Combining multiple data sources in species distribution models while accounting for spatial dependence and overfitting with combined penalised likelihood maximisation

Ian W. Renner[1*], Julie Louvrier[2], and Olivier Gimenez[3]

[1] School of Mathematical and Physical Sciences, The University of Newcastle, Australia

[2] Department of Ecological Dynamics, Leibniz Institute for Zoo and Wildlife Research, Department of Evolutionary Ecology, Alfred-Kowalke-Str. 17, D-10315 Berlin, Germany

[3] CEFE, CNRS, Univ Montpellier, Univ Paul Valéry Montpellier 3, EPHE, IRD, Montpellier, France

Word count: 6,785

*ian.renner@newcastle.edu.au

Summary

1. The increase in availability of species data sets means that approaches to species distribution modelling that incorporate multiple data sets are in greater demand. Recent methodological developments in this area have led to combined likelihood approaches, in which a log-likelihood comprised of the sum of the log-likelihood components of each data source is maximised. Often, these approaches make use of at least one presence-only data set and use the log-likelihood of an inhomogeneous Poisson point process model in the combined likelihood construction. While these advancements have been shown to improve predictive performance, they do not currently address challenges in presence-only modelling such as checking and correcting for violations of the independence assumption of a Poisson point process model or more general challenges in species distribution modelling such as overfitting.
2. In this paper, we present an extension of the combined likelihood framework which accommodates alternative presence-only likelihoods in the presence of spatial dependence as well as lasso-type penalties to account for potential overfitting. We compare the proposed combined penalised likelihood approach to the standard combined likelihood approach via simulation and apply the method to modelling the distribution of the Eurasian lynx in the Jura Mountains in eastern France.
3. The simulations show that the proposed combined penalised likelihood approach has better predictive performance than the standard approach when spatial dependence is present in the data. The lynx analysis shows that the predicted maps vary significantly between the model fitted with the proposed combined penalised approach accounting for spatial dependence and the model fitted with the standard combined likelihood.
4. This work highlights the benefits of careful consideration of the presence-only components of the combined likelihood formulation, and allows greater flexibility and ability to accommodate real datasets.

Keywords: area-interaction models; diagnostic tools; lasso; occupancy models; point process models; presence-only data

1 Introduction

Species distribution models (SDMs), in which the distributions of species are modelled as a function of environmental predictors, rely on information about where a species has been observed (Guisan *et al.*, 2017). Different SDM methods have been developed over the past few decades to accommodate the different protocols by which this species information is collected. For example, logistic regression and its extensions are often used when species detections and non-detections are recorded at a set of systematically designed locations (known as "presence-absence" data), while point process models (PPMs, see Renner *et al.* (2015) for an overview) have emerged as a unifying framework for fitting SDMs informed by "presence-only" data, in which only information about species presence locations are available. Statistically, these methods are often fitted by maximising a corresponding likelihood expression, and the parameter estimates which maximise the likelihood may be used to produce maps of relative habitat suitability, reported as a habitat suitability index (Hirzel *et al.*, 2002), probability of species presence (Phillips *et al.*, 2006), or intensity of locations per unit area (Warton & Shepherd, 2010) depending on the method.

Increasingly, species data are available from multiple sources and types. Many papers have advocated for fitting models to a combination of the available data types, illustrating benefits in model performance (Miller *et al.*, 2019). Dorazio (2014) illustrated via simulations that adding a small amount of systematically-collected presence-absence data to available presence-only data significantly improves predictive performance. Fithian *et al.* (2015) showed that fitting a combined presence-only and presence-absence model to multiple species leverages the information of more abundant species to improve predictive performance for less prevalent species and allows sampling bias inherent in presence-only data to be estimated and corrected. These models are fitted by maximising a combined log-likelihood expression which is the sum of the log-likelihoods of the presence-only and presence-absence components:

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{s}_{\mathrm{PO}}, \mathbf{y}_{\mathrm{PA}}) = \ell_{\mathrm{PO}}(\boldsymbol{\alpha}_{\mathrm{PO}}, \boldsymbol{\beta}; \mathbf{s}_{\mathrm{PO}}) + \ell_{\mathrm{PA}}(\boldsymbol{\alpha}_{\mathrm{PA}}, \boldsymbol{\beta}; \mathbf{y}_{\mathrm{PA}}).$$

Here, $\mathbf{s}_{\mathrm{PO}}$ contains the locations of a presence-only data source, while $\mathbf{y}_{\mathrm{PA}}$ contains a vector of presence-absence detections and non-detections at a set of pre-selected sites. parameters associated with the observation process unique to the presence-only and presence-absence data sets are denoted by $\boldsymbol{\alpha}_{\mathrm{PO}}$ and $\boldsymbol{\alpha}_{\mathrm{PA}}$, respectively, and collectively contained in the vector $\boldsymbol{\alpha}$. Hereafter, we refer to these parameters as sampling bias parameters, as they

may bias the intensity estimates as a result of the process of sampling the data. The key advancement of the combined likelihood approach is that the environmental response, parameterised by $\boldsymbol{\beta}$, is informed by both the presence-only and presence-absence data.

Such an approach implicitly assumes that the data sets are statistically independent, which allows for the combined log-likelihood to be expressed as a sum of the single-source log-likelihoods.

Other combinations may be done in similar fashion. For example, Koshkina *et al.* (2017) considered a combination of presence-only data with site-occupancy data, and Pacifici *et al.* (2017) developed a multivariate conditional autoregressive model to account for spatial autocorrelation in occurrence and detection error.

While these papers clearly advance the practice of fitting SDMs in important ways, they do not address some common challenges that arise in real datasets. For example, they all consider an inhomogeneous Poisson point process model (IPPPM) for the presence-only data in the combination. In many real data sets, however, the implicit assumption that the point locations are independently distributed conditional on the environment is not met. Residual clustering or repulsion of the point locations not accounted for with an IPPPM due to the observation process, unconsidered environmental covariates, or biological factors would hence render the IPPPM inappropriate. One option to account for spatial dependence is to consider a log-Gaussian Cox Process, as Gelfand & Shirota (2018) do for a combination of presence-only and presence-absence data. Furthermore, none of the current literature in combined likelihood approaches includes ways to account for possible overfitting that results from including too many covariates in the model.

However, advances in SDM literature provide solutions to these common problems. Diagnostic tools such as the inhomogeneous K function (Baddeley & Turner, 2000) and its simulation envelope (Diggle, 2003) can be used to determine departures from the independence assumption, and a wide number of alternative PPMs which account for spatial dependence may be included in the likelihood combination instead. Furthermore, penalised regression techniques such as the lasso penalty (Tibshirani, 1996) and its extension the adaptive lasso (Zou, 2006) may be used as a way to perform variable selection. Lasso regularisation has been shown to boost predictive performance of SDMs and has been applied to IPPPMs (Renner & Warton, 2013) and occupancy models (Hutchinson *et al.*, 2015).

In this paper, we present a penalised combined likelihood model in a way that it is more

suitable for real data sets. In particular, we accommodate alternative forms of presence-only models to account for spatial dependence and affix a penalty on model complexity to address overfitting. In Section 2, we present the penalised combined likelihood formulation. In Section 3, we illustrate via simulations the improvements that this formulation provides and apply the proposed formulation to analyse the distribution of the Eurasian lynx (*lynx lynx*) in the Jura Mountains in eastern France. Finally, we present a discussion and further avenues for research in this area in Section 4.

2 Materials and Methods

2.1 Combined Penalised Likelihood Formulation

We define the weighted, combined penalised log-likelihood as follows

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{y}) = \sum_{i=1}^{D} \ell_i(\boldsymbol{\alpha}_i, \boldsymbol{\beta}; \mathbf{y}_i) - p(\boldsymbol{\alpha}, \boldsymbol{\beta}). \qquad \text{(eqn 1)}$$

Here, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_D)^\top$ is a q-dimensional vector that collects coefficients for the variables $\mathbf{Z}$ used to model sampling bias for each of the D components individually. The environmental response is measured by a set of variables $\mathbf{X}$ and is parametrised by $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$, which is collectively informed by all D components. The species data for all D components is collected in a set $\mathbf{y}$, with each individual data source $\mathbf{y}_i$ determining the form of the component likelihood $\ell_i(\boldsymbol{\alpha}_i, \boldsymbol{\beta}; \mathbf{y}_i)$. Finally, $p(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is a penalty term described in further detail below.

While many possibilities for the likelihood terms $\ell_i(\boldsymbol{\alpha}_i, \boldsymbol{\beta}; \mathbf{y}_i)$ are possible, we will focus on likelihood expressions for a PPM and for an occupancy model. For an IPPPM, we typically model the intensity of points $\mu(s)$ over a given study region $\mathcal{A}$ as a log-linear function of environmental variables $\mathbf{X}$ and sampling bias terms $\mathbf{Z}$ and derive estimates $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\alpha}}_{\text{PO}}$ of the associated parameters by maximising a log-likelihood expression given by (Cressie, 1992):

$$\ell_{\text{PO}}(\boldsymbol{\alpha}_{\text{PO}}, \boldsymbol{\beta}; \mathbf{s}_{\text{PO}}) = \sum_{s \in \mathbf{s}_{\text{PO}}} \ln \mu(s) - \int_{s \in \mathcal{A}} \mu(s) ds. \qquad \text{(eqn 2)}$$

In the simple occupancy model we consider, each site i is visited J_i times. We collect the history of detections and non-detections for all N sites in a matrix $\mathbf{y}_{\text{occ}}$. Assuming that the probability that site i is occupied is given by ψ_i and that the occupancy of the sites remains constant throughout the history of visits. We further assume the probability of

detecting the species if present is p_i. Under these assumptions, we can then model the probability of observing y_i detections at site i as

$$P(Y_i = y_i) = \underbrace{\psi_i \binom{J_i}{y_i} p_i^{y_i} (1-p_i)^{J_i - y_i}}_{\text{species present}} + \underbrace{I(y_i = 0)(1-\psi_i)}_{\text{species absent}},$$

where $I(\cdot)$ is the indicator function.

We can relate the occupancy ψ_i of site i to an inhomogeneous Poisson intensity μ_i of the species distribution over site i as in Koshkina *et al.* (2017):

$$\psi_i = 1 - e^{-\mu_i \times A_i},$$

where A_i is the area of site i. Note that $\mu_i \times A_i$ is an approximation of $\int_{s \in \text{site } i} \mu(s) ds$ that is reasonable if μ_i reasonably approximates the average intensity within site i.

As with the IPPPM, we can then model intensity as a log-linear function of environmental variables $\mathbf{X}$ and model detection probability p_i as a function of some detection covariates $\mathbf{Z}$, such as the logit or complementary log-log function. We can then compute estimates $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\alpha}}_{\text{occ}}$ of the associated model parameters by maximising the log-likelihood expression given by:

$$\ell_{\text{occ}}(\boldsymbol{\alpha}_{\text{occ}}, \boldsymbol{\beta}; \mathbf{y}_{\text{occ}}) = \ln \prod_{i=1}^{N} P(Y_i = y_i)$$

The term $p(\boldsymbol{\alpha}, \boldsymbol{\beta})$ in eqn 1 is a penalty on model complexity applied to both the environmental parameters $\boldsymbol{\beta}$ and the sampling bias parameters $\boldsymbol{\alpha}$ to shrink these parameters toward zero in order to boost predictive performance. Here, we consider both the traditional lasso penalty (Tibshirani, 1996) and the adaptive lasso penalty (Zou, 2006). For the traditional lasso penalty,

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \lambda \left(\sum_{j=1}^{p} |\beta_j| + \sum_{k=1}^{q} |\alpha_k| \right),$$

where λ is the size of the tuning parameter. For the adaptive lasso penalty,

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma) = \lambda \left(\sum_{j=1}^{p} w_j |\beta_j| + w_{p+k} \sum_{k=1}^{q} |\alpha_k| \right),$$

where $\mathbf{w} = (w_1, \ldots, w_{p+q})^\top$ are weights for the adaptive lasso, typically of the form:

$$w_i = \begin{cases} \left|\hat{\beta}_i^{(\text{unp})}\right|^{-\gamma} & 1 \leq i \leq p \\ \left|\hat{\alpha}_{i-p}^{(\text{unp})}\right|^{-\gamma} & p+1 \leq i \leq p+q, \end{cases}$$

for $\gamma > 0$. Here, $\hat{\beta}_i^{(\text{unp})}$ is the unpenalised coefficient estimate corresponding to the ith environmental variable $\mathbf{x}_i$ and $\hat{\alpha}_i^{(\text{unp})}$ is the unpenalised coefficient estimate corresponding to the ith sampling bias variable $\mathbf{z}_i$. The shape of the weights is determined by the parameter γ. The data-driven choice of the adaptive weights $\mathbf{w}$ ensures that more important covariates (*i.e.* those with coefficient estimates further away from 0) will be penalised less. This construction also enables the adaptive lasso to achieve so-called oracle properties (Zou, 2006), which means that asymptotically, the correct subset of coefficients will be chosen and the procedure has an optimal estimation rate.

We can use eqn 1 to represent the simpler framework introduced by Dorazio (2014) and Fithian *et al.* (2015) by setting $p(\boldsymbol{\alpha}, \boldsymbol{\beta}) = 0$. We further extend this framework by considering alternative choices for those component likelihoods $\ell_i(\boldsymbol{\alpha}_i, \boldsymbol{\beta}; \mathbf{y}_i)$ informed by presence-only data. Rather than consider only inhomogeneous Poisson point process models, we consider area-interaction models (Widom & Rowlinson, 1970; Baddeley & van Lieshout, 1995) when diagnostic analysis of these data sources identifies spatial dependence among the presence-only locations. Area-interaction models account for spatial dependence through a vector of computed point interactions $\mathbf{t}_{\mathbf{s}}$, which measure the proportion of overlap among circles of a nominal radius around the observed points $\mathbf{s}$. They can account for both clustering and repulsion of points – the model parameter η characterises the nature of the spatial dependence, with values of η less than 1 signalling point repulsion and values of η greater than 1 signalling point clustering.

Because the likelihood expression of an area-interaction model is intractable, it is typically fitted via maximum pseudolikelihood (Besag, 1977):

$$\ell_{\text{AI}}(\boldsymbol{\alpha}_{\text{PO}}, \boldsymbol{\beta}, \eta; \mathbf{s}_{\text{PO}}) = \sum_{s \in \mathbf{s}_{\text{PO}}} \ln \mu(s; \mathbf{s}_{\text{PO}}) - \int_{s \in \mathcal{A}} \mu(s; \mathbf{s}_{\text{PO}}) ds.$$

This log-pseudolikelihood expression appears the same as eqn 2, with the exception that the intensity $\mu(s)$ is replaced by conditional intensity $\mu(s; \mathbf{s}_{\text{PO}})$ (Papangelou, 1974), reflecting the fact that for the area-interaction model, intensity at a location s is conditional on the other points in the pattern $\mathbf{s}_{\text{PO}}$.

2.2 Implementation in R

To fit models with the combined penalised log-likelihood in eqn 1, we have developed a set of functions in `R` inspired by the `optim` function and `ppmlasso` package (Renner & Warton, 2013). The main function `comb_lasso` takes as an input a list of species data, associated environmental data, and formulae for the environmental trend and sampling bias trends for each component, along with details such as type of presence-only likelihoods to use, the type of penalty, the number of models to fit, and the tuning parameter criterion. The function applies the coordinate descent algorithm of Osborne *et al.* (2000). This requires the derivatives of the component likelihoods (also known as "score equations") to be computed. Analytical score equations are supplied directly to the `optim` function, which serves as the machinery of the optimisation. A tutorial illustrating use of this code for the simulations as performed in Section 3.1 as well as some functions written to plot intensity maps and features of the lasso penalisation is provided in the supplementary material.

3 Results

3.1 Simulations

To investigate the benefits of the proposed penalised combined likelihood formulation, we used the `rpoispp` function in `spatstat` (Baddeley & Turner, 2005) to generate a large inhomogeneous Poisson pattern $\mathbf{s}_{\text{true}}$ of roughly 10,000 points on a 30×30-unit square window from an intensity pattern defined by linear and quadratic terms of two generated variables (hence four meaningful covariates $\mathbf{x}_1, \ldots, \mathbf{x}_4$ parameterised by coefficients $\beta_1, \ldots, \beta_4$).

From this pattern, we generated two presence-only subsamples $\mathbf{s}_1$ and $\mathbf{s}_2$ biased by a different observation process. The first presence-only subsample $\mathbf{s}_1$ was biased by $\mathbf{z}_1$, the distance to a simulated road network, and the other $\mathbf{s}_2$ by $\mathbf{z}_2$, the distance to a simulated categorical covariate. We varied the size of the subsamples such that each pattern had 25, 100, or 400 points. We also varied the strength of the clustering of the presence-only subsamples by setting the coefficient of the interaction term $\nu_i = \ln \eta_i$ for $i = 1, 2$. Here, the patterns either exhibit no clustering ($\nu_i = 0$), moderate clustering ($\nu_i = 0.5$) or strong clustering ($\nu_i = 1$). In each case, the radius of interactions is set to 1 spatial unit. To sample the points in $\mathbf{s}_1$, we proceed as follows:

1. Initialise the set of sampled points $\mathbf{s}_1 = \emptyset$ and the point interactions $\mathbf{t}_{\mathbf{s}_1}$ to be a vector of 0s

2. Compute the biased conditional intensity $\mu_1(s; \mathbf{s}_1)$ at every point in $\mathbf{s}_{\text{true}}$ using $\mathbf{x}_1, \ldots, \mathbf{x}_4$, the sampling bias covariate $\mathbf{z}_1$, and the current vector of point interactions $\mathbf{t}_{\mathbf{s}_1}$, where the biased conditional intensity is defined as follows:

$$\ln \mu_1(s; \mathbf{s}_1) = \beta_1 \mathbf{x}_1(s) + \beta_2 \mathbf{x}_2(s) + \beta_3 \mathbf{x}_3(s) + \beta_4 \mathbf{x}_4(s) + \alpha_1 \mathbf{z}_1(s) + \nu_1 \mathbf{t}_{\mathbf{s}_1}(s)$$

3. Set $\mu_1(s; \mathbf{s}_1) = 0$ for all $s \in \mathbf{s}_1$. That is, we set the conditional intensity for any point already selected in $\mathbf{s}_1$ to 0 to ensure these points are not resampled

4. Randomly select a point from $\mathbf{s}_{\text{true}}$ with sampling probabilities proportional to the conditional intensities and add the selected point to $\mathbf{s}_1$

5. Update the vector of point interactions $\mathbf{t}_{\mathbf{s}_1}$ for all points in $\mathbf{s}_{\text{true}}$ using the internal `evalInteraction` function in `spatstat`, which computes point interactions based on a supplied point pattern for a given set of locations and interaction radius

6. Repeat steps 2-5 until we have sampled the desired number of points

We sample $\mathbf{s}_2$ in a similar manner, using $\mathbf{z}_2$ instead of $\mathbf{z}_1$ to create the sampling bias and computing point interactions $\mathbf{t}_{\mathbf{s}_2}$.

Because the true pattern $\mathbf{s}_{\text{true}}$ is Poisson, this simulation setup emulates a scenario in which the clustering of the observed point patterns is an artefact of the observation process – this can happen if, for example, records are publicly available and enthusiasts for the species report further observations near the publicly available locations (Johnston *et al.*, 2019).

We also generated a history $\mathbf{y}_{\text{occ}}$ of detections and non-detections from 5 visits to each of 100 sites centred along a regular grid in the 30×30-unit observation window to emulate a data set for which we could consider occupancy modelling. The species was considered present at a site if the closest point in the pattern $\mathbf{s}_{\text{true}}$ was within a distance of 0.18 units of the centre of the site, such that the area of each site is roughly 0.1 square units. The history of detections and non-detections at each site where the species was considered present was randomly generated according to detection probabilities defined by the inverse of the cloglog function evaluated at a generated detection covariate $\mathbf{z}_3$.

Finally, we generated four dummy covariates $\mathbf{d}_1, \ldots, \mathbf{d}_4$ to include in fitted models that were meaningless in describing the true species distribution. We did this to reflect the fact that in real applications, we may not know which among a suite of candidate variables truly determine the species distribution. We ensured that the maximum absolute correlation among all pairs of variables was smaller than 0.5.

After generating the species data, we fit a number of models, using as input environmental covariates the four meaningful covariates $\mathbf{x}_1, \ldots, \mathbf{x}_4$ (parameterised by $\beta_1, \ldots, \beta_4$) as well as four dummy covariates $\mathbf{d}_1, \ldots, \mathbf{d}_4$ (parameterised by $\beta_5, \ldots, \beta_8$) and using as sampling bias covariates $\mathbf{z}_1$, $\mathbf{z}_2$, and $\mathbf{z}_3$ (parameterised by α_1, α_2, and α_3). For both Poisson and area-interaction presence-only likelihoods, we fit a model without any penalty, with a lasso penalty, and with an adaptive lasso penalty. For the models fitted with either a lasso or an adaptive lasso penalty, we fit regularisation paths of 1000 models, increasing the penalty from 0 to the smallest penalty $\lambda_{\max}$ that would shrink all coefficients to 0, thus covering the entire scope of possible model sizes. The model which minimised BIC was chosen among the 1000 fitted models. We considered as species data a combination of all three of $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$. This led to a total of six models being fitted, summarised in Table 1.

Model	Species Data	Presence-only likelihood	Penalty
1	$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$	IPPPM	None
2	$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$	IPPPM	Lasso
3	$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$	IPPPM	Adaptive Lasso
4	$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$	Area-interaction	None
5	$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$	Area-interaction	Lasso
6	$\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{y}_{\text{occ}}$	Area-interaction	Adaptive Lasso

Table 1: Models fitted in each simulation using the proposed combined penalised likelihood. The models also varied based on the likelihood expression for any presence-only components and the type of penalty used, if any.

To evaluate performance, we compared the integrated mean squared error of the true intensity surface with rescaled fitted intensity surfaces of the six models. The fitted intensity surfaces were rescaled to have the same mean intensity as the true intensity surface to ensure that fair comparisons are made as models using different species data sources will have varying intercepts to reflect the estimated abundance of the points.

We performed 1,000 simulations of the data sets for each of the nine combinations of

presence-only data set size and clustering strength and the resultant model fits on 512GB nodes powered by 3.0 GHz Intel Xeon Gold (E5-6154) processor from the University of Newcastle's High Performance Computing cluster. The 9,000 simulation tasks took approximately 7,000 hours.

Figure 1 shows boxplots of the calculated integrated mean squared errors from the simulations. From these results, we can draw the following conclusions. First, the models fitted with the area-interaction presence-only likelihoods have performance benefits over the models fitted with Poisson presence-only likelihoods. This is true even when there is no clustering (first column), a setting for which the Poisson likelihood is appropriate. Comparing the plots across rows and down columns, we see that the performance advantage tends to increase as the degree of clustering gets larger and as the sample size increases, respectively.

In the Appendix, we show that the parameter coefficients $\beta_1, \ldots, \beta_4$ corresponding to the meaningful covariates $\mathbf{x}_1, \ldots, \mathbf{x}_4$ are increasingly biased away from 0 for the models fitted with Poisson presence-only likelihoods, both as sample size increases and as the strength of presence-only clustering increases. The inclusion of the area-interaction term takes an increasing amount of signal from the environmental covariates as the strength of the presence-only clustering increases. For low sample sizes, there is a suggestion that this signal dampening may be too strong, though such an overcorrection disappears as sample size increases.

Second, penalisation via the lasso or adaptive lasso improves model performance when there is no presence-only clustering, and this improvement is greatest for smaller sample sizes. This is an expected conclusion given the danger of overfitting is greater with fewer observations. Models penalised with the adaptive lasso tend to outperform models penalised with the lasso when there is no presence-only clustering. However, lasso penalisation does not notably improve performance when there is presence-only clustering. In fact, there is a suggestion that applying a lasso penalty may slightly hinder performance when applying an area-interaction presence-only likelihood for small sample sizes. Although the benefits of penalisation are negligible with large data sets, fitting models with a penalty does not hurt the performance.

In summary, it appears that the proposed combined penalised likelihood framework provides the best performance. A more detailed discussion of the simulation results, including boxplots of the fitted coefficients, appears in the Appendix.

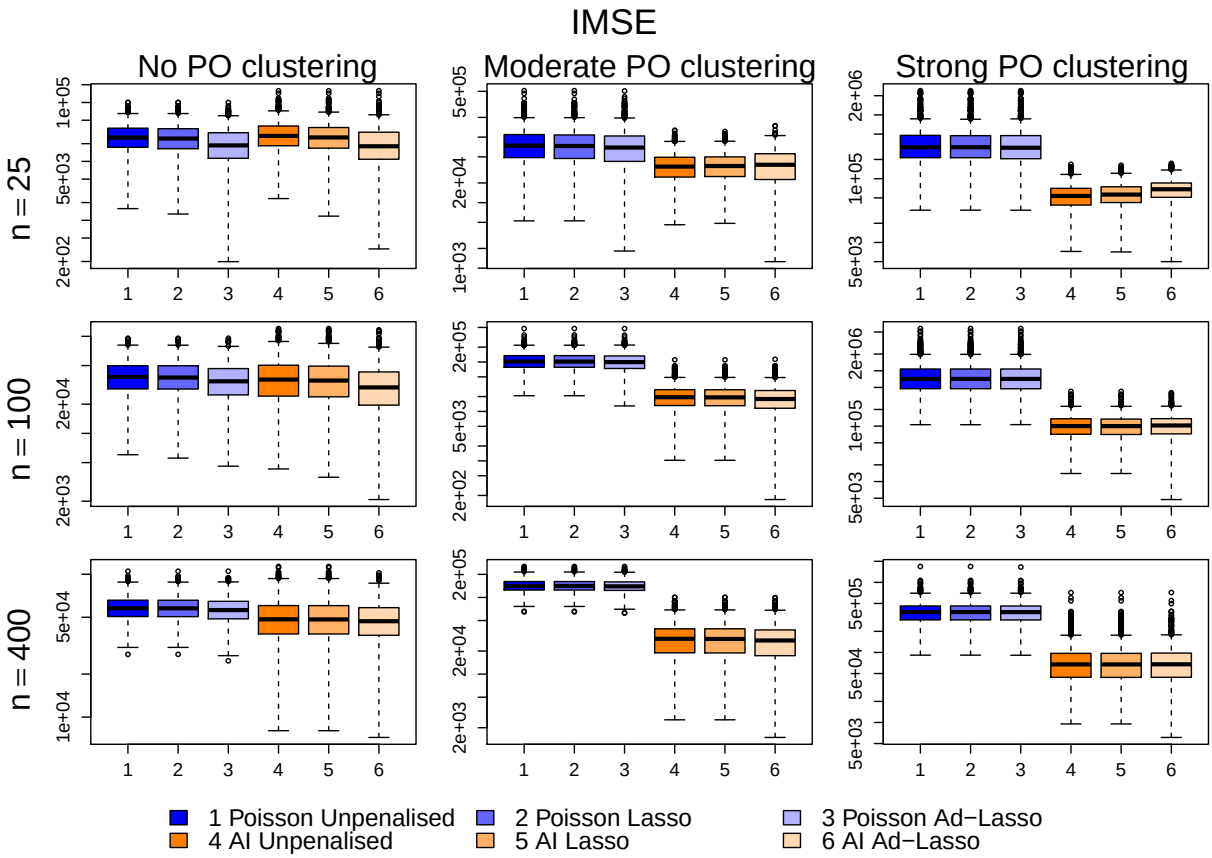


Figure 1: Boxpots of integrated mean squared error for the six models described in Table 1 for different combinations of presence-only sample size and clustering strength.

3.2 Analysis of Eurasian lynx distribution in the Jura Mountains

We now demonstrate the use of the combined penalised likelihood approach to analyse the distribution of the Eurasian lynx in the Jura Mountains in eastern France.

Lynx went extinct in France at the end of the 19th century due to habitat degradation, human persecution and decrease in prey availability (Vandel & Stahl, 2005). The species was reintroduced in Switzerland in the 1970s (Breitenmoser *et al.*, 1998), then re-colonised France through the Jura mountains in the 1980s (Vandel & Stahl, 2005). The species is listed as endangered under the 2017 IUCN Red list and is of conservation concern in France due to habitat fragmentation, poaching and collisions with vehicles. The Jura holds the bulk of the French lynx population.

We have three sources of lynx data in the Jura Mountains: a presence-only data set consisting of 440 opportunistic sightings in the wild from 2009-2011 (denoted $\mathbf{s}_{\mathrm{w}}$), another presence-only data set consisting of 240 reported interferences of lynx with domestic

livestock in 2009-2011 (denoted $\mathbf{s}_{\mathrm{d}}$), and pictures of lynx taken from cameras set up in 73 locations $\mathbf{s}_{\mathrm{c}}$ in the Jura Mountains in 2012. Lynx presence-only data were made of presence signs sampled all year long thanks to a network of professional and non-professional observers. Every observer is trained during a 3-day teaching course led by the French National Game and Wildlife Agency (ONCFS) to document signs of the species' presence (Duchamp *et al.*, 2012). Presence signs went through a standardised control process to prevent misidentification (Duchamp *et al.*, 2012). The camera data has daily reportings of the lynx across a total of 77 days. Due to this, we can consider the picture history of lynx at the camera locations in an occupancy modelling framework (Blanc *et al.*, 2014). In particular, we split the 77-day period into seven 11-day periods, such that the site history $\mathbf{y}_{\mathrm{c}}$ comprises seven detections and non-detections at each site in $\mathbf{s}_{\mathrm{c}}$ over each 11-day period.

Figure 2 shows the locations of the sightings in both presence-only data sets as well as the locations of the cameras. Both presence-only data sources appear to have different distributions, reflecting different sampling biases. There are more wild sightings in the northeast of the Jura Mountains, and more domestic interferences toward the southwest. Additionally, there appear to be some tight clusters within both data sets, with several records very close to each other.

To model the lynx distribution, we consider altitude, percentage of forest cover, distance to the nearest water source, and human population density as environmental variables. We model sampling bias in the wild records $\mathbf{s}_{\mathrm{w}}$ with distance to the nearest main road and distance to the nearest train line, and sampling bias in the domestic records $\mathbf{s}_{\mathrm{d}}$ with distance to the nearest farm and percentage of agricultural land. Finally, we model detection probability for the camera data with distance to the nearest urban area. We established this set of potential candidate environmental and detection variables based on previously studied species habitat preferences and detectability (Bouyer *et al.*, 2015). The Corine Land Cover land use repository from 2012 (Büttner *et al.*, 2014) supplies a map of land coverage including urban areas, water areas, forest areas, farm areas, and agricultural areas that was used to generate the percentage of forest areas and agricultural areas over 1 km $\times$ 1 km cells as well as distances to the nearest urban area, water source, and farm. Altitude was averaged over 1km $\times$ 1km cells from data available in the `raster` package in R, while human population density was averaged over 1 km $\times$ 1km cells taken from version 4 of the Gridded Population of the World data repository (Center for International Earth Science Information Network (CIESIN) – Columbia University, 2016). Distances

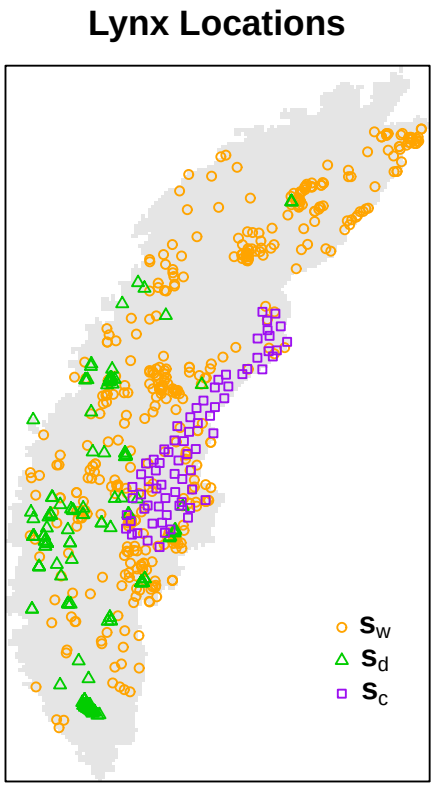


Figure 2: Locations of the lynx data in the Jura Mountains, including 440 observations in the wild $\mathbf{s}_w$, 220 reports of domestic interference $\mathbf{s}_d$, and 73 camera traps $\mathbf{s}_c$.

from the nearest main road and railway were computed from shapefiles from Version 151 of the ROUTE 500 database, accessible at http://professionnels.ign.fr/route500.

We fitted initial separate IPPPMs to the wild records $\mathbf{s}_w$ and the domestic records $\mathbf{s}_d$ using linear, quadratic, and interaction terms for the four environmental covariates, and linear terms for the sampling bias covariates. From these models, we are able to assess whether the assumption of independence inherent to the IPPPMs is appropriate with simluation envelopes of the inhomogeneous K-function in `spatstat`, as shown in Figure 3. Both of the envelopes for the IPPPMs fitted to the wild model (left panel) and the domestic model (middle panel) demonstrate additional clustering as the observed inhomogeneous K-function values plotted in red fall above the simulation envelopes for small radii. This suggests that fitting an IPPPM is inappropriate for these data sets. The right panel shows a simulation envelope of the cross K function as produced by the `Kcross.inhom` function of `spatstat`, which counts the expected number of wild sightings within a given distance of a domestic sighting, conditional on the spatially varying intensities of both patterns. We estimate the wild and domestic intensities from area-interaction models, and as the observed values of the cross K-function fall within the envelope boundaries, this suggests

that there is no clustering across the two data sets. This, in turn, suggests that the observed clustering within the wild and domestic data sets may be more likely attributable to the observation process than to some biological reality that induces clustering or a missed environmental covariate.

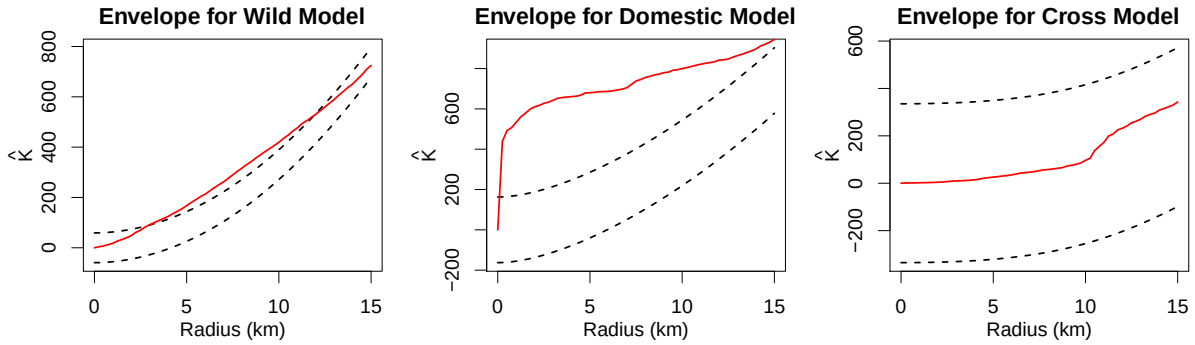


Figure 3: 95% simulation envelopes of the inhomogeneous K-function for the fitted IPPPM of the wild records (left), the fitted IPPPM of the domestic records (middle), and across the two fitted IPPPMs (right).

Consequently, we fit combined likelihood models using both the standard, unpenalised approach (analogous to Model 1 in Table 1) and the combined penalised likelihood formulation eqn 1 with a lasso penalty and area-interaction models for the presence-only data sources (analogous to Model 5 in Table 1). The radii chosen to capture the residual spatial patterning in the wild and domestic models are 2km and 5km, as chosen by the `profilepl` function in `spatstat`.

Figure 4 shows the bias-corrected fitted intensities from these two models. For the combined model which uses IPPPMs (left panel), the fitted intensity is corrected for the sampling bias terms modelled for the presence-only components using the method of Warton *et al.* (2013). For the combined penalised model which uses area-interaction models (right panel), the fitted intensity is corrected for these same sampling bias terms as well as the fitted point interactions – that is, we treat the interaction parameter ν as belonging to the set of sampling bias parameters $\boldsymbol{\alpha}$. The fitted models show strikingly different patterns, with the model which uses area-interaction components highlighting much more of the Jura Mountains as preferred habitat of lynx than the model which uses IPPPMs. The models suggest similar numbers of points throughout the Jura, but the distribution of these points are more heavily concentrated in the IPPPM model. This is because the area-interaction terms in the AI model lessen the impact of some clusters of points on the scale of the displayed bias-corrected intensities.

We do not have access to additional data with which to validate the performance of these models such as GPS data as in Gould *et al.* (2019), but the results of Section 3.1 suggest that the model which uses area-interaction components is likely to better reflect the true distribution of lynx.

The combined penalised model with the area-interaction components found the optimal lasso penalty was 0, resulting in a model which included all 18 covariates and both of the area-interaction terms. The fact that the optimal penalty is 0 suggests that the suite of covariates we chose to include, motivated by existing literature, seems to have been a good choice. In general, we recommend use of the lasso penalty as a safeguard against overfitting, particularly in contexts where the suite of candidate covariates for a species is less established as an insurance against overfitting.

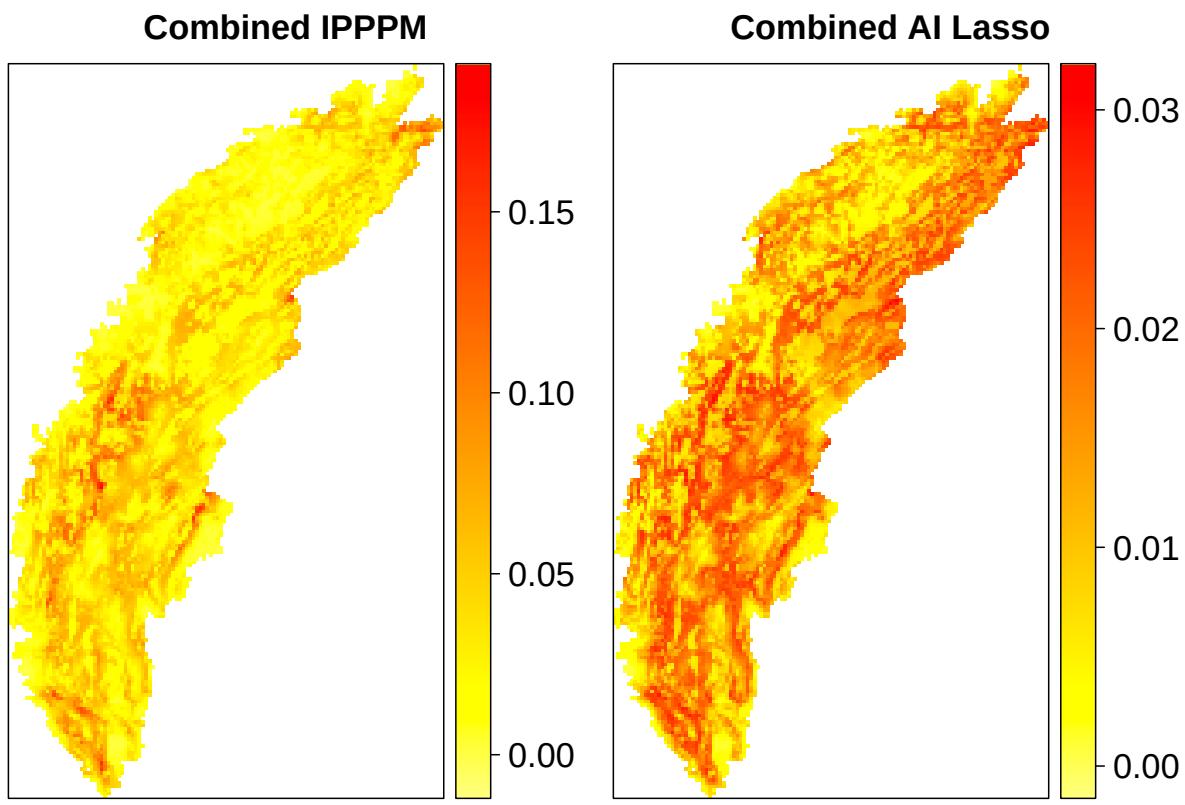


Figure 4: Fitted intensities using the combined likelihood formulation. Left: the model is fitted without any penalty and using inhomogeneous Poisson point process models for the presence-only data sources. Right: the model is fitted with a lasso penalty and using area-interaction models for the presence-only sources.

4 Discussion

The proposed combined penalised likelihood framework addresses some common problems that arise in real datasets. The flexibility to incorporate an area-interaction likelihood when there is spatial dependence in the presence-only data set and affix a penalty on model complexity enables improvements in predictive performance, as shown in Section 3.1.

4.1 Possible extensions

Despite these improvements, further advances are possible. Other penalty structures could be incorporated into the same framework. While the lasso and adaptive lasso showcased here show clear benefits in simulations, other penalised likelihood variants such as SCAD (Fan & Li, 2001) could lead to superior performance in some situations, and alternative methods to BIC of choosing the size of the penalty such as the Extended Bayesian Information Criterion ("ERIC", Hui *et al.*, 2015) could likewise be used.

While we make use of the area-interaction likelihood in this paper, there is a large family of Gibbs PPMs (Cressie, 1992) which accommodate different sorts of spatial dependence that could be used. Our choice of the area-interaction model as the alternative is motivated by the fact that it accommodates interactions of all orders instead of just pairwise interactions and that it can be used to model both clustering and repulsion of points.

The inclusion of the area-interaction terms dampens the signal of the environmental covariates. Although this makes sense when spatial dependence exists, we may dampen the signal too much. In the context of species distribution models, we might ask the question, "Does a given species record exist because its location is in particularly suitable habitat for the species, or because there are other records nearby?" If the answer to this question appears to be "both", as is often the case for presence-only data, we are at risk of "spatial confounding". In the single-source context, Hodges & Reich (2010) propose restricting the spatial effect to be orthogonal to the fixed covariate effects, while Simpson *et al.* (2017) and Sørbye *et al.* (2019) suggest careful selection of associated spatial priors to alleviate this risk. With our implementation, we could achieve something similar to the latter two papers by adjusting the magnitude of the lasso penalty for the area-interaction terms. In the Appendix, we highlight the tradeoff between the estimates of the interaction parameters $\hat{\nu}_i$ and both the estimates of the environmental parameters $\hat{\beta}_i$ and the sampling bias parameters $\hat{\alpha}_i$. However, a full exploration of the effects of spatial confounding remains

an open area of research and is beyond the scope of this paper.

In both the simulations in Section 3.1 and the lynx data analysis in Section 3.2, we made the rather limiting assumption of a closed population and that sites are either always occupied or always unoccupied. Nonetheless, occupancy models which take into account changing site dynamics could be used (MacKenzie *et al.*, 2003). Similarly, we have ignored the temporal aspect of the lynx distribution in this paper, but there is a wide suite of tools to fit spatio-temporal models in order to capture distribution dynamics for both the aforementioned occupancy modelling component as well as presence-only components (Cressie & Wikle, 2015).

Further improvements could be made by incorporating source weights in situations in which the data sources vary in quality. Indeed, presence-only data sources may be more prone to errors in coordinate locations as well as correct species identification, as they often include records by amateur enthusiasts. The combined penalised likelihood framework could easily be extended to include weights for the various data sources by adding a vector of source weights $\mathbf{w} = (w_1, \ldots, w_D)^\top$ to the formulation in eqn 1:

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{y}) = \sum_{i=1}^{D} w_i \ell_i(\boldsymbol{\alpha}_i, \boldsymbol{\beta}; \mathbf{y}_i) - p(\boldsymbol{\alpha}, \boldsymbol{\beta}). \qquad \text{(eqn 3)}$$

One possible strategy to incorporate such weights in eqn 3 could be to compare performance of single source models on independent data and upweight the contribution of data sources that are shown to have good performance.

Finally, while we incorporate sampling bias as a linear effect, non-linear effects can also be used as appropriate for a given sampling protocol, for example with distance sampling as discussed in Yuan *et al.* (2017).

4.2 Accounting for dependence within and among data sources

In the lynx data analysis in Section 3.2, we diagnosed spatial dependence within each of the presence-only data sources but found no spatial dependence across data sources. Tools such as the inhomogeneous K-envelope provide great insight into the underlying individual spatial processes that are observed. However, such diagnostic tools are not currently available for the combined likelihood models, and research in this area would be valuable as these models grow in popularity.

Another approach to constructing SDMs from multiple data sources could be to introduce a common latent spatial term $\xi(s)$, such as a Gaussian random field, which would account

for spatial dependence among points in all of the data sources as in Gelfand & Shirota (2018). The resulting likelihood expression would be:

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{y}) = \sum_{i=1}^{D} \ell_i(\boldsymbol{\alpha}_i, \boldsymbol{\beta}; \mathbf{y}_i) + \xi(\mathbf{y}) - p(\boldsymbol{\alpha}, \boldsymbol{\beta}), \qquad \text{(eqn 4)}$$

where $\xi(\mathbf{y}) \sim \text{MVN}(\mathbf{0}, \boldsymbol{\Sigma})$. Models of this type are typically fitted in a Bayesian framework. We could reduce the dimension of ξ through methods like fixed rank kriging or induce sparsity in $\boldsymbol{\Sigma}$ through lasso-type penalties such that the likelihood in eqn 4 could be fitted with software such as Template Model Builder (TMB, Kristensen *et al.*, 2016). Another way to achieve sparsity is with the stochastic partial differential equation approach (SPDE, Lindgren *et al.*, 2011), as implemented in the `inlabru` package (Bachl *et al.*, 2019).

4.3 Conclusion and Perspectives

The development of statistical methods is often motivated by new challenges raised by novel types of data sets. While the current literature on combined likelihood approaches represents a significant recent advancement, advances in other areas can be lost if not carried over with such methodological developments. This paper attempts to build a bridge between this exciting new arena for species distribution modelling and the rich suite of tools available for species distribution modelling, particularly that for presence-only data. Our hope is that other such bridges continue to be built in this spirit.

5 Acknowledgements

We thank the staff from the French National Game and Wildlife Agency, the Forest National Agency and the Departmental Federation of Hunters of Jura department, who collected the photographs during the camera-trapping session. We also thank all the volunteers that are members of the Réseau Loup-Lynx that collect every year precious presence signs of lynx all over its distribution area. We thank the University of Newcastle through the Early to Mid-Career Researcher Visiting Fellowship and the Université Paul-Valéry Montpellier through the Professeurs en mobilité universitaire fund for funding visits for I.R. which helped facilitate the collaborations that resulted in this work. O.G. was funded by CNRS and the “Mission pour l’Interdisciplinarité” through the “Osez l’Interdisciplinarité” initiative. Finally, we thank the reviewers for their very helpful

comments, which have helped us greatly improve the paper.

6 Authors' Contributions

I.R. and O.G. conceived the concept of the paper. I.R. developed the code to fit the models. J.L. sourced the species coordinates and covariates for the lynx analysis. I.R. and O.G. wrote the manuscript. I.R. and J.L. developed the tutorial in the supplementary information. All authors were involved in editing drafts of the manuscript.

References

Bachl, F.E., Lindgren, F., Borchers, D.L. & Illian, J.B. (2019) inlabru: an r package for bayesian spatial modelling from ecological survey data. *Methods in Ecology and Evolution*, **10**, 760–766.

Baddeley, A. & Turner, R. (2000) Practical maximum pseudolikelihood for spatial point patterns: (with discussion). *Australian & New Zealand Journal of Statistics*, **42**, 283–322.

Baddeley, A. & Turner, R. (2005) Spatstat: an R package for analyzing spatial point patterns. *Journal of Statistical Software*, **12**, 1–42.

Baddeley, A.J. & van Lieshout, M.N.M. (1995) Area-interaction point processes. *Annals of the Institute of Statistical Mathematics*, **47**, 601–619.

Besag, J. (1977) Some methods of statistical analysis for spatial data. *Bulletin of the International Statistical Institute*, **47**, 77–92.

Blanc, L., Marboutin, E., Gatti, S., Zimmermann, F. & Gimenez, O. (2014) Improving abundance estimation by combining capture–recapture and occupancy data: example with a large carnivore. *Journal of Applied Ecology*, **51**, 1733–1739.

Bouyer, Y., San Martin, G., Poncin, P., Beudels-Jamar, R.C., Odden, J. & Linnell, J.D. (2015) Eurasian lynx habitat selection in human-modified landscape in norway: Effects of different human habitat modifications and behavioral states. *Biological Conservation*, **191**, 291–299.

Breitenmoser, U., Breitenmoser-Würsten, C. & Capt, S. (1998) Re-introduction and present status of the lynx (lynx lynx) in switzerland. *Hystrix, the Italian Journal of Mammalogy*, **10**.

Büttner, G., Soukup, T. & Kosztra, B. (2014) Clc2012 addendum to clc2006 technical guidelines. *Final Draft, Copenhagen (EEA)*.

Center for International Earth Science Information Network (CIESIN) – Columbia University (2016) Gridded population of the world, version 4 (gpwv4): population density.

Cressie, N. (1992) *Statistics for spatial data*, volume 4. Wiley Online Library.

Cressie, N. & Wikle, C.K. (2015) *Statistics for spatio-temporal data*. John Wiley & Sons.

Diggle, P.J. (2003) *Statistical analysis of spatial point patterns*. Edward Arnold.

Dorazio, R.M. (2014) Accounting for imperfect detection and survey bias in statistical analysis of presence-only data. *Global Ecology and Biogeography*, **23**, 1472–1484.

Duchamp, C., Boyer, J., Briaudet, P.E., Leonard, Y., Moris, P., Bataille, A., Dahier, T., Delacour, G., Millisher, G., Miquel, C. *et al.* (2012) A dual frame survey to assess time- and space-related changes of the colonizing wolf population in france. *Hystrix-Italian Journal of Mammalogy*, **23**, 14–28.

Fan, J. & Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, **96**, 1348–1360.

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.

Gelfand, A. & Shirota, S. (2018) Preferential sampling for presence/absence data and for fusion of presence/absence data with presence-only data. *arXiv preprint arXiv:180901322*.

Gould, M.J., Gould, W.R., Cain III, J.W. & Roemer, G.W. (2019) Validating the performance of occupancy models for estimating habitat use and predicting the distribution of highly-mobile species: A case study using the american black bear. *Biological Conservation*, **234**, 28–36.

Guisan, A., Thuiller, W. & Zimmermann, N.E. (2017) *Habitat suitability and distribution models: with applications in R*. Cambridge University Press.

Hirzel, A.H., Hausser, J., Chessel, D. & Perrin, N. (2002) Ecological-niche factor analysis: how to compute habitat-suitability maps without absence data? *Ecology*, **83**, 2027–2036.

Hodges, J.S. & Reich, B.J. (2010) Adding spatially-correlated errors can mess up the fixed effect you love. *The American Statistician*, **64**, 325–334.

Hui, F.K., Warton, D.I. & Foster, S.D. (2015) Tuning parameter selection for the adaptive lasso using ERIC. *Journal of the American Statistical Association*, **110**, 262–269.

Hutchinson, R.A., Valente, J.J., Emerson, S.C., Betts, M.G. & Dietterich, T.G. (2015) Penalized likelihood methods improve parameter estimates in occupancy models. *Methods in Ecology and Evolution*, **6**, 949–959.

Johnston, A., Hochachka, W., Strimas-Mackey, M., Gutierrez, V.R., Robinson, O., Miller, E., Auer, T., Kelling, S. & Fink, D. (2019) Best practices for making reliable inferences from citizen science data: case study using ebird to estimate species distributions. *bioRxiv*, p. 574392.

Koshkina, V., Wang, Y., Gordon, A., Dorazio, R.M., White, M. & Stone, L. (2017) Integrated species distribution models: combining presence-background data and site-occupany data with imperfect detection. *Methods in Ecology and Evolution*, **8**, 420–430.

Kristensen, K., Nielsen, A., Berg, C.W., Skaug, H. & Bell, B.M. (2016) TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software*, **70**, 1–21.

Lindgren, F., Rue, H. & Lindström, J. (2011) An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **73**, 423–498.

MacKenzie, D.I., Nichols, J.D., Hines, J.E., Knutson, M.G. & Franklin, A.B. (2003) Estimating site occupancy, colonization, and local extinction when a species is detected imperfectly. *Ecology*, **84**, 2200–2207.

Miller, D.A., Pacifici, K., Sanderlin, J.S. & Reich, B.J. (2019) The recent past and promising future for data integration methods to estimate species' distributions. *Methods in Ecology and Evolution*, **10**, 22–37.

Osborne, M.R., Presnell, B. & Turlach, B.A. (2000) On the lasso and its dual. *Journal of Computational and Graphical Statistics*, **9**, 319–337.

Pacifici, K., Reich, B.J., Miller, D.A., Gardner, B., Stauffer, G., Singh, S., McKerrow, A. & Collazo, J.A. (2017) Integrating multiple data sources in species distribution modeling: A framework for data fusion. *Ecology*, **98**, 840–850.

Papangelou, F. (1974) The conditional intensity of general point processes and an application to line processes. *Probability Theory and Related Fields*, **28**, 207–226.

Phillips, S.J., Anderson, R.P. & Schapire, R.E. (2006) Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, **190**, 231–259.

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.

Renner, I.W., Elith, J., Baddeley, A., Fithian, W., Hastie, T., Phillips, S.J., Popovic, G. & Warton, D.I. (2015) Point process models for presence-only analysis. *Methods in Ecology and Evolution*, **6**, 366–379.

Simpson, D., Rue, H., Riebler, A., Martins, T.G., Sørbye, S.H. *et al.* (2017) Penalising model component complexity: A principled, practical approach to constructing priors. *Statistical Science*, **32**, 1–28.

Sørbye, S.H., Illian, J.B., Simpson, D.P., Burslem, D. & Rue, H. (2019) Careful prior specification avoids incautious inference for log-gaussian cox point processes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **68**, 543–564.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B (Methodological)*, pp. 267–288.

Vandel, J.M. & Stahl, P. (2005) Distribution trend of the eurasian lynx lynx lynx populations in france. *Mammalia mamm*, **69**, 145–158.

Warton, D.I., Renner, I.W. & Ramp, D. (2013) Model-based control of observer bias for the analysis of presence-only data in ecology. *PloS one*, **8**, e79168.

Warton, D.I. & Shepherd, L.C. (2010) Poisson point process models solve the "pseudo-absence problem" for presence-only data in ecology. *Annals of Applied Statistics*, **4**, 1383–1402.

Widom, B. & Rowlinson, J.S. (1970) New model for the study of liquid–vapor phase transitions. *The Journal of Chemical Physics*, **52**, 1670–1684.

Yuan, Y., Bachl, F.E., Lindgren, F., Borchers, D.L., Illian, J.B., Buckland, S.T., Rue, H., Gerrodette, T. *et al.* (2017) Point process models for spatio-temporal distance sampling data from a large-scale survey of blue whales. *The Annals of Applied Statistics*, **11**, 2270–2297.

Zou, H. (2006) The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, **101**, 1418–1429.

Appendix: Further results and analysis of simulations

In Section 3.1 of the main manuscript, we simulated two presence-only data sets and one occupancy data set to mimic the setting of the lynx data set described in Section 3.2. The simulations suggested that the combined models that incorporated an area-interaction presence-only likelihood had better predictive performance than models that incorporated a Poisson presence-only likelihood, and that this performance benefit increased with the degree of presence-only clustering and sample size. Here, we explore these results further by examining parameter estimation.

Figure A1 contains boxplots for the four coefficients $\hat{\beta}_1, \ldots, \hat{\beta}_4$ corresponding to the four meaningful environmental covariates $\mathbf{x}_1, \ldots, \mathbf{x}_4$ for each of the six models fitted to the data. The true values used to generate the species patterns $\mathbf{s}_1$ and $\mathbf{s}_2$ were $\beta_1 = 0.5$, $\beta_2 = -0.3$, $\beta_3 = 0.5$, and $\beta_4 = -0.3$. By looking down each column of plots, we see that the estimates become increasingly biased away from 0 for the models which incorporate a Poisson presence-only likelihood. This makes sense when there is presence-only clustering, as the environmental covariates absorb signal from the clustering as well as the environmental signal. Interestingly, this overinflation occurs even when there is no presence-only clustering, though not to the same degree.

The inclusion of the area-interaction terms increasingly mitigates this overinflation as the strength of presence-only clustering increases. The combined models with area-interaction presence-only likelihoods initially appear to overcorrect when $n = 25$ such that the estimates are biased toward 0, but then undercorrect when $n = 400$. This suggests that additional data is tending to amplify the environmental signal, and indicates potential spatial confounding, as mentioned in the discussion of the main manuscript. Nonetheless, the inclusion of the area-interaction terms leads to less biased parameter estimates overall, particularly as the strength of presence-only clustering and sample size increases. This coincides nicely with the simulation results presented in Figure 1 of the main manuscript.

Figure A2 contains boxplots of the coefficient estimates $\hat{\beta}_5, \ldots, \hat{\beta}_8$ corresponding to the dummy variables $\mathbf{d}_1, \ldots, \mathbf{d}_4$. This plot suggests that the lasso models and adaptive lasso models (the lighter colored boxplots within both the models with Poisson and area-interaction presence-only likelihoods) are more concentrated around the true value of 0, as expected. Once again, the medians of the models fitted with Poisson presence-only likelhoods are being dragged away from 0 as sample size increases, though the variances are smaller. The estimates are increasingly concentrated around 0 as the strength of

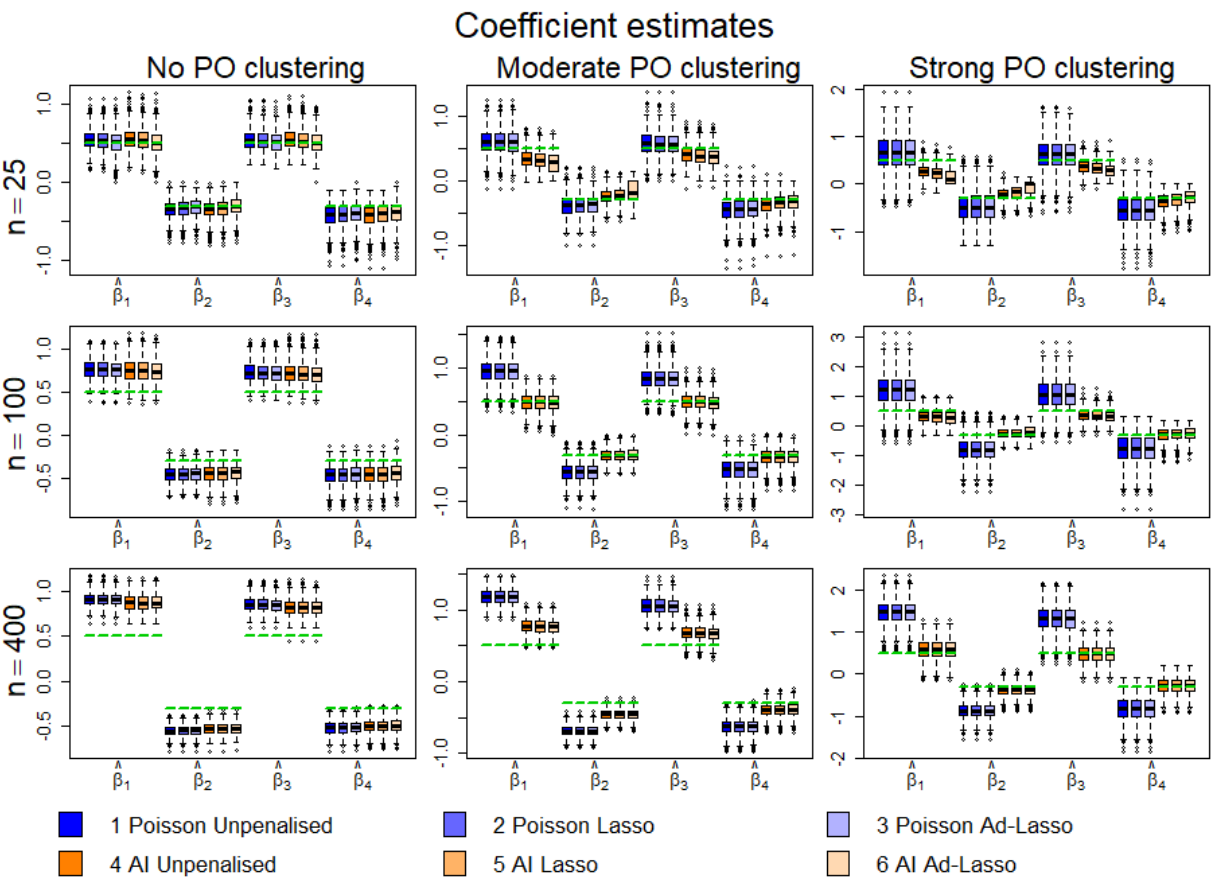


Figure A1: Boxpots of coefficient estimates corresponding to the four meaningful environmental covariates for the six models described in Table 1 of the manuscript. The green lines indicate the true values of the parameters used to generated the simulated data.

presence-only clustering increases for the models fitted with area-interaction presence-only likelihoods. This makes sense as the area-interaction terms dampen the signal of the other covariates.

Figure A3 contains boxplots of the coefficient estimates $\hat{\alpha}_1$ and $\hat{\alpha}_2$ corresponding to the two observer bias covariates $\mathbf{z}_1$ and $\mathbf{z}_2$ as well as the estimates $\hat{\nu}_1$ and $\hat{\nu}_2$ corresponding to the area-interaction terms. For the simulations, the true values of the observer bias covariates were $\alpha_1 = 1$ and $\alpha_2 = 1$. The true values of the area-interaction coefficients were $\nu = 0$ for the patterns with no presence-only clustering, $\nu = 0.5$ for the patterns with moderate presence-only clustering, and $\nu = 1$ for the patterns with strong presence-only clustering.

We see that the observer bias parameters α_1 and α_2 are better estimated with larger sample sizes, and also better estimated by the models that incorporate Poisson presence-only likelihoods, particularly as the strength of the presence-only clustering increases. This superior estimation is likely attributable to spatial confounding – the estimates in

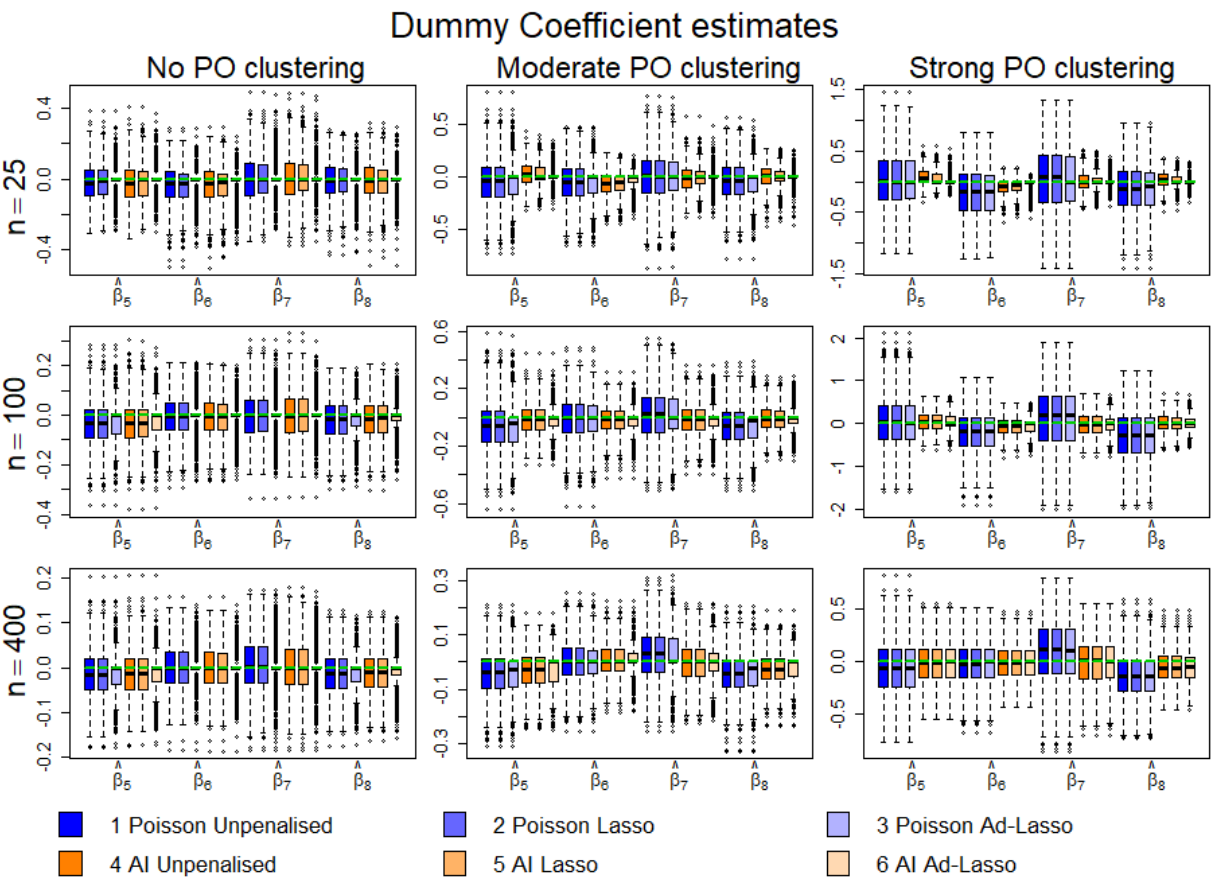


Figure A2: Boxpots of coefficient estimates corresponding to the four dummy environmental covariates for the six models described in Table 1 of the manuscript. The green lines indicate the true values of the parameters used to generated the simulated data.

the models fitted with Poisson presence-only likelihoods do not compete with estimates of the interaction terms for signal as they do in the case of the models fitted with area-interaction likelihoods.

The estimates of the interaction coefficients are naturally superior for the models fitted with area-interaction presence-only likelihoods. As sample size increases, the estimates of the interaction coefficients become larger, being too strong for $n = 100$ and $n = 400$. It seems there is a tradeoff between the estimation of the observer bias variables and the area interaction covariates. This makes sense – these variables will naturally be negatively correlated. In the case of the first presence-only data source, points are simulated to be both close to roads and close to each other. From a bias perspective, this means we can ask of a particular point, "was this point sampled because it was near to a road or because it is near to other points?" Similarly for the second presence-only data source, points are simulated to be both close to a particular level of a categorical covariate and close to each other. As Figure A4 indicates, these negative correlations grow stronger with increasing

sample size, making it difficult for the models to disentangle the effects of the observer bias covariates and the area-interaction terms even with the additional data.

As both the area-interaction terms and the observer bias variables are treated in the same way when making predictions, it is the overall contribution of both the observer bias and the area-interaction terms that defines the estimated bias surface corrected by the method of Warton *et al.* (2013). With the high negative correlations, overestimation of the area-interaction contribution may lead to underestimation of the contribution of the observer bias variables, but the overall bias surface may still be reasonable.

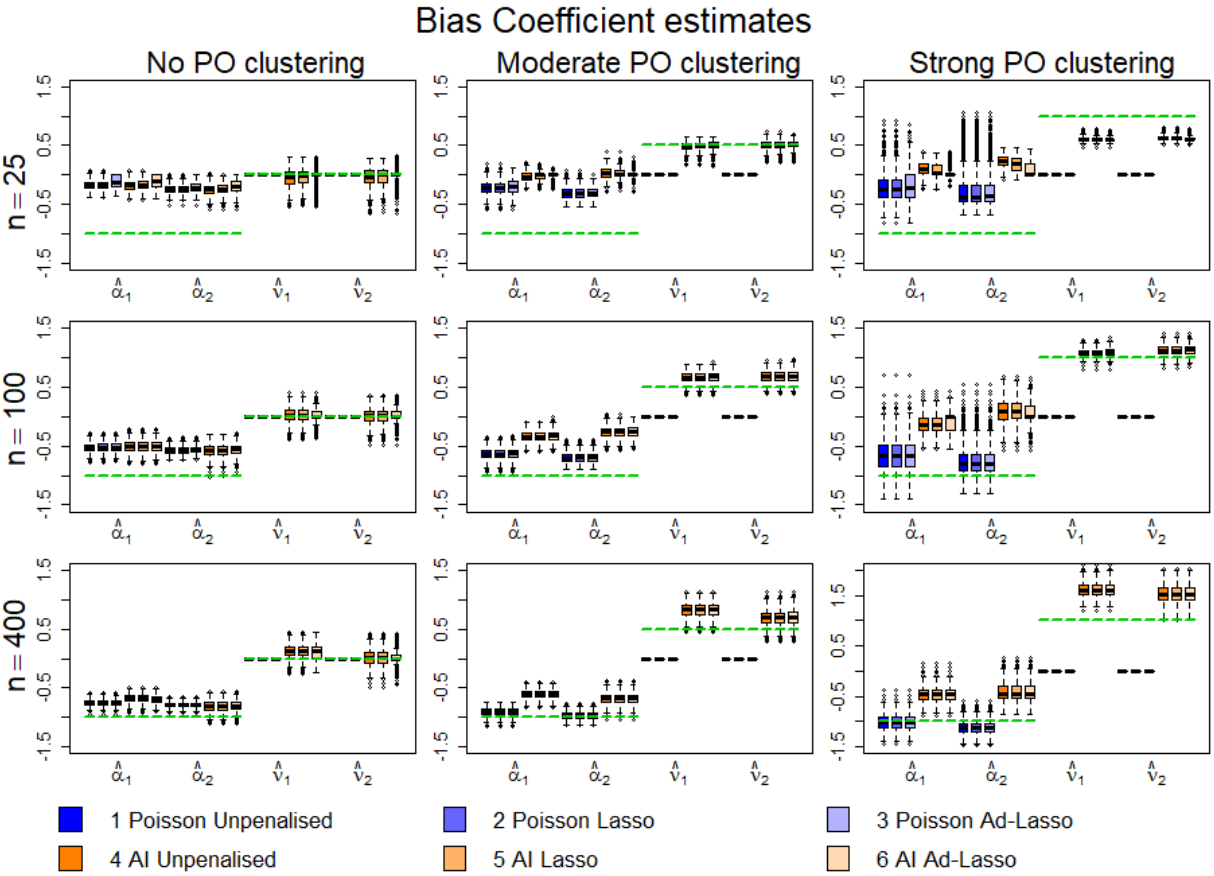


Figure A3: Boxpots of coefficient estimates corresponding to the two observer bias and two area-interaction covariates for the six models described in Table 1 of the manuscript. The green lines indicate the true values of the parameters used to generated the simulated data.

Although there could be improvements to be made by better disentangling the spatial signal from the environmental and bias signals, the incorporation of an area-interaction term when presence-only points are clustered beyond what can be explained by the other model covariates still represents a significant improvement over standard implementations of combined likelihood approaches to species distribution modelling.

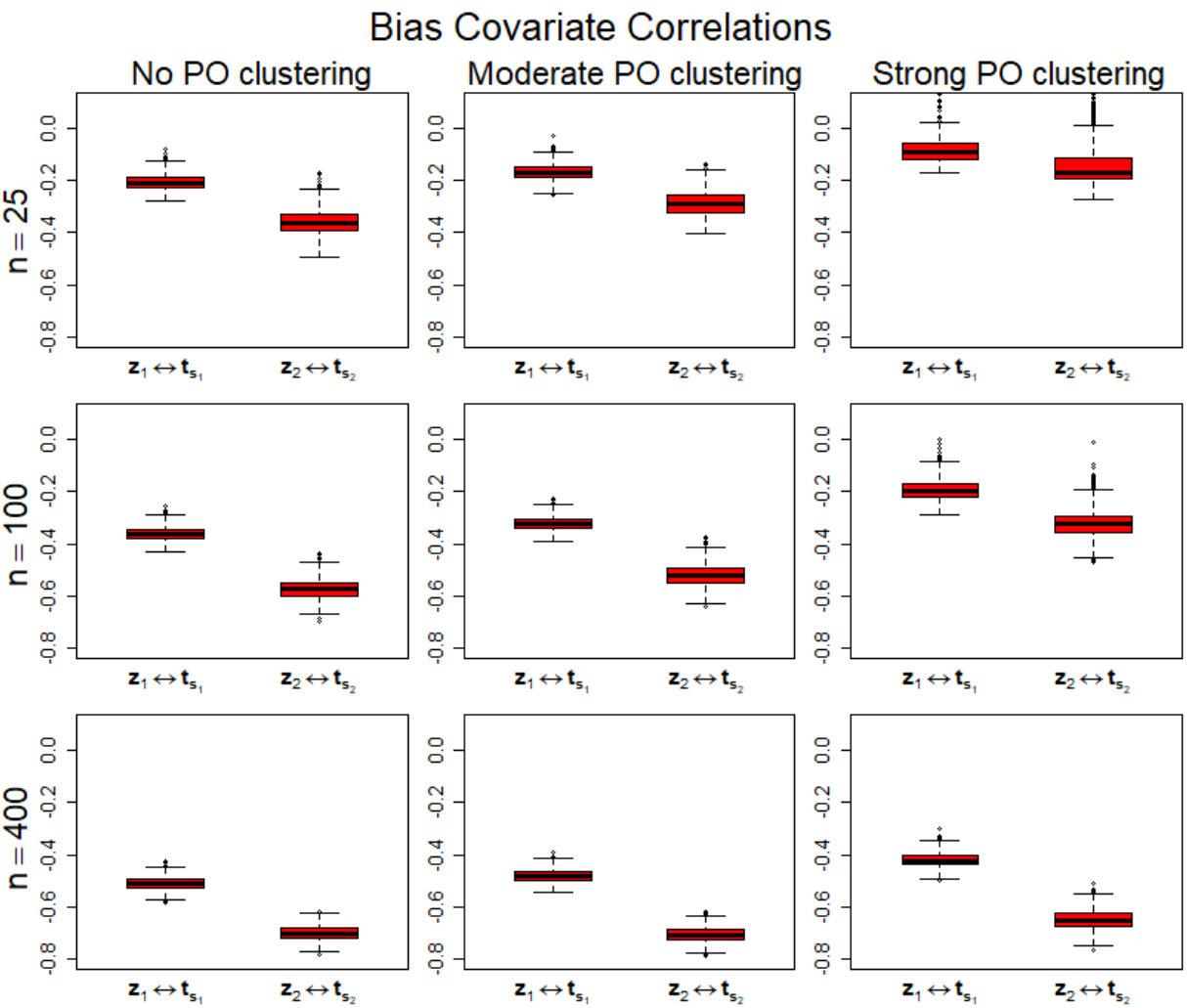


Figure A4: Boxpots of Pearson correlation coefficients between the observer bias and area-interaction covariates for both simulated data sources.

References

Warton, D.I., Renner, I.W. & Ramp, D. (2013) Model-based control of observer bias for the analysis of presence-only data in ecology. *PloS one*, **8**, e79168.

Demonstration of Optim Lasso Functions

Ian Renner, Julie Louvrier, and Olivier Gimenez

25 June 2019

This document demonstrates use of the `R` functions that were used to perform the simulations and the lynx analysis. These functions are contained in the file `Share Functions.R`. Here, we illustrate use of the functions through one of the simulations – in particular, the first simulation of presence-only patterns with 100 points that exhibit moderate clustering, as determined by the `strength` object. To ensure that roughly 100 points are generated, we set the intercepts for the two patterns with `int1` and `int2`. To ensure moderate clustering, we set the value of the interaction coefficient `intcoef`.

```
source("Share Functions.R")
seeduse = 1
strength = 3
N = 100
int1 = -2.25
int2 = -2.4
intcoef = 0.5
```

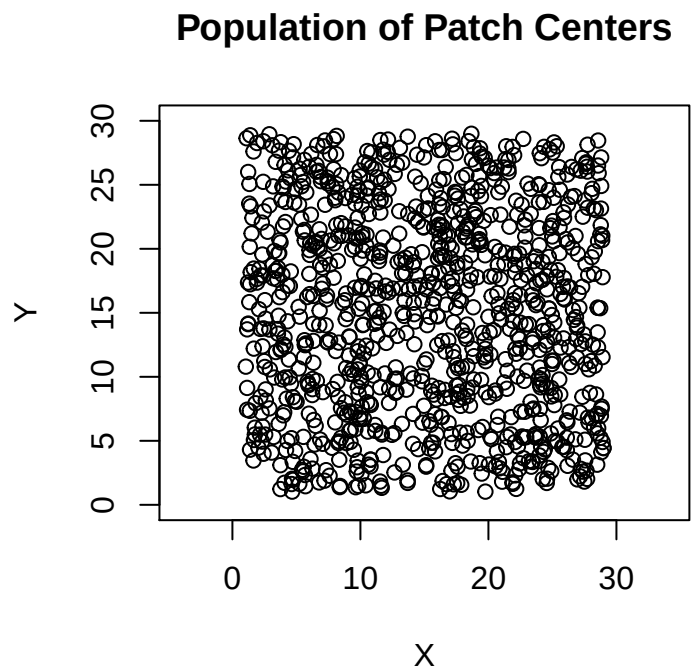
The functions we demonstrate make use of some functions of other packages, which we now load:

```
library(mvtnorm)
library(sp)
library(lattice)
library(spatstat)
library(data.table)
library(raster)
library(ROCR)
```

Generating environmental variables

First, we will generate the environmental covariates revelant to the simulated species distribution. First, we define our observation window as $[0, 30] \times [0, 30]$ and generate uniformly random "centers", from which we will randomly draw a subset to serve as the centres of habitat quality patches drawn from a truncated Gaussian distribution. These centers are set to be a distance of at least 1 unit from the window edge.

```
set.seed(3)
n_centers = 2000
centers = matrix(runif(n_centers, 1, 29), n_centers/2, 2)
plot(centers[,1], centers[,2], xlim = c(0, 30), ylim = c(0, 30),
     xlab = "X", ylab = "Y", main = "Population of Patch Centers",
     asp = 1)
```

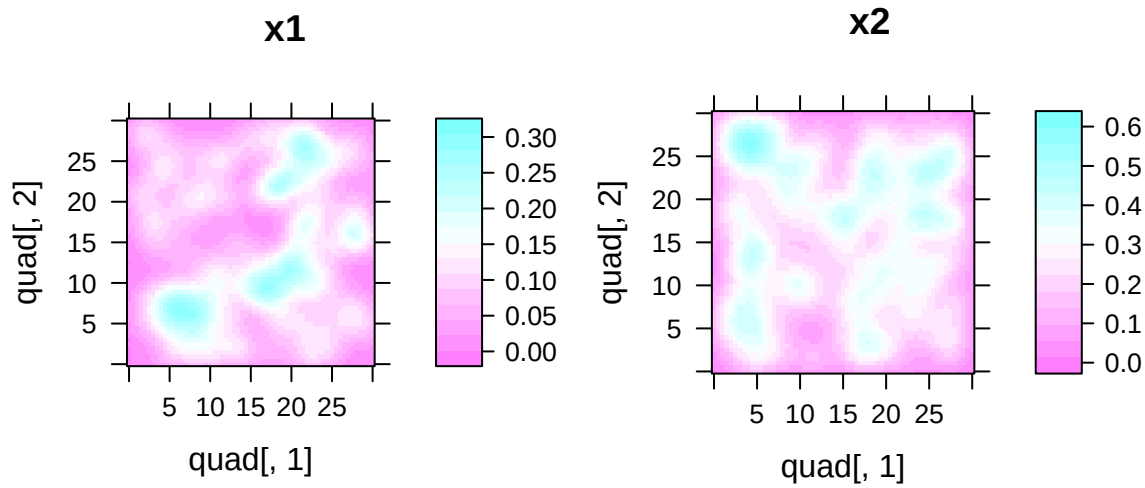


We also define a regular grid of quadrature points at a 0.5-unit spatial resolution throughout the observation window. It is at these quadrature points that we will measure the generated covariates.

```
quad = expand.grid(seq(0, 30, 0.5), seq(0, 30, 0.5))
names(quad) = c("X", "Y")
```

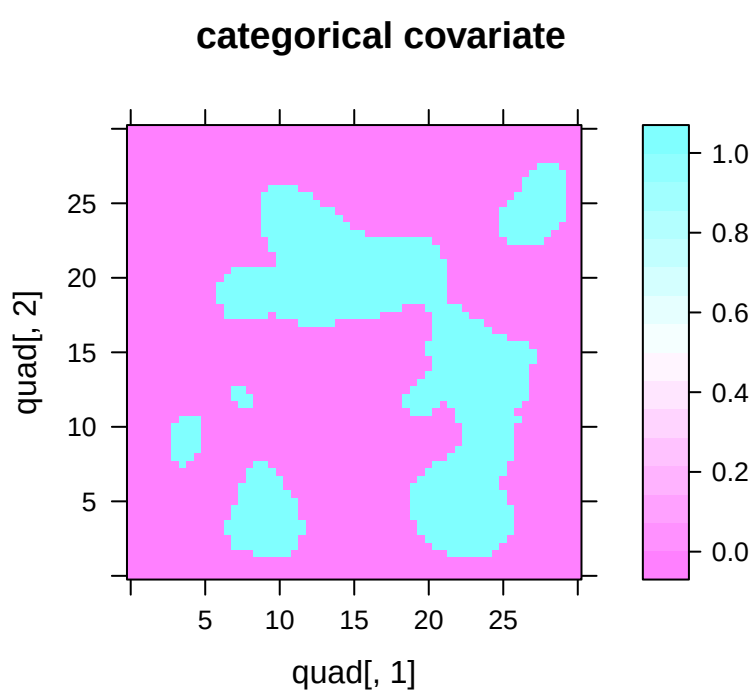
We now use the `makecovar` function to generate the two environmental covariates which define the species distribution:

```
set.seed(1)
x1 = makecovar(100)
set.seed(16)
x2 = makecovar(250)
```



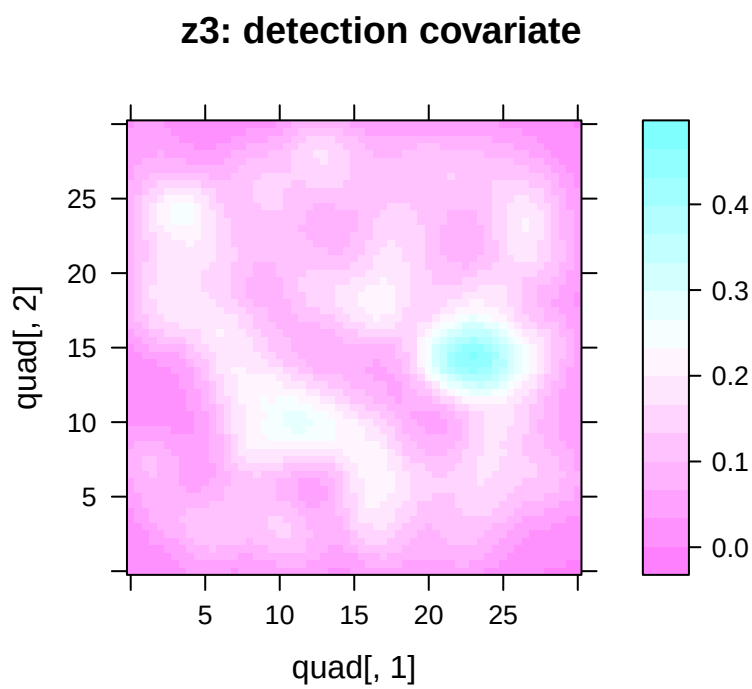
We now generate a covariate `catcov` to serve as a categorical covariate, the distance from which will be used to bias one of the presence-only data sources. First, we generate a continuous covariate `contcov` and categorise it through a threshold to generate `catcov`:

```
set.seed(18)
contcov = makecovar(100, centers, quad)
catcov = rep(0, length(contcov)) # categorical for PO bias
catcov[contcov >= quantile(contcov, 0.75)] = 1
levelplot(catcov ~ quad[,1] + quad[,2], main = "categorical covariate", asp = "iso")
```



We will also generate a covariate that defines detection probability for the occupancy data:

```r
set.seed(20)
z3 = makecovar(120, centers, quad) # detection for occupancy
levelplot(z3 ~ quad[,1] + quad[,2], main = "z3: detection covariate", asp = "iso")
```



We will also generate two "dummy" covariates that we will include in models which do not impact the true species distribution. This allows us to check the performance improvement that results from applying lasso-type penalties in the combined penalised likelihood.

```r
set.seed(5)
d1 = makecovar(50)
set.seed(15)
d2 = makecovar(50)
```

The seeds were chosen for all of the above covariates to ensure that none of the covariates are highly correlated:

```r
covframe = data.frame(x1, x2, contcov, catcov, z3, d1, d2)
cor(covframe)
```

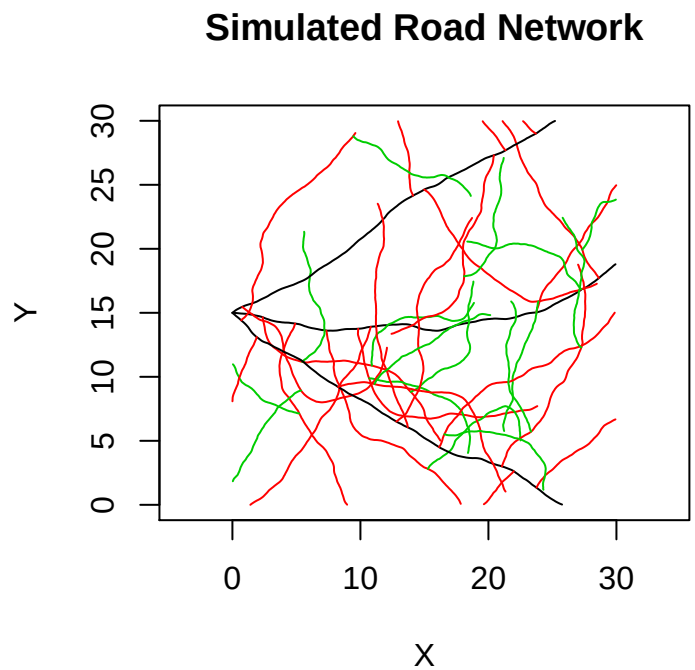
```
##                x1         x2   contcov     catcov           z3
## x1      1.0000000 0.33335245 0.2718674 0.16609826  0.290223677
## x2      0.3333524 1.00000000 0.3256248 0.18697137  0.395409641
## contcov 0.2718674 0.32562479 1.0000000 0.80270078  0.377083449
## catcov  0.1660983 0.18697137 0.8027008 1.00000000  0.307465718
## z3      0.2902237 0.39540964 0.3770834 0.30746572  1.000000000
## d1      0.1447537 0.02424448 0.1194169 0.07558192 -0.006555065
## d2      0.1955840 0.31041880 0.2102573 0.09821886  0.241281489
##                   d1          d2
## x1       0.144753684  0.19558399
## x2       0.024244476  0.31041880
## contcov  0.119416856  0.21025726
## catcov   0.075581920  0.09821886
```

```
## z3      -0.006555065  0.24128149
## d1       1.000000000 -0.13324797
## d2      -0.133247973  1.00000000
```

We will now simulate a road network in order to generate distances from the nearest road as a bias covariate for the first presence-only data source. The `addnetwork` function creates a network of roads. This function proceeds by adding n small road segments in random directions (normally distributed with a standard deviation of 3 initially centered at a value set by the `angle` argument). It randomly splits the road into junctions (a *parent* road and a *child* road), with the probability of a split set by the `p_split` argument and the angle of the child road a minimum of `min_angle` from the previous road. The number of levels (*generations*) of road splits is set by `max_levels`, with subsequent splits for each generation having split probabilities multiplied by a value set by the `split_decay` argument (to allow different levels of the network to have more or less frequent splits) and the standard deviation of angles between road segments multiplied by a value set by the `sdmult` argument (to allow different levels of the network to have different amounts of "wiggliness").

Here, we will generate 3 road networks of 3 levels each and combine them. Road segments belonging to levels 1, 2, and 3 are plotted in black, red, and green, respectively. The `fixnodes` function ensures that junctions aren't too close – any junctions within a small distance of each other are combined, producing a more realistic road network.

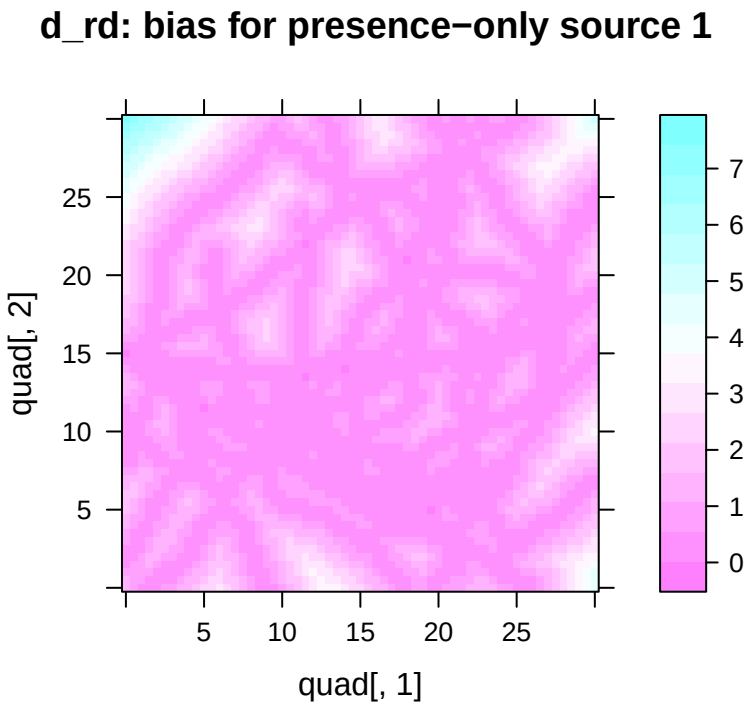
```r
set.seed(3)
network1 = addnetwork(0, 15, angle = 0, p_split = 0.03, split_decay = 0.2,
                      nmult = 0.6, min_angle = 60, sdmult = 1.5, max_levels = 3)
set.seed(12)
network2 = addnetwork(0, 15, angle = 40, p_split = 0.035, split_decay = 0.25,
                      nmult = 0.6, min_angle = 60, sdmult = 1.5, max_levels = 3)
set.seed(5)
network3 = addnetwork(0, 15, angle = -40, p_split = 0.015, split_decay = 0.2,
                      nmult = 0.6, min_angle = 60, sdmult = 1.5, max_levels = 3)
combined = c(network1, network2, network3)
combinednew = fixnodes(combined)
plotroadlist(combinednew, main = "Simulated Road Network", asp = 1)
```




```
combinednewnodes = nodelist(combinednew)
```

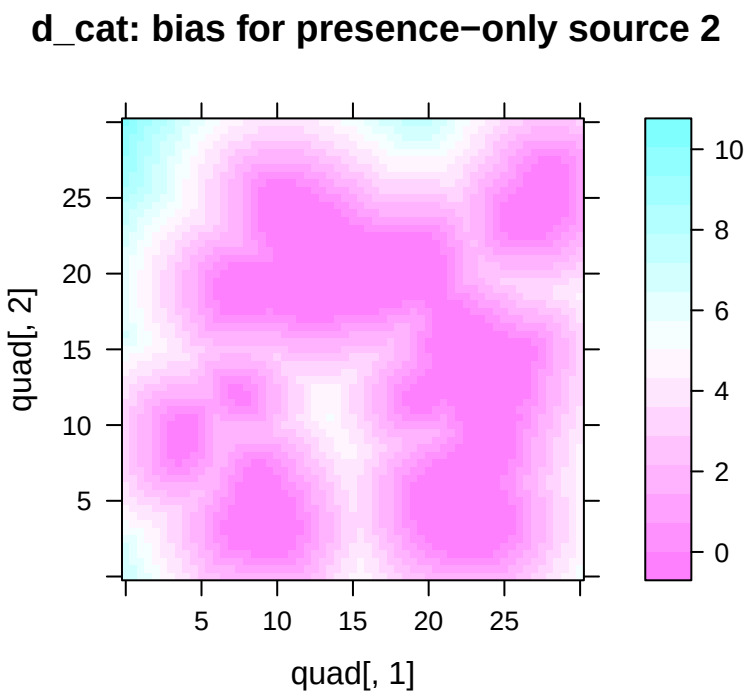
We now generate distances to the nearest road in the simulated road network as a bias variable `d_rd` ($\mathbf{z}_1$ in the manuscript) for the first presence-only data source.

```
allpts = ppp(x = c(quad$X, combinednewnodes$X),
             y = c(quad$Y, combinednewnodes$Y),
             marks = c(rep("Quad", dim(quad)[1]), rep("Road", dim(combinednewnodes)[1])),
             window = owin(c(0, 30), c(0, 30)))
nndists = nndist(allpts, k = 1, by = as.factor(marks(allpts)))
d_rd = nndists[1:dim(quad)[1], 2] # distance to the nearest road
levelplot(d_rd ~ quad[,1] + quad[,2], main = "d_rd: bias for presence-only source 1", asp = "iso")
```



We also generate distance to the categorical variable `catcov` as a bias variable `d_cat` ($\mathbf{z}_2$ in the manuscript) for the second presence-only data source.

```
catpts = ppp(x = quad$X, y = quad$Y,
             marks = paste("Cat", catcov, sep = ""),
             window = owin(c(0, 30), c(0, 30)))
catdists = nndist(catpts, k = 1, by = as.factor(marks(catpts)))
d_cat = catdists[,2]
d_cat[catcov == 1] = 0 # distance to categorical variable
levelplot(d_cat ~ quad[,1] + quad[,2], main = "d_cat: bias for presence-only source 2", asp = "iso")
```



We now standardise the covariates to have mean 0 and variance 1, so that the lasso-type penalties operate fairly on the coefficients.

```
x1 = scale(x1)
x1sq = scale(x1^2)
x2 = scale(x2)
x2sq = scale(x2^2)
z3 = scale(z3)
rt_d_rd = scale(sqrt(d_rd))
rt_d_cat = scale(sqrt(d_cat))
d1 = scale(d1)
d1sq = scale(d1^2)
d2 = scale(d2)
d2sq = scale(d2^2)
```

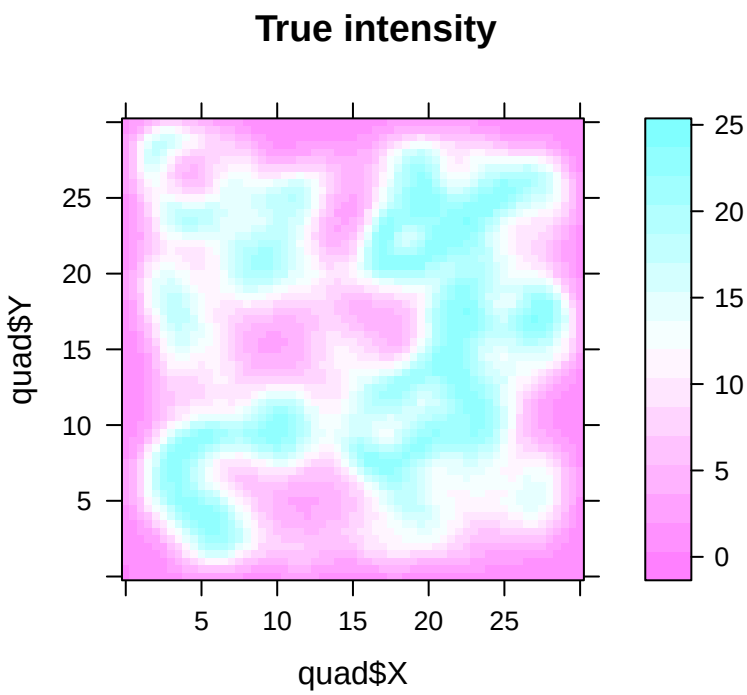
Generating point patterns

We now generate point patterns for the true species as well as two presence-only data sources as a function of the generated environmental and bias covariates. We will also generate a site history for use in an occupancy model.

```
vmat = as.matrix(data.frame(1, x1, x1sq, x2, x2sq,
                            rt_d_rd, contcov, catcov, rt_d_cat))
env.mat = vmat[,1:5] # vmat without covariates for bias
```

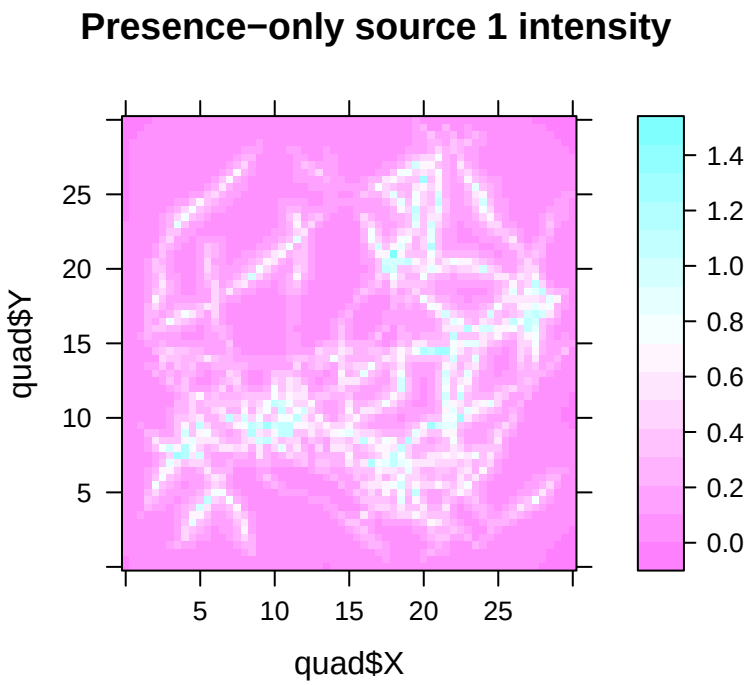
The true species intensity `sp_int` will make use of the environmental variables and an intercept (contained in `env.mat`):

```
sp_coef = c(2.15, 0.5, -0.3, 0.5, -0.3)
sp_int = exp(env.mat %*% sp_coef)
levelplot(sp_int ~ quad$X + quad$Y, asp = "iso", main = "True intensity")
```

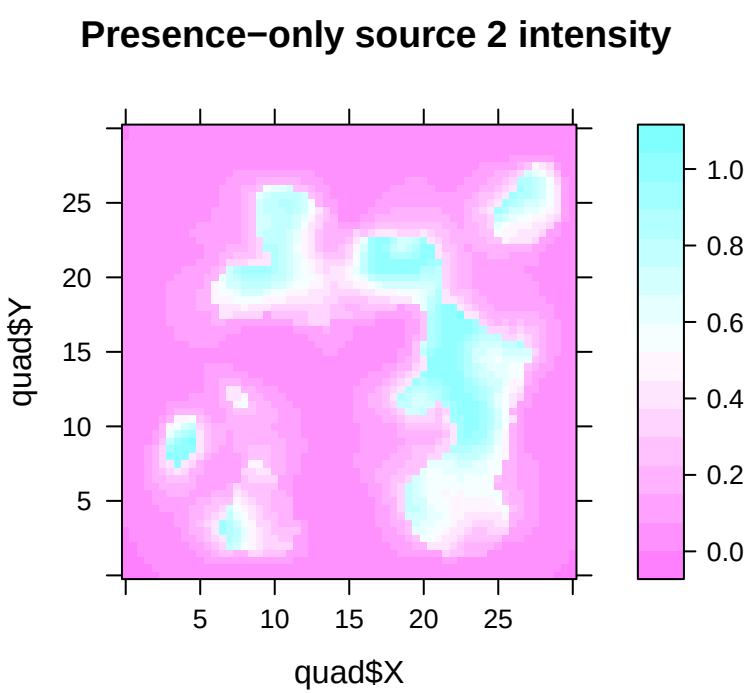


The two biased presence-only intensities will have the same environmental coefficients as the true intensity, but will be biased by `rt_d_rd` (pattern 1) and `rt_d_cat` (pattern 2):

```
po1_coef = c(int1, 0.5, -0.3, 0.5, -0.3, -1)
po1_int = exp(vmat[,1:6] %*% po1_coef)
levelplot(po1_int ~ quad$X + quad$Y, asp = "iso",
          main = "Presence-only source 1 intensity")
```




```
po2_coef = c(int2, 0.5, -0.3, 0.5, -0.3, -1)
po2_int = exp(vmat[,c(1:5, 9)] %*% po2_coef)
levelplot(po2_int ~ quad$X + quad$Y, asp = "iso",
          main = "Presence-only source 2 intensity")
```



We can see that presence-only source 1 has a strong bias toward the road network, and presence-only source 2 has a strong bias toward the categorical covariate. The `spatstat` package requires us to format these intensity surfaces as images in order to generate point patterns:

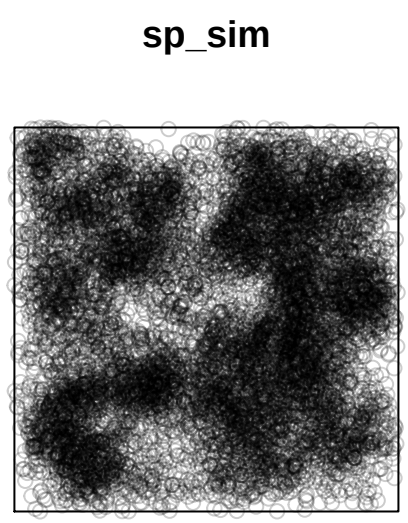
```
sp_int_im = as.im(data.frame(x = quad$X, y = quad$Y, z = sp_int))
po1_int_im = as.im(data.frame(x = quad$X, y = quad$Y, z = po1_int))
po2_int_im = as.im(data.frame(x = quad$X, y = quad$Y, z = po2_int))
```

We now generate the true species pattern with the `rpoispp` function of `spatstat`.

```
set.seed(seeduse)
sp_sim = rpoispp(sp_int_im) # true species pattern
sp_sim$n
```

```
## [1] 10349
```

```
sp_xy = data.frame(X = sp_sim$x, Y = sp_sim$y)
plot(sp_sim)
```



To fit models, we need to collect the values of the environmental and bias covariates at the grid of quadrature points and the species locations. First, we generate a data frame of quadrature point locations and associated covariates (`quads`), and then interpolate values at the species locations (`sp_env`) using the `newenv.var` function:

```
quads = data.frame(X = quad$X, Y = quad$Y, X1 = x1, X2 = x1sq,
                   X3 = x2, X4 = x2sq, Z3 = z3, rt_d_rd = rt_d_rd,
                   rt_d_cat = rt_d_cat, D1 = d1, D2 = d1sq,
                   D3 = d2, D4 = d2sq)
sp_env = newenv.var(sp.xy = sp_xy, env.grid = quads,
                    env.scale = 0.5, coord = c("X", "Y"), file.name = NA)
```

```
## Calculating species environmental data for variable: X1
```

```
## Calculating species environmental data for variable: X2
## Calculating species environmental data for variable: X3
## Calculating species environmental data for variable: X4
## Calculating species environmental data for variable: Z3
## Calculating species environmental data for variable: rt_d_rd
## Calculating species environmental data for variable: rt_d_cat
## Calculating species environmental data for variable: D1
## Calculating species environmental data for variable: D2
## Calculating species environmental data for variable: D3
## Calculating species environmental data for variable: D4
```

To generate the presence-only sources with clustering, we will proceed with an iterative algorithm that sequentially samples points from the species pattern `sp_sim` proportionally to conditional intensities and adds them to the set of observed locations. To compute conditional intensities, we essentially add an extra covariate – the *point interactions* of the pattern. For an area-interaction model, we can compute the point interaction at a location s by drawing circles of a given radius r around the location s and compute the proportion of the circle that overlaps with the circles of radius r around the other points in the pattern `s`. This proportion is edge-corrected for points near the border of the observation window. By setting the coefficient for this interaction term to be positive, we increase the conditional intensity at points that have high point interactions (i.e. points that are clustered around other observed points), encouraging clustering.

First, we initialise the interaction covariate to be a vector of zeros, and we sample the first point from the pattern `sp_sim` at random, with sampling probabilities proportional to the computed intensities at the locations in `sp_sim`.

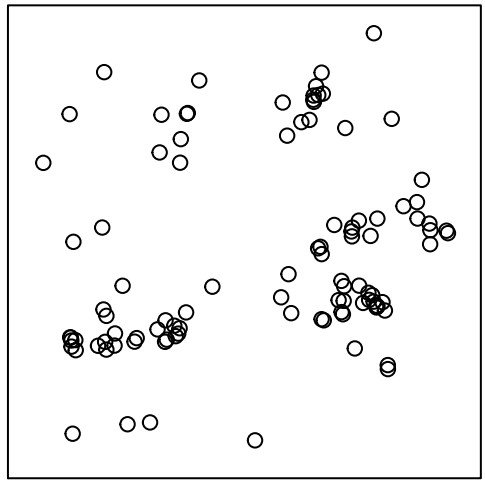
```
po1_X = data.frame(Intercept = 1, sp_env[,c(3:6, 8)], Interaction = 0)
po1_beta = c(int1, 0.5, -0.3, 0.5, -0.3, -1, intcoef)
po1_intensity = exp(as.matrix(po1_X) %*% po1_beta)
PO1_rows = c() #vector indicating the sampled rows
X_add = sample(1:sp_sim$n, 1, prob = po1_intensity)
PO1_rows = c(PO1_rows, X_add)
PO1 = sp_sim[PO1_rows]
```

We then compute the point interactions for all of the points in `sp_sim` based on the first sampled point, and standardise this vector as we standardised the environmental and bias covariates.

```
ints = evalInteraction(PO1, sp_sim, interaction = AreaInter(1), correction = "border")
stand_ints = scale(ints)
```

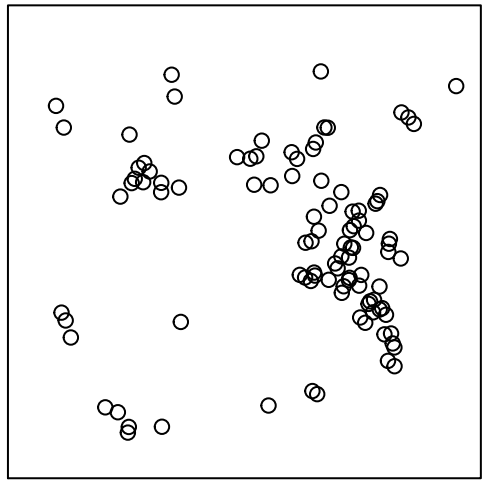
We now iterate through steps to add the remaining points. We first redefine the vector of point interactions, compute the conditional intensity `int_i` (setting it to 0 for the points already sampled to ensure they aren't resampled), sampling a new point based on probabilities proportional to the updated conditional intensities, and recomputing standardised point interactions based on the new set of points.

```
for (i in 2:N)
{
  po1_X$Interaction = stand_ints
  int_i = exp(as.matrix(po1_X) %*% po1_beta)
  int_i[PO1_rows] = 0
  X_add = sample(1:sp_sim$n, 1, prob = int_i)
  PO1_rows = c(PO1_rows, X_add)
  PO1 = sp_sim[PO1_rows]
  ints = evalInteraction(PO1, sp_sim, interaction = AreaInter(1), correction = "border")
  stand_ints = scale(ints)
}
po1_xy = data.frame(X = PO1$x, Y = PO1$y)
plot(PO1)
```



We follow the same procedure to sample points for the second presence-only data set, biased by distance from the categorical covariate.

```
po2_X = data.frame(Intercept = 1, sp_env[,c(3:6, 9)], Interaction = 0)
po2_beta = c(int2, 0.5, -0.3, 0.5, -0.3, -1, intcoef)
po2_intensity = exp(as.matrix(po2_X) %*% po2_beta)
PO2_rows = c()
X_add = sample(1:sp_sim$n, 1, prob = po2_intensity)
PO2_rows = c(PO2_rows, X_add)
PO2 = sp_sim[PO2_rows]
ints = evalInteraction(PO2, sp_sim, interaction = AreaInter(1), correction = "border")
stand_ints = scale(ints)
for (i in 2:N)
{
  po2_X$Interaction = stand_ints
  int_i = exp(as.matrix(po2_X) %*% po2_beta)
  int_i[PO2_rows] = 0
  X_add = sample(1:sp_sim$n, 1, prob = int_i)
  PO2_rows = c(PO2_rows, X_add)
  PO2 = sp_sim[PO2_rows]
  ints = evalInteraction(PO2, sp_sim, interaction = AreaInter(1), correction = "border")
  stand_ints = scale(ints)
}
po2_xy = data.frame(X = PO2$x, Y = PO2$y)
plot(PO2)
```



We now set up a grid of sites to be used in an occupancy model. We set up 100 sites along a regular 1.5×1.5-unit grid:

```
occ_sites = expand.grid(seq(1.5, 28.5, 3), seq(1.5, 28.5, 3))
occ_x = occ_sites[,1]
occ_y = occ_sites[,2]
```

We then determine whether each site is occupied by the species or not by computing distances between the site centers and the nearest point in the true species pattern. Any site center within a distance of 0.18 units from the nearest point in `sp_sim` is considered to be occupied, and the other sites are considered unoccupied.

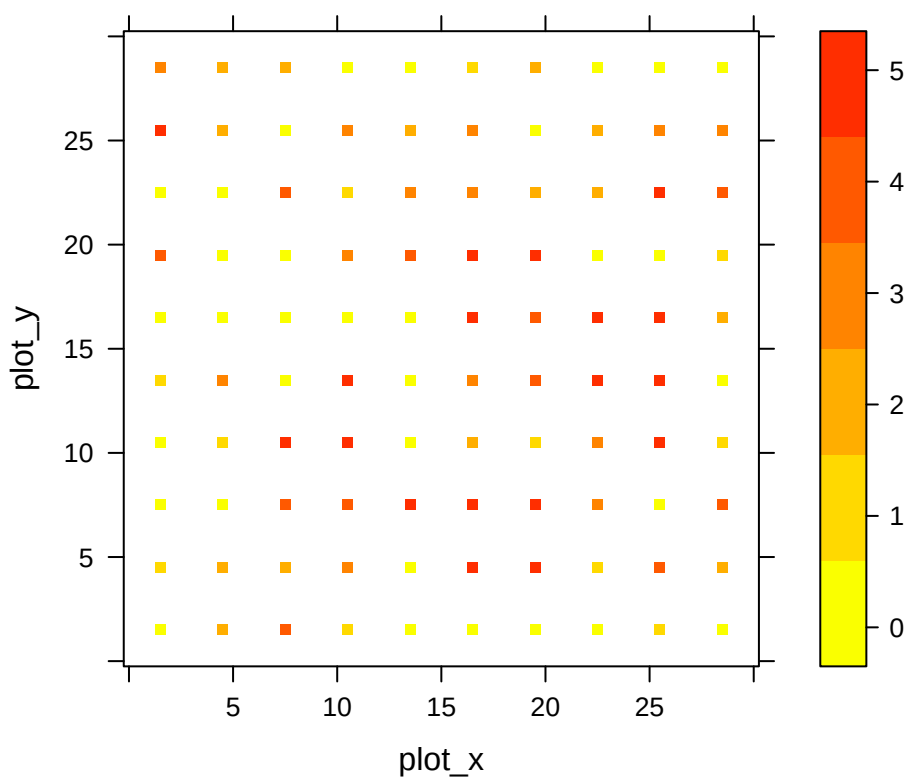
```
sp_and_occ = ppp(x = c(occ_x, sp_sim$x), y = c(occ_y, sp_sim$y),
                 marks = c(rep("Occ", length(occ_x)), rep("Sp", sp_sim$n)),
                 window = owin(c(-0.25, 30.25), c(-0.25, 30.25)))
dist_sp_occ = nndist(sp_and_occ, k = 1, by = as.factor(marks(sp_and_occ)))
# compute distance to nearest species
sp_occ_dists = dist_sp_occ[1:length(occ_x), 2]
# species considered present at site if nearest one is within 0.25 units
occ_present = as.numeric(sp_occ_dists <= 0.18)
```

To determine detection probability at the sites, we map them to the quadrature points, which contain the values of all of the covariates. We use the inverse of the complementary log-log function to define detection probability. We set detection probability to 0 for those sites considered unoccupied by the species.

```
occ_paste = paste(occ_x, occ_y)
quad_paste = paste(quad$X, quad$Y)
occ_quadrow = match(occ_paste, quad_paste)
p_detect = clogloginv(z3[occ_quadrow])*occ_present
```

We now generate a history `sim_history` of detections and non-detections at each site for 5 visits, randomly generating detections and non-detections according to the computed detection probabilities:

```
n_visits = 5
sim_history = matrix(as.integer(matrix(rep(p_detect, times = n_visits),
                                       length(occ_x), n_visits) >
                                  matrix(runif(length(occ_x)*n_visits),
                                         length(occ_x), n_visits)),
                     length(occ_x), n_visits)
site_sum = apply(sim_history, 1, sum)
plot_xy = expand.grid(seq(0, 30, 0.5), seq(0, 30, 0.5))
plot_x = plot_xy[,1]
plot_y = plot_xy[,2]
plot_id = paste(plot_x, plot_y)
occ_id = paste(occ_x, occ_y)
occ_match = match(occ_id, plot_id)
plot_z = rep(NA, length(plot_x))
plot_z[occ_match] = site_sum
levelplot(plot_z ~ plot_x + plot_y, asp = "iso", cuts = 5,
          col.regions = rainbow(6, start = 0.03, end = 0.17)[6:1])
```



Fitting models

To fit models, we need the values of the environmental and bias variables at the species locations in the two presence-only data sets as well as at the locations of the occupancy sites. Since the presence-only locations are subsets of the pattern `sp_sim`, for which we already derived the values of the covariates in `sp_env`, we just need to extract the appropriate rows of the `sp_env` data frame.

```
po1_env = sp_env[PO1_rows,]
po2_env = sp_env[PO2_rows,]
```

Finally, we generate the values of the covariates at the occupancy sites by subsetting the appropriate rows from `quads`, since the occupancy sites were set up to be directly on one of the quadrature points:

```
occ_env = quads[occ_quadrow,]
```

We now create some objects that will be used in the multiple calls to the `comb_lasso` function to fit models. In particular, we specify the environmental formula used to parameterise $\boldsymbol{\beta}$ as well as the formulas for the various bias terms used to parametrise $\boldsymbol{\alpha}$. We specify the interaction radius for models fitted with the area-interaction likelihood with the `int.radius` object.

```
env_formula = ~ X1 + X2 + X3 + X4 +
  D1 + D2 + D3 + D4
bias_formula = list(~ rt_d_rd, ~ rt_d_cat, ~ Z3)
n.fits = 1000
int.radius = c(1, 1)
tol = 1.e-9
b.min = 1.e-6
link = "cloglog"
sp_res = 0.5
```

We first generate models with the inhomogeneous Poisson point process model presence-only likelihood. The arguments are as follows:

- `intercept_env`: whether to include (1) or not (NA) an intercept for the environmental component of the model
- `intercept_bias`: a list of indicators for inclusion of intercepts in the various bias components of the model
- `quad_data`: a data frame for the quadrature data with columns containing values for all covariates
- `sp_data`: a list of data frames for the species data with columns containing values for all covariates
- `sp_y`: a list of objects containing the response variables for each species data source. For presence-only data sources, this is a vector of 1s. For occupancy models, it is a matrix of detections and non-detections.
- `dat.type`: a vector of species data types, with "PO" indicating presence-only and "Occ" indicating occupancy.
- `coord`: names for the columns that include the x- and y-coordinates in the `sp_data` objects
- `sp_res`: spatial resolution to be used in fitting the presence-only point process models
- `penalty_vec`: a pre-supplied vector of penalties to be used. If `NULL`, this vector will be computed automatically
- `alpha`: the elastic net coefficient. If `alpha` is set to 1, a lasso penalty will be applied. If `alpha` is set to 0, a ridge regression penalty will be applied. If `alpha` is between 0 and 1, an elastic net penalty will be applied, with the value of `alpha` controlling the lasso portion of the penalty.
- `gamma`: exponent for adaptive lasso weights
- `init.coef`: initial coefficients used to determine adaptive lasso weights. If `NULL`, a standard lasso is fitted
- `standardise`: a logical value indicating whether the covariates should be standardised to have mean 0 and variance 1 prior to model fitting
- `criterion`: which criterion is used to determine the optimal model among the fitted models. Set to "BIC" by default but AIC ("AIC"), corrected AIC ("AICc") and the Hannah-Quinn criterion ("HQC") are also permitted.
- `family`: the canonical link family used in the point process models – set to Poisson by default
- `tol`: a tolerance threshold controlling convergence of the lasso algorithm. The lasso is fitted by iteratively reweighted least squares, and if the difference in likelihoods between successive iterations falls below `tol`, the model is considered to have converged
- `b.min`: the smallest absolute value of β permitted. The function sets any estimated coefficient with an

absolute value less than `b.min` to 0

- `max.it`: the maximum number of iterations of the iteratively reweighted least squares algorithm for the lasso
- `n.fits`: the number of models to fit, if `penalty_vec` is not specified.
- `noshrink`: a vector indicating the covariates not to be penalised. If `NULL`, only the intercept terms will be unpenalised.
- `method`: the optimisation method used by `optim`
- `link`: the link function for the detection covariate – either "cloglog" or "logit" is permitted.
- `site.area`: the area of the occupancy sites
- `area.int`: a logical value indicating the form of the presence-only likelihood components. If `TRUE`, area-interaction model likelihoods are used for each presence-only source. If `FALSE`, an inhomogeneous Poisson PPM likelihood is used instead.
- `r`: The radius of the point interactions if `area.int` is set to `TRUE`
- `wt.vec`: A vector of source weights to be applied to each likelihood component. By default, set to `NULL` such that the weight is 1 for each source.
- `pen.min`: The smallest non-zero penalty to be applied. By default, this is set to 0.01.

```
comb_ppm = comb_lasso(env_formula = env_formula,
                      bias_formula = bias_formula,
                      intercept_env = 1,
                      intercept_bias = list(NA, NA, NA),
                      quad_data = quads,
                      sp_data = list(po1_env, po2_env, occ_env),
                      sp_y = list(rep(1, nrow(po1_env)), rep(1, nrow(po2_env)), sim_history),
                      dat.type = c("PO", "PO", "Occ"),
                      coord = c("X", "Y"),
                      sp_res = sp_res,
                      penalty_vec = NULL,
                      alpha = 1, gamma = 0, init.coef = NA, standardise = TRUE,
                      family = "poisson", tol = tol, b.min = b.min,
                      max.it = 25, n.fits = n.fits, noshrink = NULL, method = "BFGS",
                      link = link, site.area = pi*0.18^2,
                      area.int = FALSE, r = NULL)
```

```
## [1] "Output saved in the file Quad0.5.RData"
```

To fit a model with an adaptive lasso penalty, we specify a value for the adaptive weight exponent `gamma` as well as a vector of initial coefficients `init.coef`:

```
comb_ad_ppm = comb_lasso(env_formula = env_formula,
                      bias_formula = bias_formula,
                      intercept_env = 1,
                      intercept_bias = list(NA, NA, NA),
                      quad_data = quads,
                      sp_data = list(po1_env, po2_env, occ_env),
                      sp_y = list(rep(1, nrow(po1_env)), rep(1, nrow(po2_env)), sim_history),
                      dat.type = c("PO", "PO", "Occ"),
                      coord = c("X", "Y"),
                      sp_res = sp_res,
                      penalty_vec = NULL,
                      alpha = 1, gamma = 1, init.coef = comb_ppm$betas[,1],
                      standardise = TRUE,
                      family = "poisson", tol = tol, b.min = b.min,
                      max.it = 25, n.fits = n.fits, noshrink = NULL, method = "BFGS",
                      link = link, site.area = pi*0.18^2,
                      area.int = FALSE, r = NULL)
```

```
## [1] "Output saved in the file Quad0.5.RData"
```

We can fit models with an area-interaction term by setting the `area.int` argument to `TRUE` and supplying a vector of radii for the point interactions to the `r` argument:

```
comb_ai = comb_lasso(env_formula = env_formula,
                     bias_formula = bias_formula,
                     intercept_env = 1,
                     intercept_bias = list(NA, NA, NA),
                     quad_data = quads,
                     sp_data = list(po1_env, po2_env, occ_env),
                     sp_y = list(rep(1, nrow(po1_env)), rep(1, nrow(po2_env)), sim_history),
                     dat.type = c("PO", "PO", "Occ"),
                     coord = c("X", "Y"),
                     sp_res = sp_res,
                     penalty_vec = NULL,
                     alpha = 1, gamma = 0, init.coef = NA, standardise = TRUE,
                     family = "poisson", tol = tol, b.min = b.min,
                     max.it = 25, n.fits = n.fits, noshrink = NULL, method = "BFGS",
                     link = link, site.area = pi*0.18^2,
                     area.int = TRUE, r = c(int.radius, NA))
```

```
## [1] "Output saved in the file Quad0.5.RData"
## Calculating point interactions
## Calculating point interactions
```

We can also fit models using just a single source of data, though such models are not considered in the manuscript. As an example, the following fits models using only the first presence-only data source:

```
po1_ppm = comb_lasso(env_formula = env_formula,
                         bias_formula = list(bias_formula[[1]]),
                         intercept_env = 1,
                         intercept_bias = NULL,
                         quad_data = quads,
                         sp_data = list(po1_env),
                         sp_y = list(rep(1, nrow(po1_env))),
                         dat.type = c("PO"),
                         coord = c("X", "Y"),
                         sp_res = sp_res,
                         penalty_vec = NULL,
                         alpha = 1, gamma = 0, init.coef = NA, standardise = TRUE,
                         family = "poisson", tol = tol, b.min = b.min,
                         max.it = 25, n.fits = n.fits, noshrink = NULL, method = "BFGS",
                         link = link, site.area = pi*0.18^2,
                         area.int = FALSE, r = NULL)
```

```
## [1] "Output saved in the file Quad0.5.RData"
```

Exploring Fitted Models

The created objects contain a lot of information about the fitted models, including the fitted coefficients for all models (`betas`), the vector of penalties (`penalty_vec`), the penalised likelihoods (`pen_likelihoods`), the values of the various criteria to choose the lasso penalty (`criterion_matrix`), the fitted intensity (`mu`), occupancy probability (`psi`), and detection probability (`p_detect`) of the optimal model, the runtime (`runtime`) and the data used in the model fitting.

We can examine the fitted coefficients for the model which optimised BIC:

```
comb_ppm$beta
```

```
##   Intercept          X1          X2          X3          X4          D1
## -2.74433124  0.83598140 -0.40490977  0.75378210 -0.41387000 -0.06719399
##          D2          D3          D4     rt_d_rd    rt_d_cat          Z3
## -0.08224648 -0.02960134 -0.01281307 -0.53227289 -0.72330739  0.93211371
```

```
comb_ad_ppm$beta
```

```
##   Intercept          X1          X2          X3          X4          D1
## -2.73712493  0.83219212 -0.40722838  0.73666386 -0.39834125 -0.03346626
##          D2          D3          D4     rt_d_rd    rt_d_cat          Z3
## -0.08568419  0.00000000  0.00000000 -0.53113523 -0.72172138  0.92847091
```

```
comb_ai$beta
```

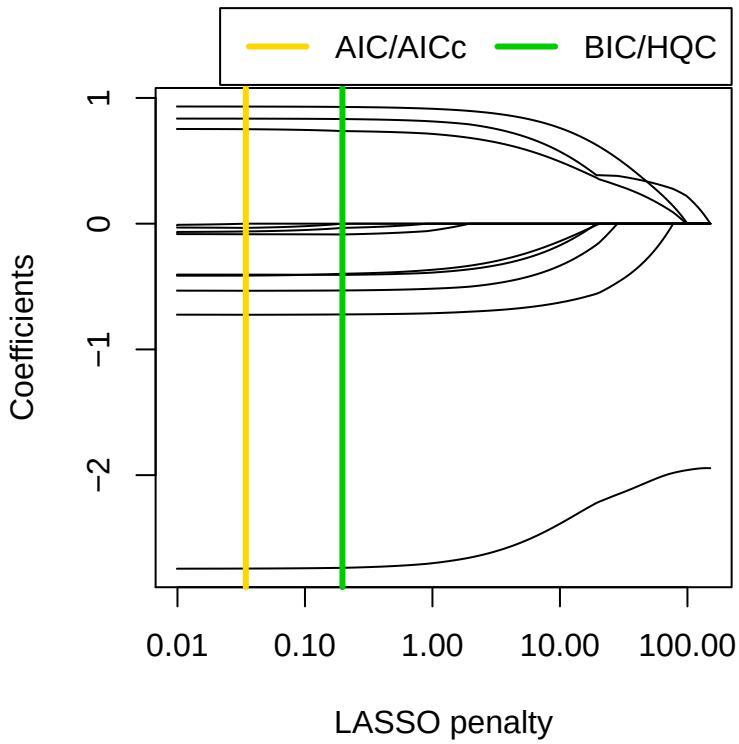
```
##   Intercept          X1          X2          X3          X4          D1
## -2.69681660  0.41282015 -0.20131157  0.43632514 -0.23054465  0.00000000
##          D2          D3          D4     rt_d_rd Interaction    rt_d_cat
## -0.07917936 -0.08556838  0.02062521 -0.26909853  0.67793812 -0.23966321
## Interaction          Z3
##  0.69291701  0.92757279
```

```
po1_ppm$beta
```

```
##    Intercept           X1           X2           X3           X4
## -4.186028204  1.812580070 -0.925689009  1.386875530 -0.755216546
##           D1           D2           D3           D4      rt_d_rd
##  0.148990250 -0.398124979 -0.153440610 -0.004871374 -0.663559637
```

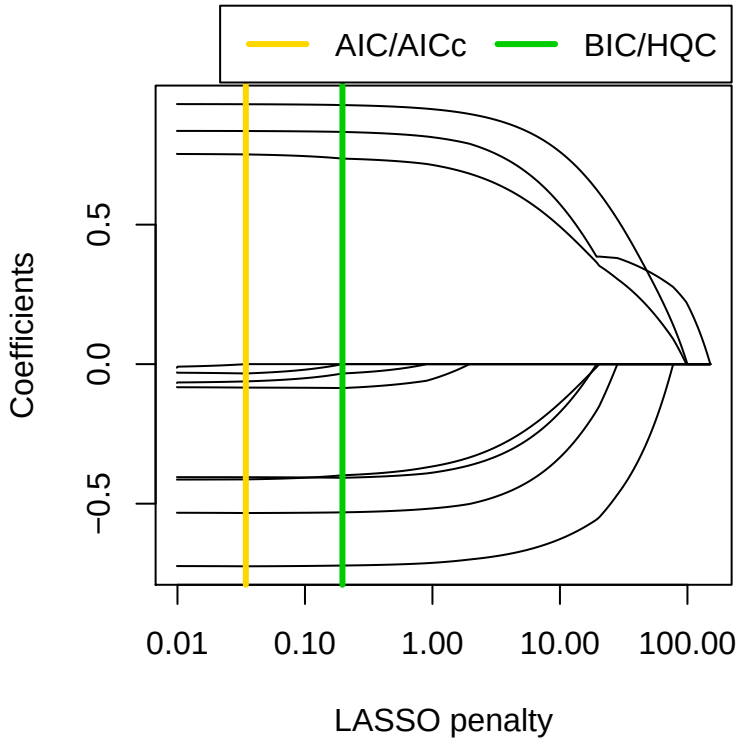
We may wish to see some details of the *regularisation path* of fitted models. To view a plot of the coefficients as the penalty increases, we can use the `plotpath` function. For the combined model with IPPPM likelihoods fitted with an adaptive lasso penalty:

```
plotpath(comb_ad_ppm)
```



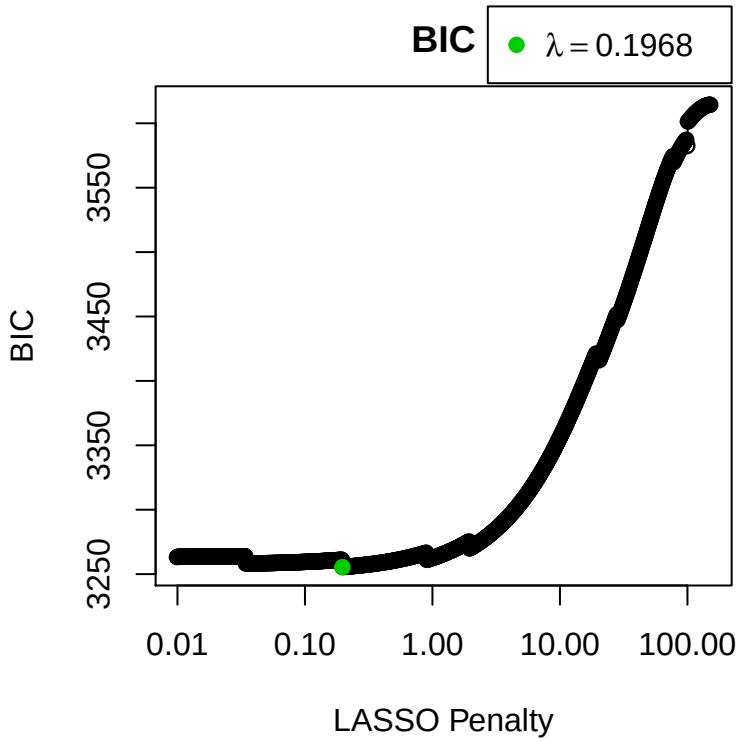
We can wish to view the regularisation path without the intercept in this case, as it is not peanlised. To do so, we supply the index of the intercept to the `v.cut` argument:

```
plotpath(comb_ad_ppm, v.cut = which(names(comb_ppm$beta) == "Intercept"))
```



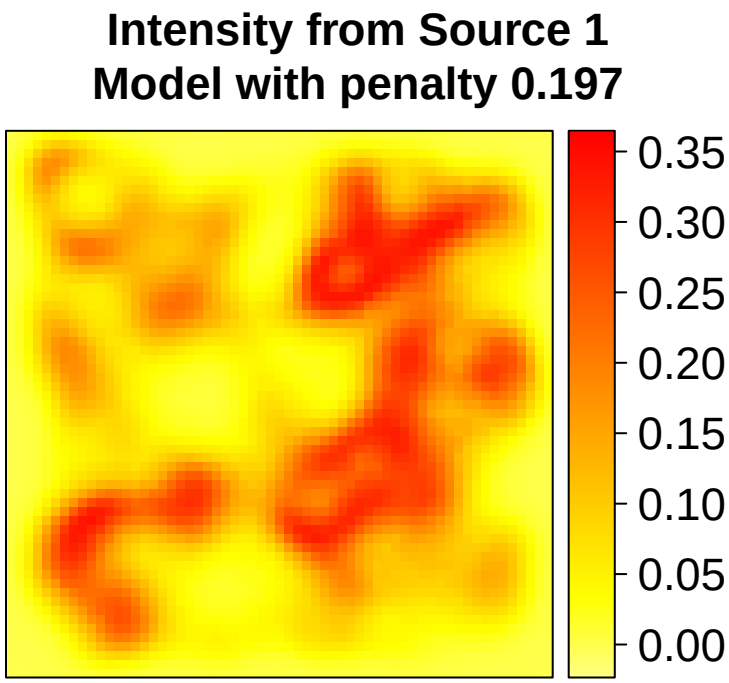
We can also plot the value of the criterion used to choose the best model (in this case BIC) for all fitted models with the `criterion_curve` function:

```
criterion_curve(comb_ad_ppm)
```



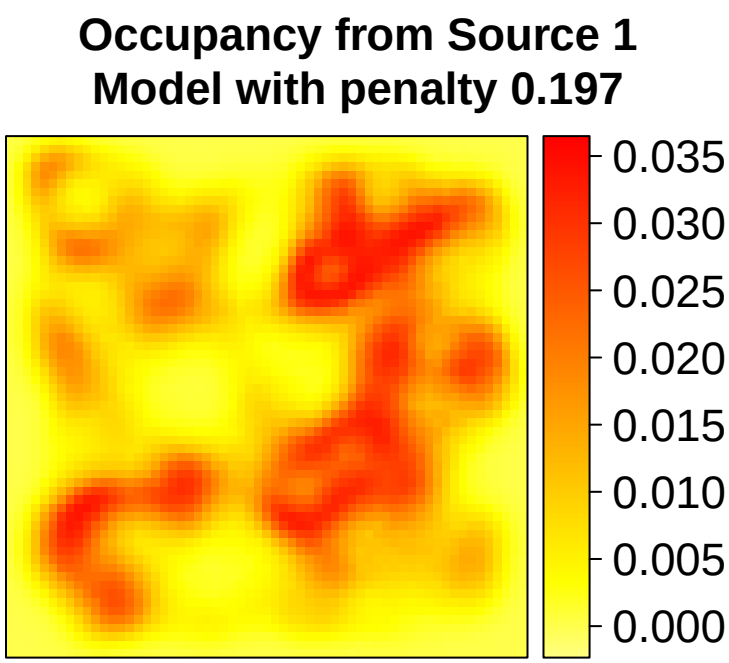
The `plotfit` function allows us to produce heatmaps for various quantities for our optimal fitted model. To plot the fitted intensity for the optimal model:

```
plotfit(comb_ad_ppm)
```



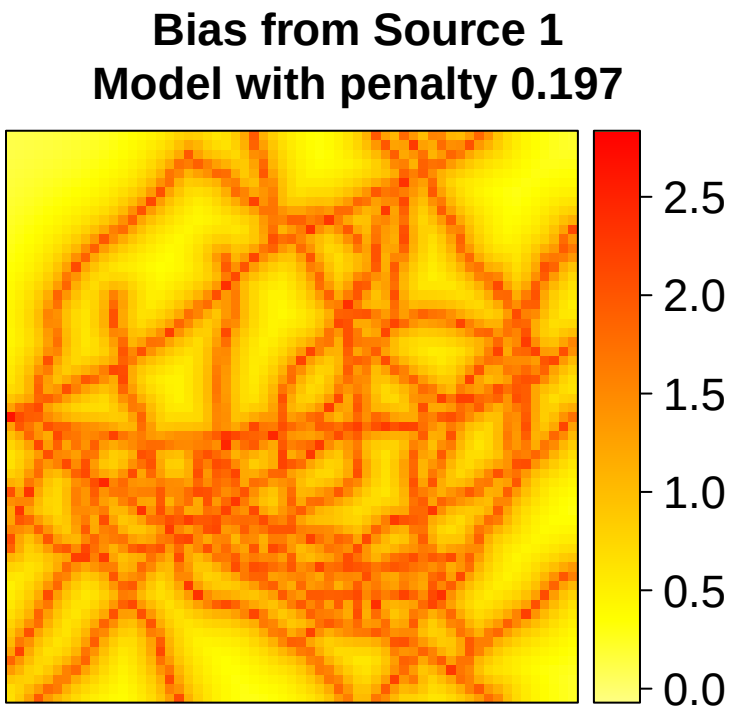
To plot the occupancy probability:

```
plotfit(comb_ad_ppm, z = "occupancy", link = "cloglog")
```

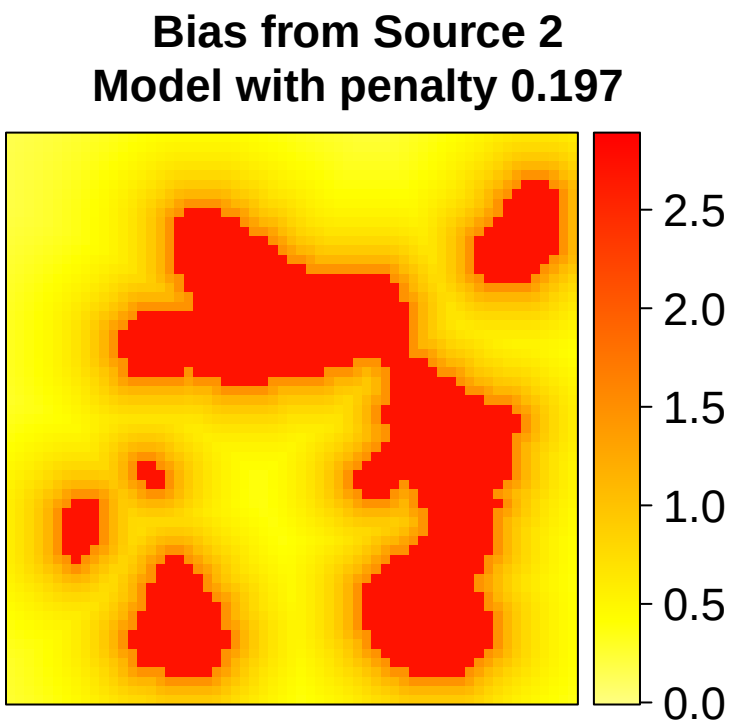


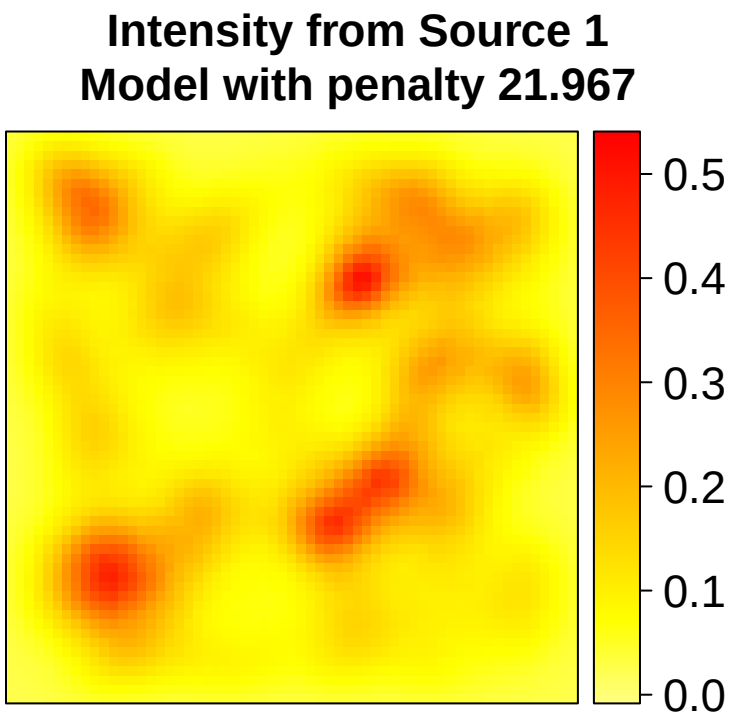
To plot the estimated bias for each source:

```r
plotfit(comb_ad_ppm, z = "bias", source = 1)
```




```r
plotfit(comb_ad_ppm, z = "bias", source = 2)
```



We can also choose to plot these quantities for any of the fitted models. For example, the fitted intensity for the 800th fitted model:

```
plotfit(comb_ad_ppm, model = 800)
```