vector format) and response variables; explore their distribution, apply rescaling and several
transformations for use as future input to other tools (MaxEnt and or GIS)
... oh, and I guess do just a few other things! ... like building by variable manual forward selection (without the manual part)
> cat("\nYou will be asked a few questions at the beginning, place the cursor on the console then type. The rest the tools will take
care of it!", fill=TRUE,sep="")

You will be asked a few questions at the beginning, place the cursor on the console then type. The rest the tools will take care of
it !
> cat("\nPlease place your input data in one folder (which you will be asked to identify), along with the latest copy of the maxent.
jar file.
+ Within this folder, place explanatory variables in 2 separate folders: 1 for continous and one for categorical data (if the latter
is present)!", fill=TRUE,sep="")

Please place your input data in one folder (which you will be asked to identify), along with the latest copy of the maxent.jar file.
Within this folder, place explanatory variables in 2 separate folders: 1 for continous and one for categorical data (if the latter i
s present)!
> cat("\nseveral options for loading input the data for these tools have been scripted. \nThis script is for loading Maxent software
standard practice data, in ASCII and comma separated \ntext files. Please refer contact developers for additional data input format
scripts", fill=TRUE,sep="")

Several options for loading input the data for these tools have been scripted.
This script is for loading Maxent software standard practice data, in ASCII and comma separated
text files. Please refer contact developers for additional data input format scripts
```

3

4

```
..$ PR_BVCALL : num [1:17479] 0.00772 0.00134 0.003 0.01049 0.00092 ...
..$ PR_BVCS1 : num [1:17479] 0 0 0 0.00202 0 ...
..$ PR_KULT : num [1:17479] 0 0 0 0 0 0 0 0 0 ...
..$ PR_TILANY : num [1:17479] 0 0 0.3898 0.0884 0.1316 ...
..$ PR_TILT4 : num [1:17479] 0 0 0.3898 0.0872 0 ...
..$ PR_VEGIAL : num [1:17479] 0.01351 0.00938 0.006 0.02058 0.0184 ...
..$ PR_VEG2TR : num [1:17479] 0.01351 0.00536 0.006 0.00928 0.01104 ...
..$ TERASPIF : num [1:17479] 180 95 20 111 116 61 24 34 35 35 ...
..$ TERDEM : num [1:17479] 0 0 0 12 0 0 0 11 26 39 ...
..$ TERSLPDG : num [1:17479] 0 0.827 1.433 1.175 0.813 ...
..$ TERSLHEAT : num [1:17479] 0.8394 0.6781 1.0282 0.0694 0.8634 ...
..$ TERSLRADE : num [1:17479] 977 966 1018 956 960 ...
..$ TERSLRADW : num [1:17479] 3772 3759 3817 3750 3752 ...
..$ TERTRP103 : num [1:17479] 0 -4.33 -6.89 3.22 -4.44 ...
..$ TERTRP109 : num [1:17479] -3.34 -7.31 -6.11 -2.38 -15.16 ...
..$ TERTRP115 : num [1:17479] -7.89 -9.21 -7.83 1.56 -10.7 ...
$ FctRaw : 'data.frame': 17479 obs. of 12 variables:
..$ GEOBERG : Factor w/ 16 levels "0","2","3","4",...: 10 10 9 10 10 10 10
..$ GEOLMJA1 : Factor w/ 15 levels "11","12","15",...: 8 7 15 7 8 7 8 7 7 ...
..$ GEOLMJA2 : Factor w/ 17 levels "-17","2","3",...: 10 9 17 9 10 9 10 9 9 ...
..$ LCUARSCAT : Factor w/ 57 levels "1","2","3","4",...: 57 57 57 3 57 49 57 8 4
..$ LCUARSTYP : Factor w/ 9 levels "11","12","21",...: 9 9 9 3 9 7 9 5 7 ...
..$ LCUOR1 : Factor w/ 21 levels "111","112","121",...: 21 21 21 8 21 8 21 8
..$ LCUITL4 : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 2 1 ...
..$ LCUITLANY : Factor w/ 2 levels "0","1": 1 1 2 2 1 1 1 1 2 1 ...
..$ TERLF3X12 : Factor w/ 10 levels "1","2","3","4",...: 4 1 1 8 1 1 1 1 3 3 ...
..$ TERLF3X5 : Factor w/ 9 levels "1","2","3","4",...: 4 1 1 9 1 1 1 1 2 3 ...
..$ TERSLPPS03 : Factor w/ 6 levels "1","2","3","4",...: 3 1 1 6 1 1 1 1 1 1 ...
..$ TERSLPPS09 : Factor w/ 6 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 6 6 ...
$ Ntys : int 8
$ QntCls : int 9
$ QntClsFr : int 10
$ QntClsTo : int 11
$ QntClsMid : int 12
$ RVFrqNtys : int 13
$ RVOptNtys : int 14
$ RVFrqSmthAll : int 15
$ RVFrqSmthLast : int 16
$ RVOptSmthLast : int 17
$ FB1n : int 18
$ Lin : int 19
$ ZSk : int 20
$ Dev : int 21
$ Hfk : int 22
$ HRK : int 23
$ THk : int 24
```

5

Vignettes 6–7

Accessing external tools (maxent.jar) and running it via script.

Vignettes 8–13

Nested modularity and automation: creating objects with names and structure reflecting user input, module that is being run and properties of the data and model parameters, internal and external R objects, and in other formats (.csv, .jpg, .bat, folders)

8

```

..$ JORDART:'data.frame': 198726 obs. of 12 variables:
..$ JORDART_11 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_12 : int [1:198726] 1 1 1 1 1 0 1 1 1 ...
..$ JORDART_15 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_20 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_41 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_42 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_43 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_50 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_72 : int [1:198726] 0 0 0 0 0 1 0 0 0 ...
..$ JORDART_90 : int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_120: int [1:198726] 0 0 0 0 0 0 0 0 0 ...
..$ JORDART_130: int [1:198726] 0 0 0 0 0 0 0 0 0 ...
$ Lin      :'data.frame': 198726 obs. of 6 variables:
..$ ASPECT      : num [1:198726] 0.875 0.901 0.909 0.946 0.859 ...
..$ CONCAVITY    : num [1:198726] 0.497 0.55 0.585 0.602 0.549 ...
..$ DEM10        : num [1:198726] 0.817 0.852 0.872 0.87 0.846 ...
..$ RANDOM_MATRIX : num [1:198726] 0.859 0.923 0.271 0.528 0.567 ...
..$ SLOPE        : num [1:198726] 0.333 0.383 0.292 0.269 0.288 ...
..$ WETNESS      : num [1:198726] 0.133 0.148 0.155 0.184 0.13 ...
$ Zsk        :'data.frame': 198726 obs. of 6 variables:
..$ ASPECT      : num [1:198726] 0.885 0.909 0.916 0.951 0.87 ...
..$ CONCAVITY    : num [1:198726] 0.637 0.684 0.713 0.727 0.683 ...
..$ DEM10        : num [1:198726] 0.911 0.929 0.939 0.938 0.926 ...
..$ RANDOM_MATRIX : num [1:198726] 0.864 0.926 0.279 0.539 0.578 ...
..$ SLOPE        : num [1:198726] 0.698 0.735 0.663 0.641 0.66 ...
..$ WETNESS      : num [1:198726] 0.513 0.537 0.547 0.586 0.507 ...
$ Dev        :List of 5
..$ ASPECT      :'data.frame': 198726 obs. of 3 variables:
..$ ASPECT_Dev1 : num [1:198726] 0.196 0.104 0.051 0.195 0.238 ...
..$ ASPECT_Dev2 : num [1:198726] 0.03918 0.01111 0.00281 0.03863 0.05727 ...
..$ ASPECT_Dev3 : num [1:198726] 1.54e-03 1.23e-04 7.90e-06 1.49e-03 3.28e-03 ...
..$ CONCAVITY    :'data.frame': 198726 obs. of 3 variables:
..$ CONCAVITY_Dev1 : num [1:198726] 0.3927 0.2511 0.0717 0.1574 0.2545 ...
..$ CONCAVITY_Dev2 : num [1:198726] 0.15468 0.06338 0.00527 0.02501 0.06514 ...
..$ CONCAVITY_Dev3 : num [1:198726] 2.39e-02 4.02e-03 2.78e-05 6.26e-04 4.24e-03 ...
..$ RANDOM_MATRIX :'data.frame': 198726 obs. of 3 variables:
..$ RANDOM_MATRIX_Dev1 : num [1:198726] 0.8989 0.9464 0.0873 0.5974 0.6406 ...
..$ RANDOM_MATRIX_Dev2 : num [1:198726] 0.80834 0.89575 0.00787 0.35764 0.41103 ...
..$ RANDOM_MATRIX_Dev3 : num [1:198726] 6.53e-01 8.02e-01 6.19e-05 1.28e-01 1.69e-01 ...
..$ SLOPE        :'data.frame': 198726 obs. of 3 variables:
..$ SLOPE_Dev1 : num [1:198726] 0.2606 0.3713 0.105 0.148 0.0724 ...
..$ SLOPE_Dev2 : num [1:198726] 0.068 0.13799 0.01107 0.02196 0.00528 ...
..$ SLOPE_Dev3 : num [1:198726] 4.62e-03 1.90e-02 1.22e-04 4.82e-04 2.79e-05 ...
..$ WETNESS      :'data.frame': 198726 obs. of 3 variables:
..$ WETNESS_Dev1 : num [1:198726] 0.169 0.213 0.231 0.293 0.157 ...
..$ WETNESS_Dev2 : num [1:198726] 0.0289 0.0458 0.0535 0.0862 0.025 ...
..$ WETNESS_Dev3 : num [1:198726] 0.000833 0.002097 0.00286 0.007423 0.000625 ...
$ Hfk        : int 22
$ Hrk        : int 23
$ Thk        : int 24

```

9

```

List of 1
$ M2_N_SWD_RV1:List of 8
..$ ASPECT      :'data.frame': 198726 obs. of 11 variables:
..$ BkgnID      : num [1:198726] 100884 99427 98455 97967 97477 ...
..$ SWD         : chr [1:198726] "nyresoleie" "nyresoleie" "nyresoleie" "nyresoleie" ...
..$ XCoord      : num [1:198726] 619055 619065 619065 619045 619005 ...
..$ YCoord      : num [1:198726] 7070478 7070508 7070528 7070538 7070548 ...
..$ ASPECT_Lin  : num [1:198726] 0.875 0.901 0.909 0.946 0.859 ...
..$ ASPECT_Zsk  : num [1:198726] 0.885 0.909 0.916 0.951 0.87 ...
..$ ASPECT_Dev1 : num [1:198726] 0.196 0.104 0.051 0.195 0.238 ...
..$ ASPECT_Dev2 : num [1:198726] 0.03918 0.01111 0.00281 0.03863 0.05727 ...
..$ ASPECT_Dev3 : num [1:198726] 1.54e-03 1.23e-04 7.90e-06 1.49e-03 3.28e-03 ...
..$ ASPECT_Hrk17 : num [1:198726] 0 0 0 0 0 ...
..$ ASPECT_Thk13 : num [1:198726] 1 1 1 1 1 1 1 1 1 ...
..$ CONCAVITY    :'data.frame': 198726 obs. of 13 variables:
..$ BkgnID      : num [1:198726] 100884 99427 98455 97967 97477 ...
..$ SWD         : chr [1:198726] "nyresoleie" "nyresoleie" "nyresoleie" "nyresoleie" ...
..$ XCoord      : num [1:198726] 619055 619065 619065 619045 619005 ...
..$ YCoord      : num [1:198726] 7070478 7070508 7070528 7070538 7070548 ...
..$ CONCAVITY_Lin : num [1:198726] 0.497 0.55 0.585 0.602 0.549 ...
..$ CONCAVITY_Zsk : num [1:198726] 0.637 0.684 0.713 0.727 0.683 ...
..$ CONCAVITY_Dev1 : num [1:198726] 0.3927 0.2511 0.0717 0.1574 0.2545 ...
..$ CONCAVITY_Dev2 : num [1:198726] 0.15468 0.06338 0.00527 0.02501 0.06514 ...
..$ CONCAVITY_Dev3 : num [1:198726] 2.39e-02 4.02e-03 2.78e-05 6.26e-04 4.24e-03 ...
..$ CONCAVITY_Hfk07 : num [1:198726] 0.254 0.334 0.385 0.411 0.332 ...
..$ CONCAVITY_Hfk10 : num [1:198726] 0.0414 0.1436 0.2086 0.2425 0.1416 ...
..$ CONCAVITY_Hrk12 : num [1:198726] 0.1361 0.0428 0 0 0.0446 ...
..$ CONCAVITY_Thk11 : num [1:198726] 0 1 1 1 1 0 1 1 1 ...
..$ DEM10       :'data.frame': 198726 obs. of 10 variables:
..$ BkgnID      : num [1:198726] 100884 99427 98455 97967 97477 ...
..$ SWD         : chr [1:198726] "nyresoleie" "nyresoleie" "nyresoleie" "nyresoleie" ...
..$ XCoord      : num [1:198726] 619055 619065 619065 619045 619005 ...
..$ YCoord      : num [1:198726] 7070478 7070508 7070528 7070538 7070548 ...
..$ DEM10_Lin   : num [1:198726] 0.817 0.852 0.872 0.87 0.846 ...
..$ DEM10_Zsk   : num [1:198726] 0.911 0.929 0.939 0.938 0.926 ...
..$ DEM10_Hfk13 : num [1:198726] 0.511 0.606 0.658 0.654 0.59 ...
..$ DEM10_Hrk05 : num [1:198726] 0 0 0 0 0 0 0 0 ...
..$ DEM10_Thk04 : num [1:198726] 1 1 1 1 1 1 1 1 1 ...
..$ DEM10_Thk16 : num [1:198726] 1 1 1 1 1 0 1 1 1 ...
..$ RANDOM_MATRIX :'data.frame': 198726 obs. of 13 variables:
..$ BkgnID      : num [1:198726] 100884 99427 98455 97967 97477 ...
..$ SWD         : chr [1:198726] "nyresoleie" "nyresoleie" "nyresoleie" "nyresoleie" ...
..$ XCoord      : num [1:198726] 619055 619065 619065 619045 619005 ...
..$ YCoord      : num [1:198726] 7070478 7070508 7070528 7070538 7070548 ...
..$ RANDOM_MATRIX_Lin : num [1:198726] 0.859 0.923 0.271 0.528 0.567 ...
..$ RANDOM_MATRIX_Zsk : num [1:198726] 0.864 0.926 0.279 0.539 0.578 ...
..$ RANDOM_MATRIX_Dev1 : num [1:198726] 0.8989 0.9464 0.0873 0.5974 0.6406 ...
..$ RANDOM_MATRIX_Dev2 : num [1:198726] 0.80834 0.89575 0.00787 0.35764 0.41103 ...
..$ RANDOM_MATRIX_Dev3 : num [1:198726] 6.53e-01 8.02e-01 6.19e-05 1.28e-01 1.69e-01 ...
..$ RANDOM_MATRIX_Hfk11 : num [1:198726] 0.70334 0.83864 0 0.00573 0.08837 ...
..$ RANDOM_MATRIX_Hfk18 : num [1:198726] 0 0.387 0 0 0 ...
..$ RANDOM_MATRIX_Thk05 : num [1:198726] 1 1 1 1 1 1 1 1 1 ...
..$ RANDOM_MATRIX_Thk12 : num [1:198726] 1 1 1 0 0 0 1 0 1 ...

```


PCNHM329 ▸ d ▸ IDA ▸ A01_T4 ▸ RData ▸

New folder

Name	Date modified	Type	Size
R A01_M1b5_End_AliObj_Complete.RData	23.04.2014 11:39	R Workspace	10 110 KB
R A01_M1b4_End_AliObj_Complete.RData	23.04.2014 11:40	R Workspace	20 266 KB
R A01_M1b4_End_AliObj_KeyOnes.RData	23.04.2014 11:40	R Workspace	20 266 KB
R A01_M2a_FBin.RData	23.04.2014 11:41	R Workspace	20 597 KB
R A01_M2b_Lin.RData	23.04.2014 11:42	R Workspace	23 328 KB
R A01_M2c_ZSk.RData	23.04.2014 11:42	R Workspace	26 303 KB
R A01_M2d_Dev.RData	23.04.2014 11:43	R Workspace	32 693 KB
R A01_M2f_AliSplinesReady.RData	23.04.2014 11:43	R Workspace	92 572 KB
R A01_M2g_End_AliObj.RData	23.04.2014 12:55	R Workspace	220 869 KB
R A01_M2h_End_BareMin.RData	23.04.2014 12:56	R Workspace	208 837 KB
R A01_M2hEnd_Ali.RData	23.04.2014 12:56	R Workspace	243 564 KB
R A01_M3A_End_Ali.RData	05.05.2014 19:43	R Workspace	230 516 KB
R A01_M3B_aNI_End_Ali_Alphas_RV.RData	05.05.2014 20:39	R Workspace	249 593 KB
R A01_M3B_aNI_EndAIRV_Temp.RData	05.05.2014 20:38	R Workspace	249 585 KB
R A01_M3Bx_REDUCED_5EV_Ali_Prod_Read...	05.05.2014 20:39	R Workspace	253 536 KB
R A01_M3C_aWI_End_Ali_Alphas_RV.RData	05.05.2014 20:51	R Workspace	242 738 KB
R A01_M3C_aWI_EndAIRV_Temp.RData	05.05.2014 20:51	R Workspace	242 737 KB
R A01_M3C_SelMod_AM.RData	23.05.2014 10:13	R Workspace	78 276 KB
R A01_M4a_SelMod_AM.RData	25.05.2014 16:41	R Workspace	84 895 KB
R A01_M4b_ModParAli_End.RData	25.05.2014 17:29	R Workspace	84 907 KB
R A01_M4c_FD_AUC_Ali_ModPar.RData	25.05.2014 22:36	R Workspace	83 310 KB
R A01_M4c_SelMods_PRO_AM.RData	25.05.2014 17:30	R Workspace	83 261 KB
R A01_M4cMaxModPar_SM.RData	25.05.2014 23:21	R Workspace	83 315 KB
R A01_M4cMaxModPar_SM_Dtfrm_1st.RData	25.05.2014 22:38	R Workspace	83 310 KB
R A01_M4smMaxModPar_SM.RData	26.05.2014 11:28	R Workspace	83 318 KB
R A01_M5a_FD_AUC_Ali_ModPar.RData	25.05.2014 18:48	R Workspace	83 257 KB
R A01_M5a_SelMods_PRO_AM.RData	25.05.2014 17:59	R Workspace	83 261 KB
R A01_M5a_Test_AUC.RData	25.05.2014 18:11	R Workspace	83 254 KB

10

M2A_AliNames_FinDVSet		M3A_FinPredNames_AIRV1.csv		M3B_FinPredNames_aNIRV1A1P07.csv		M3C_FinPredNames_aWVRV1A1P06.csv					
#	A	#	A	#	A	#	A	B	C	D	
1	M2A_AliNames	1	M3A_FinPred_EVName	1	M3B_FinPred_EVName	1	M3C_FinPred_EVName	M3C_FinPred_EVName			
2	ASPECT_Lin	2	ASPECT	2	ATIL_23	2	DEMIDISLOPE	EV01EV04_1kx1			
3	ASPECT_ZSk	3	ASPECT	3	ATIL_25	3	DEMIDISLOPE	EV01EV04_1kx2	ASPECTWETNESS	EV02EV03_1kx1	
4	ASPECT_Dev1	4	ASPECT	4	ATIL_26	4	DEMIDISLOPE	EV01EV04_1kx3	ASPECTWETNESS	EV02EV03_1kx2	
5	ASPECT_Dev2	5	CONCAVITY	5	CONCAVITY_HR132	5	ATIL_24	DEMIDISLOPE	EV01EV04_1kx4	ASPECTWETNESS	EV02EV03_1kx3
6	ASPECT_Dev3	6	CONCAVITY	6	CONCAVITY_Dev1	6	ATIL_27	DEMIDISLOPE	EV01EV04_1kx5	ASPECTWETNESS	EV02EV03_1kx4
7	ASPECT_HR137	7	CONCAVITY	7	CONCAVITY_Dev2	7	DEM10	DEMIDISLOPE	EV01EV04_1kx6	ASPECTWETNESS	EV02EV03_1kx5
8	ASPECT_TH133	8	CONCAVITY	8	CONCAVITY_TH133	8	DEM10_HR105	DEMIDISLOPE	EV01EV04_1kx7	ASPECTWETNESS	EV02EV03_1kx6
9	CONCAVITY_Lin	9	DEM10	9	DEM10_TH136	9	DEM10_HR113	DEMIDISLOPE	EV01EV04_1kx8	SLOPERANDOM_MATRIX	EV04EV05_1kx1
10	CONCAVITY_ZSk	10	DEM10	10	DEM10_HR105	10	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx2
11	CONCAVITY_Dev1	11	DEM10	11	DEM10_HR113	11	JORDART	JORDART_43	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx3
12	CONCAVITY_Dev2	12	RANDOM_MATRIX	12	RANDOM_MATRIX_TH105	12	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx4
13	CONCAVITY_Dev3	13	RANDOM_MATRIX	13	RANDOM_MATRIX_TH132	13	JORDART	JORDART_42	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx5
14	CONCAVITY_HR107	14	RANDOM_MATRIX	14	RANDOM_MATRIX_TH136	14	JORDART	JORDART_30	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx6
15	CONCAVITY_HR110	15	SLOPE	15	ASPECT	15	ASPECT	ASPECT_Dev1	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx7
16	CONCAVITY_HR132	16	SLOPE	16	ASPECT_Lin	16	ASPECT	ASPECT_Lin	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx8
17	CONCAVITY_TH133	17	SLOPE	17	SLOPE_HR111	17	ASPECT	ASPECT_TH133	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx9
18	DEM10_Lin	18	SLOPE	18	SLOPE_ZSk	18	WETNESS	WETNESS_Dev2	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx10
19	DEM10_ZSk	19	SLOPE	19	SLOPE_Dev1	19	WETNESS	WETNESS_ZSk	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx11
20	DEM10_HR133	20	WETNESS	20	WETNESS_Dev2	20	SLOPE	SLOPE_TH106	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx12
21	DEM10_HR105	21	WETNESS	21	WETNESS_ZSk	21	SLOPE	SLOPE_ZSk	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx13
22	DEM10_TH104	22	ATIL	22	ATIL_23	22	SLOPE	SLOPE_HR111	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx14
23	DEM10_TH116	23	ATIL	23	ATIL_25	23	SLOPE	SLOPE_ZSk	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx15
24	RANDOM_MATRIX_Lin	24	ATIL	24	ATIL_26	24	SLOPE	SLOPE_Dev1	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx16
25	RANDOM_MATRIX_ZSk	25	ATIL	25	RANDOM_MATRIX_TH105	25	RANDOM_MATRIX	RANDOM_MATRIX_TH105	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx17
26	RANDOM_MATRIX_Dev1	26	ATIL	26	ATIL_27	26	RANDOM_MATRIX	RANDOM_MATRIX_TH132	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx18
27	RANDOM_MATRIX_Dev2	27	JORDART	27	JORDART_12	27	RANDOM_MATRIX	RANDOM_MATRIX_TH136	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx19
28	RANDOM_MATRIX_Dev3	28	JORDART	28	JORDART_43	28	JORDART	JORDART_42	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx20
29	RANDOM_MATRIX_HR111	29	JORDART	29	JORDART_11	29	JORDART	JORDART_30	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx21
30	RANDOM_MATRIX_HR118	30	JORDART	30	JORDART_42	30	JORDART	JORDART_43	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx22
31	RANDOM_MATRIX_TH105	31	JORDART	31	JORDART_30	31	JORDART	JORDART_90	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx23
32	RANDOM_MATRIX_TH112	32	JORDART	32	JORDART_90	32	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx24
33	SLOPE_Lin	33	JORDART	33	JORDART_11	33	JORDART	JORDART_43	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx25
34	SLOPE_ZSk	34	JORDART	34	JORDART_43	34	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx26
35	SLOPE_Dev1	35	JORDART	35	JORDART_11	35	JORDART	JORDART_43	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx27
36	SLOPE_Dev2	36	JORDART	36	JORDART_43	36	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx28
37	SLOPE_Dev3	37	JORDART	37	JORDART_11	37	JORDART	JORDART_43	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx29
38	SLOPE_HR111	38	JORDART	38	JORDART_43	38	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx30
39	SLOPE_HR107	39	JORDART	39	JORDART_11	39	JORDART	JORDART_43	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx31
40	SLOPE_TH106	40	JORDART	40	JORDART_43	40	JORDART	JORDART_11	DEMIDISLOPE	SLOPERANDOM_MATRIX	EV04EV05_1kx32

11

12

```

##### Saving parameters in a traceable form so that can be extracted later #####
MaxHTMLFile <- paste(MaxOut,
  RVName,
  ".html",
  sep="")

MaxPredFile <- paste(MaxOut,
  RVName,
  ".csv",
  sep="")

MaxLambdaFile <- paste(MaxOut,
  RVName,
  "lambdas",
  sep="")

##### Preparing the Maxent Batch File #####
## with the JarFlags Settings chosen in Module 1
MaxRunBat<-paste("java -Xmx512m -jar ", ## user to decide this based
  jarFileFlag,
  JarPar$M3A_JarFlags,
  " samplesfile=",
  Max.Sample.Name,
  " outputdirectory=",
  MaxOut,
  " environmentalayers=",
  Max.Bkgnd.SWD.Name , # this is the "Uninformed Background"
  " logfile=",
  RVName,
  ".log",
  " visible=FALSE autorun",
  sep="")

write(MaxRunBat,paste(MaxRunDir,
  " bat",
  sep=""))

system(paste(MaxRunBat),wait=TRUE) # running maxent.jar bat file

##### Building the MaxOutputsResults Dataframe #####
#####

MP3R$EV_Name[pv] <- EVName
the Maxent Batch File :

```

13

Vignettes 14–15

Flexible integrated traceability throughout the full modelling process, exemplified by m2g, in which the derived variables of the spline type which explain the most variation are selected. The two vignettes show what is displayed on the console and outside R as csv files, respectively.

14

15

Vignettes 16–17

The first of the three phases of forward selection procedure; selecting a parsimonious set of derived variables from each environmental variable

16

```

List of 1
$ :list of 7
..$ PCA2_Lin : num [1:10320] 0.779 0.769 0.758 0.761 0.801 ...
..$ PCA2_Zsk : num [1:10320] 0.768 0.757 0.746 0.749 0.791 ...
..$ PCA2_Hfk04: num [1:10320] 0.733 0.72 0.706 0.71 0.759 ...
..$ PCA2_Hrk14: num [1:10320] 0 0 0 0 0 ...
..$ PCA2_Hrk17: num [1:10320] 0.0554 0.0682 0.0816 0.0781 0.0291 ...
..$ PCA2_Thk09: num [1:10320] 1 1 1 1 1 1 1 1 1 ...
..$ PCA2_Thk13: num [1:10320] 1 1 1 1 1 1 1 1 1 ...

checking EV 2 PCA2 with 7 derived variables
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Lin
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Zsk
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Hfk04
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Hrk14
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Hrk17
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Thk09
Iter: 1, MaxEnt Model: RV01A11E02P01.
Derived Variables Used: PCA2_Thk13
Done with Maxent Runs

checking EV 2 PCA2 with 7 derived variables
Iter: 2, MaxEnt Model: RV01A11E02P02.
Derived Variables Used: PCA2_Hrk17 PCA2_Hfk04
Iter: 2, MaxEnt Model: RV01A11E02P02.
Derived Variables Used: PCA2_Hrk17 PCA2_Hrk14
Iter: 2, MaxEnt Model: RV01A11E02P02.
Derived Variables Used: PCA2_Hrk17 PCA2_Lin
Iter: 2, MaxEnt Model: RV01A11E02P02.
Derived Variables Used: PCA2_Hrk17 PCA2_Zsk
Iter: 2, MaxEnt Model: RV01A11E02P02.
Derived Variables Used: PCA2_Hrk17 PCA2_Thk09
Iter: 2, MaxEnt Model: RV01A11E02P02.
Derived Variables Used: PCA2_Hrk17 PCA2_Thk13
Done with Maxent Runs

checking EV 2 PCA2 with 7 derived variables
Iter: 3, MaxEnt Model: RV01A11E02P03.
Derived Variables Used: PCA2_Hrk17 PCA2_Thk09 PCA2_Thk13
Done with Maxent Runs

```

M3A for rv 1, pvalue = .05 (alpha 1)
PCA2 (ev=2) starts with 7 derived variables
Models parameterised and selected as a
process of manual forward selection.
Three iterations of MaxEnt models run.
Process stops after last significant variable
PCA2 will start M3B with 3 DV (3 splines)

17

```

checking EV 25 LCUAR5CAT with 57 derived variables
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_21
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_1
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_53
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_62
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_50
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_20
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_61
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_40
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_46
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_67
Iter: 4, MaxEnt Model: RV02A11E25P04.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_17
Done with Maxent Runs

checking EV 25 LCUAR5CAT with 57 derived variables
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_21
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_1
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_62
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_53
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_61
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_20
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_46
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_50
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_40
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_67
Iter: 5, MaxEnt Model: RV02A11E25P05.
Derived Variables Used: LCUAR5CAT_32 LCUAR5CAT_60 LCUAR5CAT_80 LCUAR5CAT_56 LCUAR5CAT_17

```

Vignettes 18–19

The second of the three phases of the forward selection procedure (M3B); selection of environmental variables (two different levels of strictness, start and end, shown)

```
Module 3B checking 33 grouped variables for rv n. 0 1 anirv01a1i Alpha 1 0.05
Module 3B iteration: 1 , RV & Alpha: anirv01a1iP01
Number of Grouped Predictors to test: 33
Max Checking 1 Grouped Predictors: PCA1
Max Checking 1 Grouped Predictors: PCA2
Max Checking 1 Grouped Predictors: PCA3
Max Checking 1 Grouped Predictors: PCA4
Max Checking 1 Grouped Predictors: PR_BYGALL
Max Checking 1 Grouped Predictors: PR_BYGSL
Max Checking 1 Grouped Predictors: PR_KULT
Max Checking 1 Grouped Predictors: PR_TILANY
Max Checking 1 Grouped Predictors: PR_TILT4
Max Checking 1 Grouped Predictors: PR_VEGIAL
Max Checking 1 Grouped Predictors: PR_VEG2TR
Max Checking 1 Grouped Predictors: TERASPIF
Max Checking 1 Grouped Predictors: TERDEM
Max Checking 1 Grouped Predictors: TERSLPDG
Max Checking 1 Grouped Predictors: TERSOLRADE
Max Checking 1 Grouped Predictors: TERSOLRADS
Max Checking 1 Grouped Predictors: TERSOLRADW
Max Checking 1 Grouped Predictors: TERTPI03
Max Checking 1 Grouped Predictors: TERTPI09
Max Checking 1 Grouped Predictors: TERTPI15
Max Checking 1 Grouped Predictors: GEOBERG
Max Checking 1 Grouped Predictors: GEOLMJA1
Max Checking 1 Grouped Predictors: GEOLMJA2
Max Checking 1 Grouped Predictors: LCUAR5CAT
Max Checking 1 Grouped Predictors: LCUAR5TYP
Max Checking 1 Grouped Predictors: LCUCORI
Max Checking 1 Grouped Predictors: LCUTIL_T4
Max Checking 1 Grouped Predictors: LCUTILANY
Max Checking 1 Grouped Predictors: TERLF3X12
Max Checking 1 Grouped Predictors: TERLF3X5
Max Checking 1 Grouped Predictors: TERSLPPS03
Max Checking 1 Grouped Predictors: TERSLPPS09
Max Checking 1 Grouped Predictors: TERSLPPS15
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3B iteration: 2 , RV & Alpha: anirv01a1iP02
Number of Grouped Predictors to test: 32
Max Checking 2 Grouped Predictors: PR_BYGSL TERTPI09
Max Checking 2 Grouped Predictors: PR_BYGSL TERTPI15
Max Checking 2 Grouped Predictors: PR_BYGSL TERDEM
Max Checking 2 Grouped Predictors: PR_BYGSL PR_BYGALL
Max Checking 2 Grouped Predictors: PR_BYGSL PR_KULT
Max Checking 2 Grouped Predictors: PR_BYGSL PCA2
Max Checking 2 Grouped Predictors: PR_BYGSL TERSLPPS15
Max Checking 2 Grouped Predictors: PR_BYGSL LCUAR5TYP

18 Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 GEOLMJA1
Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 PCA4
Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERASPIF
Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERSLPPS15
Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERTPI15
Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERSOLRADS
Max Checking 12 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERSOLRADW
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3B iteration: 13 , RV & Alpha: anirv01a1iP13
Number of Grouped Predictors to test: 8
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA1
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 PCA4
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 TERASPIF
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 TERSLPPS15
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 TERTPI15
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 TERSOLRADS
Max Checking 13 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 TERSOLRADW
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3B iteration: 14 , RV & Alpha: anirv01a1iP14
Number of grouped Predictors to test: 7
Max Checking 14 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 GEOLMJA1
Max Checking 14 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4
Max Checking 14 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 TERASPIF
Max Checking 14 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 TERSLPPS15
Max Checking 14 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 TERSOLRADS
Max Checking 14 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 TERSOLRADW
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3B iteration: 15 , RV & Alpha: anirv01a1iP15
Number of grouped Predictors to test: 5
Max Checking 15 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERASPIF
Max Checking 15 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERSLPPS15
Max Checking 15 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERSOLRADS
Max Checking 15 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERTPI15
Max Checking 15 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERSOLRADW
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3B iteration: 16 , RV & Alpha: anirv01a1iP16
Number of Grouped Predictors to test: 2
Max Checking 16 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERASPIF TERSLPPS15
Max Checking 16 grouped Predictors: PR_BYGSL TERTPI09 PR_KULT PCA3 TERDEM LCUAR5TYP LCUCORI GEOBERG PR_BYGALL LCUAR5CAT PCA1 TERLF3X12 GEOLMJA2 PCA4 TERASPIF TERSOLRADS
Done with Maxent Runs

Last significant predictors found, finished with maxent.jar

Module 3B checking 30 grouped variables for rv n. 0 2 anirv02a1i Alpha 1 0.05
Module 3B iteration: 1 , RV & Alpha: anirv02a1iP01
Number of grouped Predictors to test: 30
Max Checking 1 Grouped Predictors: PCA1
Max Checking 1 Grouped Predictors: PCA2
Max Checking 1 Grouped Predictors: PCA3
Max Checking 1 Grouped Predictors: PCA4
Max Checking 1 Grouped Predictors: PR_BYGALL
Max Checking 1 Grouped Predictors: PR_BYGSL
Max Checking 1 Grouped Predictors: PR_KULT
Max Checking 1 Grouped Predictors: PR_TILANY
```

19

Vignettes 20–21

The third of the three phases of the forward selection procedure (M3bx); creating product variables (note that two different datasets are used in the examples)

20

```
> str(M3Bx_Proj_Prod_A1)
List of 2
 $ M3Bx_Proj_Prod_A1:List of 1
  ..$ M3Bx_Proj_Prod_RV1:List of 10
   ..$ DEM10xASPECT      :'data.frame': 198726 obs. of  9 variables:
   .. ..$ EV01xEV02_j1xk1: num [1:198726] 0.196 0.104 0.051 0.195 0.238 ...
   .. ..$ EV01xEV02_j1xk2: num [1:198726] 0.875 0.901 0.909 0.946 0.859 ...
   .. ..$ EV01xEV02_j1xk3: num [1:198726] 1 1 1 1 0 1 1 1 1 ...
   .. ..$ EV01xEV02_j2xk1: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV02_j2xk2: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV02_j2xk3: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV02_j3xk1: num [1:198726] 0.1003 0.0627 0.0336 0.1274 0.1402 ...
   .. ..$ EV01xEV02_j3xk2: num [1:198726] 0.447 0.546 0.598 0.619 0.506 ...
   .. ..$ EV01xEV02_j3xk3: num [1:198726] 0.511 0.606 0.658 0.654 0.59 ...
   ..$ DEM10xWETNESS      :'data.frame': 198726 obs. of  6 variables:
   .. ..$ EV01xEV03_j1xk1: num [1:198726] 0.031 0.0492 0.0574 0.0925 0.0269 ...
   .. ..$ EV01xEV03_j1xk2: num [1:198726] 0.551 0.576 0.587 0.63 0.544 ...
   .. ..$ EV01xEV03_j2xk1: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV03_j2xk2: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV03_j3xk1: num [1:198726] 0.0158 0.0298 0.0378 0.0605 0.0158 ...
   .. ..$ EV01xEV03_j3xk2: num [1:198726] 0.281 0.349 0.387 0.412 0.321 ...
   ..$ DEM10xSLOPE        :'data.frame': 198726 obs. of 15 variables:
   .. ..$ EV01xEV04_j1xk1: num [1:198726] 1 1 1 0 1 0 1 0 1 ...
   .. ..$ EV01xEV04_j1xk2: num [1:198726] 4.62e-03 1.90e-02 1.22e-04 4.82e-04 2.79e-05 ...
   .. ..$ EV01xEV04_j1xk3: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j1xk4: num [1:198726] 0.698 0.735 0.663 0.641 0.66 ...
   .. ..$ EV01xEV04_j1xk5: num [1:198726] 0.2606 0.3713 0.105 0.148 0.0724 ...
   .. ..$ EV01xEV04_j2xk1: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j2xk2: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j2xk3: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j2xk4: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j2xk5: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j3xk1: num [1:198726] 0.511 0.606 0.658 0 0.59 ...
   .. ..$ EV01xEV04_j3xk2: num [1:198726] 2.36e-03 1.15e-02 8.06e-05 3.15e-04 1.64e-05 ...
   .. ..$ EV01xEV04_j3xk3: num [1:198726] 0 0 0 0 0 0 0 0 0 ...
   .. ..$ EV01xEV04_j3xk4: num [1:198726] 0.357 0.446 0.437 0.419 0.389 ...
   .. ..$ EV01xEV04_j3xk5: num [1:198726] 0.1332 0.225 0.0691 0.0967 0.0427 ...
   ..$ DEM10xRANDOM_MATRIX :'data.frame': 198726 obs. of  9 variables:
   .. ..$ EV01xEV05_j1xk1: num [1:198726] 1 1 1 1 1 0 1 1 1 ...
   .. ..$ EV01xEV05_j1xk2: num [1:198726] 1 1 0 0 0 1 0 0 1 ...
   .. ..$ EV01xEV05_j1xk3: num [1:198726] 0 0.387 0 0 0 ...
   .. ..$ EV01xEV05_j2xk1: num [1:198726] 0 0 0 0 0 0 0 0 0
```

21

```
Module 3C Checking 10 grouped variables for rv n. 1 awrv1A1I Alpha 1 0.05
Module 3C iteration: 1 , RV & Alpha: awrv1A1I1P01
Number of Grouped Predictors to test: 10
Max Checking 1 Grouped Predictors: DEM10xASPECT
Max Checking 1 Grouped Predictors: DEM10xWETNESS
Max Checking 1 Grouped Predictors: DEM10xSLOPE
Max Checking 1 Grouped Predictors: DEM10xRANDOM_MATRIX
Max Checking 1 Grouped Predictors: ASPECTxWETNESS
Max Checking 1 Grouped Predictors: ASPECTxSLOPE
Max Checking 1 Grouped Predictors: ASPECTxRANDOM_MATRIX
Max Checking 1 Grouped Predictors: WETNESSxSLOPE
Max Checking 1 Grouped Predictors: WETNESSxRANDOM_MATRIX
Max Checking 1 Grouped Predictors: SLOPExRANDOM_MATRIX
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3C iteration: 2 , RV & Alpha: awrv1A1I1P02
Number of Grouped Predictors to test: 9
Max Checking 2 Grouped Predictors: DEM10xSLOPE DEM10xRANDOM_MATRIX
Max Checking 2 Grouped Predictors: DEM10xSLOPE DEM10xWETNESS
Max Checking 2 Grouped Predictors: DEM10xSLOPE SLOPExRANDOM_MATRIX
Max Checking 2 Grouped Predictors: DEM10xSLOPE DEM10xASPECT
Max Checking 2 Grouped Predictors: DEM10xSLOPE WETNESSxSLOPE
Max Checking 2 Grouped Predictors: DEM10xSLOPE ASPECTxSLOPE
Max Checking 2 Grouped Predictors: DEM10xSLOPE ASPECTxRANDOM_MATRIX
Max Checking 2 Grouped Predictors: DEM10xSLOPE ASPECTxWETNESS
Max Checking 2 Grouped Predictors: DEM10xSLOPE WETNESSxRANDOM_MATRIX
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3C iteration: 3 , RV & Alpha: awrv1A1I1P03
Number of Grouped Predictors to test: 8
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT DEM10xRANDOM_MATRIX
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT SLOPExRANDOM_MATRIX
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT ASPECTxSLOPE
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT ASPECTxRANDOM_MATRIX
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT WETNESSxSLOPE
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT DEM10xWETNESS
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT ASPECTxWETNESS
Max Checking 3 Grouped Predictors: DEM10xSLOPE DEM10xASPECT WETNESSxRANDOM_MATRIX
Done with Maxent Runs

additional significant predictors found, ranking and testing again
Module 3C iteration: 4 , RV & Alpha: awrv1A1I1P04
Number of Grouped Predictors to test: 7
Max Checking 4 Grouped Predictors: DEM10xSLOPE DEM10xASPECT DEM10xRANDOM_MATRIX SLOPExRANDOM_MATRIX
Max Checking 4 Grouped Predictors: DEM10xSLOPE DEM10xASPECT DEM10xRANDOM_MATRIX ASPECTxSLOPE
```

Vignettes 22–23

Model tracking table (highlighted), which contains detailed information of model characteristics extracted from module 3 internal R objects (i.e. type and strictness of model selection criteria, training and background data parameters), and *maxent.jar* results files (lambda, maxent.res) produced.

```

Console D:\IDA\A03_RankA\
> dim(M3C_MP3RSe1GP_1st)
[1] 10 37
> head(M3C_MP3RSe1GP_1st[-37]) # don't visualise the batfile now, otherwise you cant see a thing
  A1_Num RV_Num FTCode MI_Num
3      1      1    awI      1
4      1      1    awI      2
31     1      1    awI      3
32     1      1    awI      4
33     1      1    awI      5
6      1      1    awI      6

GP_Name
3      DEM10xSLOPE
4      DEM10xSLOPE__DEM10xASPECT
31     DEM10xSLOPE__DEM10xASPECT__DEM10xRANDOM_MATRIX
32     DEM10xSLOPE__DEM10xASPECT__DEM10xRANDOM_MATRIX__ASPECTxWETNESS
33     DEM10xSLOPE__DEM10xASPECT__DEM10xRANDOM_MATRIX__ASPECTxWETNESS__SLOPExRANDOM_MATRIX
6      DEM10xSLOPE__DEM10xASPECT__DEM10xRANDOM_MATRIX__ASPECTxWETNESS__SLOPExRANDOM_MATRIX__DEM10xWETNESS

GP_Num  PV_NameGP  RV_Name GP_Best NPred_Used NPred_Test
3      1  GrPr01pv03 awIRV1A11P01      NA      41      41
4      2  GrPr02pv04 awIRV1A11P02      NA      50      9
31     3  GrPr03pv01 awIRV1A11P03      NA      59      9
32     4  GrPr04pv03 awIRV1A11P04      NA      65      6
33     5  GrPr05pv03 awIRV1A11P05      NA      80     15
6      6  GrPr06pv01 awIRV1A11P06      6      86      6

MaxOut
3 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP01\\GrPr01pv3\\
4 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP02\\GrPr02pv4\\
31 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP03\\GrPr03pv1\\
32 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP04\\GrPr04pv3\\
33 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP05\\GrPr05pv3\\
6 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP06\\GrPr06pv1\\

MaxHTMLFile
3 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP01\\GrPr01pv3\\awIRV1A11P01.html
4 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP02\\GrPr02pv4\\awIRV1A11P02.html
31 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP03\\GrPr03pv1\\awIRV1A11P03.html
32 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP04\\GrPr04pv3\\awIRV1A11P04.html
33 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP05\\GrPr05pv3\\awIRV1A11P05.html
6 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP06\\GrPr06pv1\\awIRV1A11P06.html

MaxPredFile
3 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP01\\GrPr01pv3\\awIRV1A11P01.csv
4 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP02\\GrPr02pv4\\awIRV1A11P02.csv
31 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP03\\GrPr03pv1\\awIRV1A11P03.csv
32 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP04\\GrPr04pv3\\awIRV1A11P04.csv
33 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP05\\GrPr05pv3\\awIRV1A11P05.csv
6 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP06\\GrPr06pv1\\awIRV1A11P06.csv

MaxLambdaFile
3 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP01\\GrPr01pv3\\awIRV1A11P01.lambdas
4 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP02\\GrPr02pv4\\awIRV1A11P02.lambdas
31 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP03\\GrPr03pv1\\awIRV1A11P03.lambdas
32 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP04\\GrPr04pv3\\awIRV1A11P04.lambdas
33 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP05\\GrPr05pv3\\awIRV1A11P05.lambdas
6 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP06\\GrPr06pv1\\awIRV1A11P06.lambdas

MaxResultsFile MaxRes_E
3 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP01\\GrPr01pv3\\awIRV1A11P01Results.csv 6.6011
4 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP02\\GrPr02pv4\\awIRV1A11P02Results.csv 6.5505
31 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP03\\GrPr03pv1\\awIRV1A11P03Results.csv 6.5017
32 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP04\\GrPr04pv3\\awIRV1A11P04Results.csv 6.4569
33 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP05\\GrPr05pv3\\awIRV1A11P05Results.csv 6.4056
6 D:\\IDA\\A03_RankA\\AM\\M3C_A11\\M3C_awIRV1A11\\GP06\\GrPr06pv1\\awIRV1A11P06Results.csv 6.3819

MaxRes_n MaxRes_N MaxRes_AUC FTVE_PV FTVE_PrevPV FTVE_AddPV dfuPV dfePV FratioTestPV P_ValuePV
3      54      10054      0.9744 0.5002401 0.4817582 0.018481877 9957 15 24.54833 0
4      54      10054      0.9751 0.5099211 0.5002401 0.009680983 9948 9 21.83467 0
31     54      10054      0.9767 0.5192577 0.5099211 0.009336600 9939 9 21.44750 0
32     54      10054      0.9772 0.5278290 0.5192577 0.008571305 9933 6 30.05224 0
33     54      10054      0.9770 0.5376439 0.5278290 0.009814910 9918 15 14.03597 0
6      54      10054      0.9774 0.5421783 0.5376439 0.004534374 9912 6 16.36179 0

P_Value_BestPV
3      0
Name_BestPV
DEM10xSLOPE

```


Network ▶ Pcnhm329 ▶ D ▶ IDA ▶ A03_RanKA ▶ AM ▶

Burn New folder

Name	Date modified	Type	Size
M2g_All	22.05.2014 14:08	File folder	
M2h_FullSetDV	22.05.2014 14:10	File folder	
M3A_All	22.05.2014 14:44	File folder	
M3B_All	22.05.2014 14:53	File folder	
M3C_All	22.05.2014 15:25	File folder	
M4_All	26.05.2014 12:12	File folder	
M5_All	26.05.2014 12:15	File folder	
A03_M4b_MaxModPar_AM_ModName1st...	26.05.2014 12:12	Microsoft Excel Co...	2 KB
A03_M4sm_MaxModPar_SM_Dtfrm_1st.csv	26.05.2014 12:15	Microsoft Excel Co...	1 KB
A03_M4sm_MaxModPar_SM_Dtfrm_all.csv	26.05.2014 12:15	Microsoft Excel Co...	2 KB
A03_M5a_FD_AUC_ModAll.csv	26.05.2014 12:15	Microsoft Excel Co...	1 KB
A03_ResAll.xlsx	26.05.2014 12:23	Microsoft Excel W...	19 KB
sMaxRes_SM_All.csv	26.05.2014 12:13	Microsoft Excel Co...	14 KB

23

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	ModName	DAC	RV_Num	SelMeth	Lambda	Alpha	FTCode	DVs	EVs	IEV	FTVE_PV	MaxRes_AI Test_AUC	MaxRes_n	numBackgr	GPName						
2	A03_aNIRV1AI2P06	A03_	1 a		0	0.05 aWI		58	13	6	0.542178	0.9774	0	54	10054	DEM10xSLOPE_DEM10xASPECT_DEM10xRANDOM_MATRIX_ASPEC					
3	A03_aNIRV1AI2P07	A03_	1 a		0	0.05 aNI		26	7	4	0.481758	0.9713	0	54	10054	ATIL DEM10_JORDART_ASPECT_WETNESS_SLOPE_RANDOM M					
4	A03_aNIRV1AI2P04	A03_	1 a		0	1.00E-06 aWI		21	9	4	0.481203	0.9706	0	54	10054	DEM10xASPECT_ASPECTxWETNESS_DEM10xSLOPE_DEM10xWETNE					
5	A03_aNIRV1AI2P06	A03_	1 a		0	1.00E-06 aNI		14	6	0	0.464233	0.9677	0	54	10054	ATIL DEM10_ASPECT_JORDART_WETNESS_SLOPE					
6	A03_aNIRV1AI2P04	A03_	1 a		0	1.00E-06 aNI		11	4	0	0.453461	0.9636	0	54	10054	ATIL DEM10_ASPECT_JORDART					
7	A03_aNIRV1AI2P03	A03_	1 a		0	1.00E-06 aNI		7	3	0	0.421721	0.9463	0	54	10054	ATIL DEM10_ASPECT					
8	A03_aNIRV1AI2P02	A03_	1 a		0	1.00E-06 aNI		5	2	0	0.40387	0.9325	0	54	10054	ATIL DEM10					
9	A03_aNIRV1AI2P01	A03_	1 a		0	1.00E-06 aNI		3	1	0	0.345957	0.9056	0	54	10054	ATIL					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19																					
20																					
21																					
22																					
23																					
24																					
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					
34																					
35																					
36																					
37																					
38																					
39																					
40																					
41																					
42																					
43																					
44																					
45																					
46																					
47																					
48																					
49																					
50																					
51																					
52																					
53																					
54																					
55																					
56																					
57																					
58																					
59																					
60																					
61																					
62																					
63																					
64																					
65																					
66																					
67																					
68																					
69																					
70																					
71																					
72																					
73																					
74																					
75																					
76																					
77																					
78																					
79																					
80																					
81																					
82																					
83																					
84																					
85																					
86																					
87																					
88																					
89																					
90																					
91																					
92																					
93																					
94																					
95																					
96																					
97																					
98																					
99																					
100																					
101																					
102																					
103																					
104																					
105																					
106																					
107																					
108																					
109																					
110																					
111																					
112																					
113																					
114																					
115																					
116																					
117																					
118																					
119																					
120																					
121																					
122																					
123																					
124																					
125																					
126																					
127																					
128																					
129																					
130																					
131																					
132																					
133																					
134																					
135																					
136																					
137																					
138																					
139																					
140																					
141																					
142																					
143																					
144																					
145																					
146																					
147																					
148																					
149																					
150																					
151																					
152																					
153																					
154																					
155																					
156																					
157																					
158																					
159																					
160																					
161																					
162																					
163																					
164																					
165																					
166																					
167																					
168																					
169																					
170																					
171																					
172																					
173																					
174																					
175																					
176																					
177																					
178																					
179																					
180																					
181																					
182																					
183																					
184																					
185																					
186																					
187																					
188																					
189																					
190																					
191																					
192																					
193																					
194																					
195																					
196																					
197																					
198																					
199																					
200																					
201																					
202																					
203																					
204																					
205																					
206																					
207																					
208																					
209																					
210																					
211																					
212																					
213																					
214																					
215																					
216																					
217																					
218																					
219																					
220																					
221																					
222																					
223																					
224																					
225																					
226																					
227																					
228																					
229																					
230																					
231																					
232																					
233																					
234																					
235																					
236																					
237																					
238																					
239																					
240																					
241																					
242																					
243																					
244																					
245																					
246																					
247																					
248																					
249																					
250																					
251																					
252																					
253																					
254																					
255																					
256																					
257																					
258																					
259																					
260																					
261																					
262																					
263																					
264																					
265																					
266																					
267																					
268																					
269																					
270																					
271																					
272																					
273																					
274																					
275																					
276																					
277																					
278																					
279																					
280																					
281																					
282																					
283																					
284																					
285																					
286																					
287																					
288																					
289																					
290																					
291																					
292																					
293																					
294																					
295																					
296																					
297																					
298																					
299																					
300																					
301																					
302																					
303																					
304																					
305																					
306																					
307																					
308																					
309																					
310																					
311																					
312																					
313																					
314																					
315																					
316																					
317																					
318																					
319																					
320																					
321																					
322																					
323																					
324																					
325																					
326																					
327																					
328																					
329																					
330																					
331																					
332																					
333																					
334																					
335																					
336																					
337																					
338																					
339																					
340																					
341																					
342																					
343																					
344																					
345																					
346																					
347																					
348																					
349																					
350																					
351																					
352																					
353																					
354																					
355																					
356																					
357																					
358																					
359																					
360																					
361																					
362																					
363																					
364																					
365																					
366																					
367																					
368																					
369																					
370																					
371																					
372																					
373																					
374																					
375																					
376																					
377																					
378																					
379																					
380																					
381																					
382																					
383																					
384																					
385																					
386																					
387																					
388																					
389																					
390																					
391																					
392																					
393																					
394																					
395																					
396																					
397																					
398																					
399																					
400																					
401																					
402																					
403																					
404																					
405																					
406																					
407																					
408																					
409																					
410																					
411																					
412																					
413																					
414																					
415																					
416																					
417																					
418																					
419																					
420																					
421																					
422																					
423																					
424																					
425																					
426																					
427																					
428																					
429																					
430																					
431																					
432																					
433																					
434																					
435																					
436																					
437																					
438																					
439																					
440																					
441																					
442																					
443																					
444																					
445																					
446																					
447																					
448																					
449																					
450																					
451																					
452																					
453																					
454																					
455																					
456																					
457																					
458																					
459																					
460																					
461																					
462																					
463																					
464																					
465																					
466																					
467																					
468																					
469																					
470																					
471																					
472																					
473																					
474																					
475																					
476																					
477																					
478																					
479																					
480																					
481																					
482																					
483																					
484																					
485																					
486																					
487																					
488																					
489																					
490																					
491																					
492																					
493																					
494																					
495																					
496																					
497																					
498																					
499																					
500																					
501																					
502																					
503																					
504																					
505																					
506																					
507																					
508																					
509																					
510																					
511																					
512																					
513																					
514																					
515																					
516																					
517																					
518																					
519																					
520																					
521																					
522																					
523																					
524																					
525																					
526																					
527																					
528																					
529																					
530																					
531																					
532																					
533																					
534																					
535																					
536																					
537																					
538																					
539																					
540																					
541																					
542																					
543																					
544																					
545																					
546																					
547																					
548																					
549																					
550																					

24

Vignette 25

Final models selected in module 4 get collated into one folder, using the model tracking table and the script.

Vignette 27

Series of code that cleans up unnecessary objects and saves the entire session or just portions of it in files in .RData format, which can be used as input for the next stages.

```
323. #####
324. ##### Tidy and Saving Objects to RData workspaces #####
325. #####
326. rm(CSVVect1,CSVVect2) # Some objects not needed to keep all the times
327. save(list=M1a_ParVect, # Saving M1aParameters and Directory objects as "RData" workspaces
328.      file=paste(dir.in,
329.                  "Obj_", # distinguish single objects from entire Images
330.                  ScriptNameCode, # which script? which step, part in the process etc
331.                  "_M1aParAll", # object name
332.                  ".RData",sep=""))
333.
334. save.image(file=paste(dir.in,
335.                       if(DAC>0){DAC},
336.                       ScriptNameCode,
337.                       "_end_",
338.                       "AllObj_Key0nes", # object name
339.                       ".RData",sep=""))
340.
341. #####
342. ### next script ###
343. #####
3437. #####
3438. #####
3439. Obj2Save <- objects(pattern="M3C")
3440. Obj2Save # These are all the M3C objects created in this module
3441. #####
3442. save(list=Obj2Save,
3443.      file=paste(dir.rdata,
3444.                  "Obj_", # These are all the objects created in this module
3445.                  if(DAC>0){DAC},
3446.                  ScriptNameCode,"_",
3447.                  "End_All",".RData",sep=""))
3448.
3449. rm(TmpCopyText,Obj2Save)
3450. save.image(paste(dir.rdata,
3451.                  if(DAC>0){DAC},
3452.                  ScriptNameCode,
3453.                  "_awI_End_All_Alphas_RV",
3454.                  ".RData",sep=""))
```

27

PAPER 3. Sampling bias in presence-only data used for species distribution modelling: Assessment and effects on models.

Bente Støa, Rune Halvorsen, Sabrina Mazzoni, and Vladimir I. Gusarov.
Sommerfeltia. (In press)

PAPER 4. **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?**
Mika Bendiksby, Sabrina Mazzoni, Marte Holten Jørgensen, Rune Halvorsen and Håkon Holien. *Journal of Biogeography* (2014) 41, 2020–2031.

PAPER 5. **How important are choice of model selection method and spatial autocorrelation of presence data for distribution modelling by MaxEnt?** Rune Halvorsen, Sabrina Mazzoni, John W. Dirksen, Erik Næsset, Terje Gobakken and Mikael Ohlson.

Ecological Modelling (2016), **328**, 108-118.

DOI: [dx.doi.org/10.1016/j.ecolmodel.2016.02.021](https://doi.org/10.1016/j.ecolmodel.2016.02.021)

PAPER 6. Optimal MaxEnt model selection strategy evaluated by use of independent test data.

Sabrina Mazzoni, Rune Halvorsen, Vegar Bakkestuen, Inger Auestad, Trine Bekkby, Johannes Breidenbach, Desalegn Chala, John W. Dirksen, Anette Edvardsen, Dag Endresen, Lars Erikstad, Hege Gundersen, Einar Heegaard, Knut Anders Hovstad, Eli Rinde and Anders K. Wollan.
(Manuscript submitted to Global Ecology and Biogeography)