Variable selection for species distribution modeling: an example with the western spadefoot toad

Kevin Neal

June 12, 2015

Contents

Introduction

Species distribution modeling (SDM), or sometimes "ecological niche modeling," (ENM), is a tool for predicting species distributions and for determining environmental or other correlates of a species' presence in a given location. These models use a combination of spatial environmental layers, where each pixel has a value of some environmental parameter, and species occurrence points for its predictions. A popular program for conducting these is Maxent (Phillips et al. 2006; Phillips and Dudík 2008), which models species occupancy probabilities on a landscape using maximum entropy.

As explained by Elith and colleagues (2011), Maxent uses environmental covariate data from known presence/occurrence points and from a random sampling of background points across a landscape ("pseudoabsences") to estimate the ratio of the covariate distribution at presence sites f1(z) to the covariate distribution of the background f(z), choosing f1(z) among possible distributions by minimizing the distance to f(z) (minimizing this distance can be thought of as minimizing the distance to a null model without any known occurrence data). The distance between the covariate distribution of the presence points and the background is the Kullback-Leibler divergence, i.e. the relative entropy of f1(z). The name for Maxent (the program) derives from the program's maximizing the entropy of the "raw" geographic distribution of occupancy probabilities, which is equivalent to minimizing the relative entropy of the environmental covariate distribution f1(z) relative to f(z) (Phillips et al. 2006; Elith et al. 2011).

To most ecologists, these details of the Maxent "black box" are an afterthought. Indeed, that had been my situation. I began using Maxent with a tangible goal of using species distribution modeling to predict past, present, and future distributions of the western spadefoot toad, Spea hammondii. The initial motivation for doing SDM's with Spea hammondii at all involved a peculiar discordance between evolutionary trees of the genus Spea generated from different genes. There are four *Spea* species: *hammondii*, *multiplicata*, *intermontana*, and *bombifrons*; all are found only in western North America; *hammondii* has no current range overlap with the others, while the other three do have some overlap throughout their respective ranges. In a phylogeny generated from eight nuclear genes, one sees a pattern of *S. hammondii* (SHAM hereafter) as the

outgroup to the remaining three. However, using a mitochondrial DNA phylogeny, all SHAM individuals in the southern portion of their range come out as the sister group to *S. multiplicata* (SMULT hereafter). One hypothesis for why this may be is mitochondrial introgression from SMULT into only the southern population of SHAM, with the introgression failing to make its way north of the Tehachapi Mountains in central California. The problem with this hypothesis of introgression is that it requires there to have been interbreeding between SHAM and SMULT, when they are currently completely separated by the Mojave Desert (and no individuals of any *Spea* species are known from the Mojave). If introgression truly occurred, then, it must have taken place under different climatic circumstances under which the ranges overlapped. With the availability of global climate models from different points in the recent geologic past, to test the hypothesis I decided to use SDMs to predict the ranges of both species at representative climatic slices: a period of peak glaciation (Last Glacial Maximum, ~21ka), minimum glaciation (Last Interglacial, ~120ka), and an intermediate stage of glaciation (mid-Holocene, ~6ka). If distribution models under these different climatic scenarios showed expansions that led to the overlap of the two species, although it would not prove interbreeding, it would suggest that mitochondrial introgression could remain a viable hypothesis for the observed phylogenetic patterns.

In the literature where Maxent has been applied to real species, many if not most of the 4000+ publications use what have become the standard environmental covariates for modeling species distributions, known as "bioclim" layers (Beaumont et al. 2005; Booth et al. 2014). Bioclim layers are nineteen environmental variables generated from worldwide temperature and precipitation records presumed to be biologically relevant (that is, some combination of these variables determines, or helps to determine, the bounds of species range limits). These Bioclim layers are easily accessible and widely used, and the precedent of simply throwing all nineteen into the Maxent black box perpetuates the same; this in spite of the fact that many of the variables are often highly correlated with one another. Maxent has been described as robust to this sort of overparameterization through its inbuilt method for regularization, with Elith et al. (2011) noting Maxent "is unlikely to be improved – and more likely, degraded – by procedures that use other modelling methods to pre-select variables." Even if this is the case [and my personal dealings with it actually show this to be accurate], variable selection is likely to be important when the Maxent model is used to make predictions in different regions or climates that have novel combinations of variable values outside the bounds of the input environmental parameter space. Because in my research I am using Maxent to predict occurrences in past and future climate scenarios, variable selection ought to be important for creating a more generalizable model (to say nothing of compliance with philosophical parsimony).

For this project I explored variable selection for species distribution modeling of *Spea hammondii* using both stepwise regression and mutual information. For project manageability/as a first-run dive into understanding the applications of regression and information theoretic statistics to variable selection for an SDM outside of a black box, I used generalized linear modeling (GLM) in R, rather than Maxent, to predict species occurrence probability across the landscape. I additionally used bootstrapping to examine variability in the models, and compared models using the area under the receiver operating characteristic curve (AUC).

Materials and Methods

To obtain presence records of SHAM, I downloaded raw occurrence data from the Global Biodiversity Information Facility (GBIF). I then cleaned up this data (raw data has 200+ columns to avoid any information loss from the different field notes of researchers) to get just Longitude and Latitude. I went through a number of clean-up steps in R for quality control to remove spatially-aberrant points or entries with missing data. To avoid biasing the models' determination of habitat suitability from the presence points, I split the study area into grid cells and subsampled a single occurrence point from each grid; this substantially reduced the number of occurrence entries (794 to 88) but is a necessary step to control for biased spatial sampling (F. Dormann et al. 2007; VanDerWal et al. 2009; Stokland et al. 2011; Barbet-Massin et al. 2012; Syfert et al. 2013; Warren et al. 2014).

I generated pseudoabsence points first by randomly sampling 1000 points from the region within 300 km of any given presence point, excluding an area of 10km around any given presence point. I then followed the same grid-sampling procedure as with the presence points to limit spatial autocorrelation. While there are

arguments for both a large and small amount of pseudoabsences in SDM (VanDerWal et al. 2009), to limit computation times in later analyses I arbitrarily subsampled this down to double the number of presence points, at 176 pseudoabsences.

Using the coordinates of the presence and pseudoabsence points I then extracted values from the nineteen bioclim layers into a dataframe. For the response variable I assigned a value of "1" to presence points and "0" to pseudoabsences.

To get a sense of the correlations among the nineteen bioclim variables, in R I calculated Spearman correlation using the cor function in the corrplot package (Wei 2013), and I calculated the maximal information criteria (MIC) using the mine function in the minerva package (Filosi et al. 2014). I plotted these correlations using the corrplot function in corrplot.

I utilized mutual information (mutinformation function in package infotheo (Meyer 2014)) as suggested in Ivannikova et al (2013) to select predictors by iteratively adding predictors to the function (can think of it in a way as a forward stepwise regression, but using mutual information). The strategy is to maximize the information with the least amount of variables, with X as the predictor variables (bioclim) and Y as the response variable (i.e. presence/absence). I began the process with running mutinformation() on the whole dataframe to see which single predictor variable had the highest mutual information value with the response variable; this became the first variable selected for the model. Subsequently, I looked for the greatest information addition adding a single other predictor, one at a time, among the remaining predictors. This method precludes highly correlated variables, because the addition of a new predictor that is highly correlated with one already included will offer little to no new information, even if it is itself highly correlated with the response variable on its own—and thus the method iteratively adds variables that offer new information to the model until the information plateaus.

As a comparison, I utilized the built-in step function to implement a backwards stepwise GLM regression of the bioclim variables and presence/absence to determine the optimal combination of variables, as determined by the lowest Akaike Information Criterion (AIC) value. The stepwise regression works by calculating the AIC of models in which variables are iteratively removed, until the lowest possible AIC is achieved.

I generated three GLM's, modeled using a binomial distribution to conduct logistic regression, as appropriate for a binary response variable. The models are: a "full" model using all nineteen bioclim variables; the "step" model using those variables producing the lowest AIC in the stepwise regression; and the third being the "MI" model using those variables determined as optimal using mutual information. I then wrote two bootstrapping functions to look at sampling effects and to generate null prediction maps. These functions bootstrap resample from the presence points and the absence points separately (so each iteration keeps the same number of presence and absence points): the standard bootstrap function resamples whole rows so a given point has the same environmental values, while the null bootstrap function shuffles data within columns, eliminating any correlations. Within the functions, the model put into the function is re-run with the resampled datapoints, and the output of the functions is the map of the median probability of all bootstrapped predictions, accomplished with the predict function in dismo (Hijmans et al. 2015). These maps, along with the prediction maps generated from the original GLM's, allow for easy spatial visualization of probabilities of occurrence, given the model. Finally I used the evaluate function in dismo to calculate the area under the receiver operator curve (AUC). The area under these curves is assumed to be a measure of model performance, with higher AUC indicating a better model.

Results and discussion

Correlation and variable selection

Both Spearman correlation and maximal information criteria reveal strong correlations between several variable pairs within the nineteen bioclim variables, validating the concern over collinearity and model overfitting in a species distribution model of Spea hammondii in the sampled environmental space.

Using mutual information to guide variable selection, bio1 (annual mean temperature) is the first best predictor of presence or absence. The forward-stepping method settles on a GLM of presence ~ bio1 + bio2 +

bio4 + bio12 + bio15 + bio16 + bio17 + bio18, with the mutual information between the predictors and response equal to 0.631. Among these variable pairs, bio1+bio17 and bio12+bio16 still appear to be highly correlated (MIC=0.710 and 0.918, respectively). Additional variables did not increase the information, and testing MI between presence and all nineteen variables also returns MI = 0.631, indicating the eight chosen variables contain as much information as all nineteen. This method of variable selection may be imperfect, however, as it fails to consider interactions between variables (which may be accomplished with conditional mutual information (Fleuret 2004), but its application is beyond the scope here). The AIC of the MI-informed GLM is 216.44.

The GLM model with the lowest AIC according to backwards stepwise regression is presence ~ bio1 + bio4 + bio9 + bio14 + bio15 + bio17 + bio18 + bio19, with AIC = 191.45. Among these variable pairs, bio1+bio17 and bio14+bio17 still appear to be highly correlated (MIC=0.710 and 0.790, respectively).

The AIC of the GLM with all nineteen variables is 207.25.

Model predictions

The ecology literature considers a model with AUC > 0.8 to be a "good" model (Phillips and Dudík 2008; Elith et al. 2011): all three models (full, step, MI) exceed this and can be considered "good." All prediction maps stumble in areas where there were no samples (presence or absence) taken (see e.g. lower right in all maps), demonstrating the pitfalls of predicting into areas of novel environments. Median bootstrapped prediction maps do not differ substantially from their unbootstrapped counterparts, although they produce more conservative predictions, with the predicted probabilities of occurrence outside the sampling range notably reduced. The MAD maps (not shown, but were evaluated) also reveal areas that appear to be sensitive to sampling in the model by indicating particularly high levels of deviation from the median. In practice it is not clear to me that bootstrapping the samples and re-running the model is the best method, as it will introduce spatial autocorrelation, but permuting with an arbitrary subsample does not seem ideal either. This is likely a case where cross-validation would be the appropriate way to test the robustness of the model (Hijmans 2012), and I will explore this in future pursuits.

Qualitatively, if we ignore the overreaching predictions into novel environments, all three GLM models do an adequate job of reconstructing the understood range of *Spea hammondii*, and quantitatively the AUC scores agree. Ranking the models in terms of AUC score, the GLM from the stepwise regression is the best (AUC=0.927), followed by the full model (AUC=0.923), and with the MI-informed GLM in last (AUC=0.902). This outcome on the surface may make one hesitant to dive deeper into applying information theoretic methods to species distribution modeling, but Maxent's success as an information theoretic model already makes it clear that these methods are truly capable. My use of mutual information in this project was merely in selecting variables for an optimal model, and not used in the predictions or evaluation of the model itself. So while this exercise of variable selection using generalized linear models has been extremely useful in expanding my familiarity with regression and prediction, the next step will be examining these reduced-predictor datasets in Maxent itself and evaluating their utility to my continued research in amphibian ecology and conservation.

References

Barbet-Massin, M., F. Jiguet, C. H. Albert, and W. Thuiller. 2012. Selecting pseudo-absences for species distribution models: how, where and how many? Methods Ecol. Evol. 3:327–338.
Beaumont, L. J., L. Hughes, and M. Poulsen. 2005. Predicting species distributions: use of climatic parameters in BIOCLIM and its impact on predictions of species' current and future distributions. Ecol. Model. 186:251–270.
Booth, T. H., H. A. Nix, J. R. Busby, and M. F. Hutchinson. 2014. bioclim: the first species distribution modelling package, its early applications and relevance to most current MaxEnt studies. Divers. Distrib. 20:1–9.
Elith, J., S. J. Phillips, T. Hastie, M. Dudík, Y. E. Chee, and C. J. Yates. 2011. A statistical explanation of

MaxEnt for ecologists. Divers. Distrib. 17:43–57.
F. Dormann, C., J. M. McPherson, M. B. Araújo, R. Bivand, J. Bolliger, G. Carl, R. G. Davies, A. Hirzel, W. Jetz, W. Daniel Kissling, I. Kühn, R. Ohlemüller, P. R. Peres-Neto, B. Reineking, B. Schröder, F. M. Schurr, and R. Wilson. 2007. Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. Ecography 30:609–628.
Filosi, M., R. Visintainer, D. Albanese, S. Riccadonna, G. Jurman, and C. Furlanello. 2014. minerva: minerva: Maximal Information-Based Nonparametric Exploration R package for Variable Analysis.
Fleuret, F. 2004. Fast Binary Feature Selection with Conditional Mutual Information. J Mach Learn Res 5:1531–1555.
Hijmans, R. J. 2012. Cross-validation of species distribution models: removing spatial sorting bias and calibration with a null model. Ecology 93:679–688.
Hijmans, R. J., S. Phillips, and J. L. and J. Elith. 2015. dismo: Species Distribution Modeling.
Ivannikova, E., T. Hamalainen, and K. Luostarinen. 2013. Information-theoretic approach to variable selection in predictive models applied to paper machine data. Pp. 000946–000950 in 2013 IEEE Symposium on Computers and Communications (ISCC).
Meyer, P. E. 2014. infotheo: Information-Theoretic Measures.
Phillips, S. J., R. P. Anderson, and R. E. Schapire. 2006. Maximum entropy modeling of species geographic distributions. Ecol. Model. 190:231–259.
Phillips, S. J., and M. Dudík. 2008. Modeling of species distributions with Maxent: new extensions and a comprehensive evaluation. Ecography 31:161–175.
Stokland, J. N., R. Halvorsen, and B. Støa. 2011. Species distribution modelling—Effect of design and sample size of pseudo-absence observations. Ecol. Model. 222:1800–1809.
Syfert, M. M., M. J. Smith, and D. A. Coomes. 2013. The Effects of Sampling Bias and Model Complexity on the Predictive Performance of MaxEnt Species Distribution Models. PLoS ONE 8:e55158.
VanDerWal, J., L. P. Shoo, C. Graham, and S. E. Williams. 2009. Selecting pseudo-absence data for presence-only distribution modeling: How far should you stray from what you know? Ecol. Model. 220:589–594.
Warren, D. L., A. N. Wright, S. N. Seifert, and H. B. Shaffer. 2014. Incorporating model complexity and spatial sampling bias into ecological niche models of climate change risks faced by 90 California vertebrate species of concern. Divers. Distrib. 20:334–343.
Wei, T. 2013. corrplot: Visualization of a correlation matrix.

SUPPLEMENT: R-scripts and figures

Loading the spadefoot toad presence and absence point data:

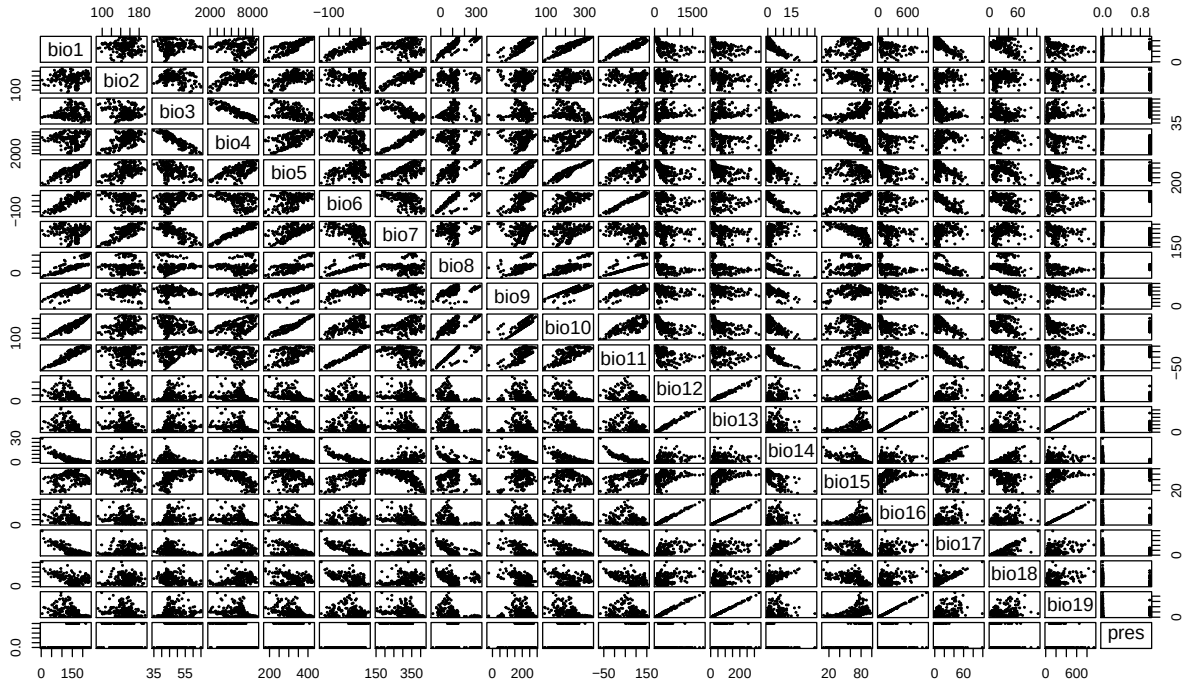
```
# using spea and bioclim variables
spea.ll <- read.csv("speaLatLonBioclim.csv") # dataframe generated by extracting bioclim variable value
spea.ll <- spea.ll[-1]
spea <- spea.ll[-c(1:2)] # creates spea which is ONLY bioclim variables, no latlon

colnames(spea.ll)[1] <- "lon" # simplifying column names
colnames(spea.ll)[2] <- "lat"
spea2 <- spea
```

```
present <- rep(1, nrow(spea))
spea2[,20] <- present
colnames(spea2)[20] <- "pres"
#summary(spea)
#corrplot(cor(spea, method="spearman"))
#pairs(spea)

### "absence" points (sampled from random background points on same grid as presence points, using ches

absent.ll <- read.csv("speaabsentLatLonBioclim.csv")
absent.ll <- absent.ll[-1]
absent <- absent.ll[,-c(1:2)] # creates dataframe of ONLY bioclim variables, no latlon
colnames(absent.ll)[1] <- "lon" # simplifying column names
colnames(absent.ll)[2] <- "lat"
absent2 <- absent

absences <- rep(0, nrow(absent))
absent2[,20] <- absences
colnames(absent2)[20] <- "pres"

allpts <- rbind(spea2, absent2)
hammy <- allpts # hammy refers to hammondii. This is the dataframe with all the presence AND absence po
hammy.ll <- cbind(hammy, rbind(spea.ll[,c(1:2)], absent.ll[,c(1:2)])) # the dataframe INCLUDING latlon.

spea.llonly <- spea.ll[,c(1:2)]
absent.llonly <- absent.ll[,c(1:2)]
```

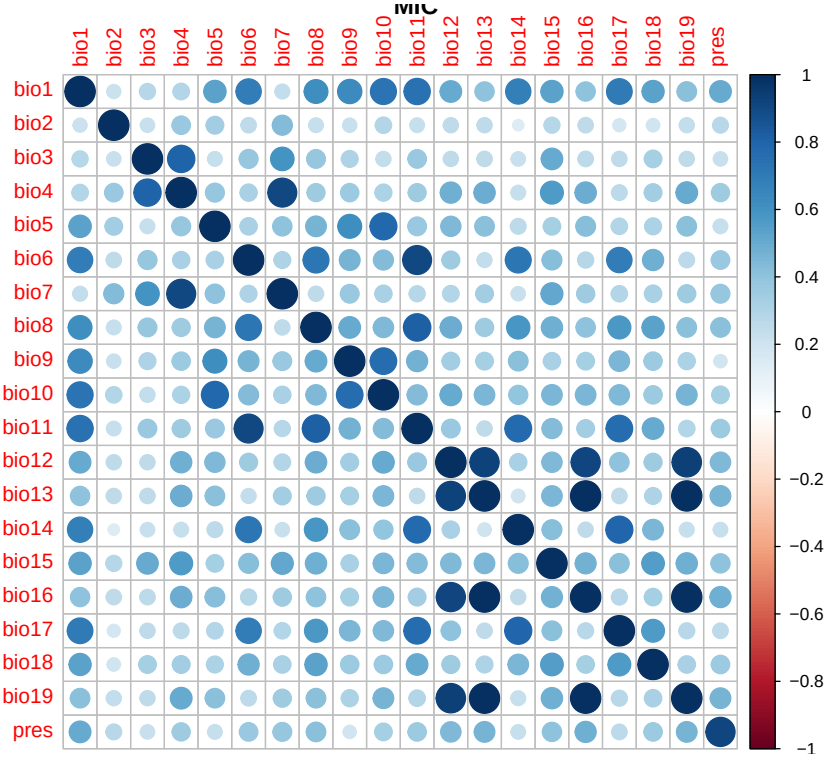
The nineteen bioclim layers are:
BIO1 = Annual Mean Temperature
BIO2 = Mean Diurnal Range (Mean of monthly (max temp - min temp))
BIO3 = Isothermality (BIO2/BIO7) (* 100)
BIO4 = Temperature Seasonality (standard deviation *100)*
BIO5 = Max Temperature of Warmest Month
BIO6 = Min Temperature of Coldest Month
BIO7 = Temperature Annual Range (BIO5-BIO6)
BIO8 = Mean Temperature of Wettest Quarter
BIO9 = Mean Temperature of Driest Quarter
BIO10 = Mean Temperature of Warmest Quarter
BIO11 = Mean Temperature of Coldest Quarter
BIO12 = Annual Precipitation
BIO13 = Precipitation of Wettest Month
BIO14 = Precipitation of Driest Month
BIO15 = Precipitation Seasonality (Coefficient of Variation)
BIO16 = Precipitation of Wettest Quarter
BIO17 = Precipitation of Driest Quarter
BIO18 = Precipitation of Warmest Quarter
BIO19 = Precipitation of Coldest Quarter

Correlation plots using MIC and Spearman correlation:

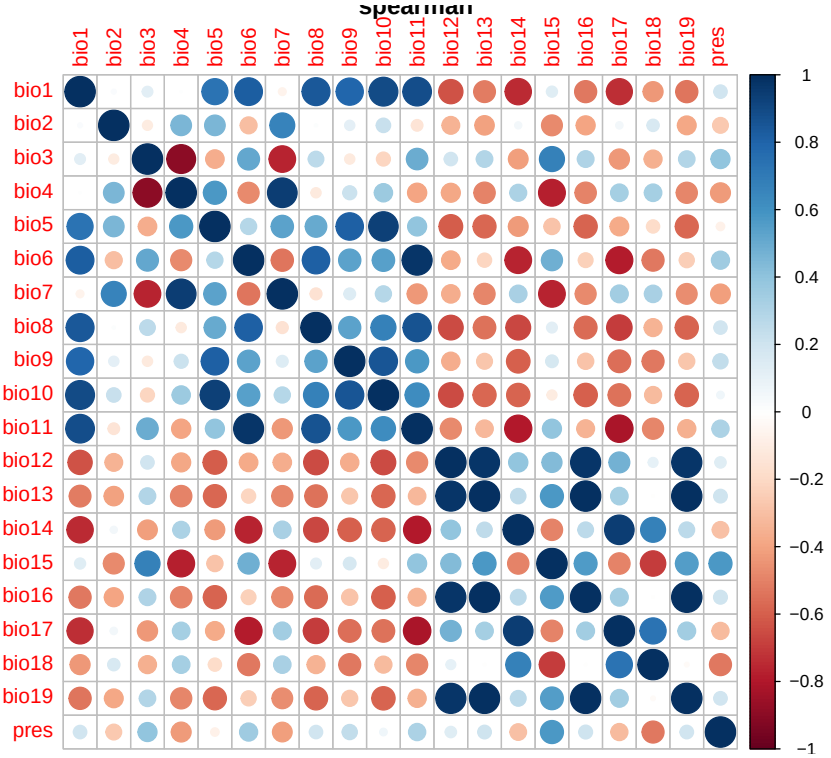
```
pairs(hammy, cex=0.4, pch=20, gap=0.3)
```




```
par(mfrow=c(1,1))
corrplotMIC <- corrplot(mine(hammy)$MIC, main="MIC")
```




```
corrplotSpear <- corrplot(cor(hammy, method="spearman", use="pairwise.complete.obs"), main="spearman")
```



In the corrplots, notice the strong correlations between several variable pairs. In the optimized models, it is unlikely that both of a highly correlated pair will be included because after including one, the second will contribute little to no new information to the model.

Mutual information variable selection (sensu Ivannikova et al.) with forward steps:

```
# corrplot(hammy.mutinfo/max(hammy.mutinfo)) # corrplot requires values between -1 and 1. I only did th

hammy.mutinfo <- mutinformation(discretize(hammy)) # bio1 has highest mutual info, i.e. strongest corre
hammy.mutinfo[,20]

#hammybc <- hammy[,-20]
#mutinformation(discretize(hammybc), discretize(hammy[,20])) # MI = 0.631 including all variables

# mutinformation(cbind(discretize(hammy[,1]), discretize(hammy[,2])), discretize(hammy[,20], nbins=2)) 
# this example is still using dummy-absences, though. Get "true"" pseudoabsences AND add more presences

# I can apply bootstraps to every step here... hmm then get confidence intervals on the MI of every add

# mutin1 accomplishes the same thing as just running mutinformation(discretize(hammy))
mutin1 <- rep(NA, 19)
for (i in 1:19){
  mutin1[i] <- mutinformation(discretize(hammy[,i]), discretize(hammy[,20]))
}
mutin1
#rank(mutin1) # bio1 by itself has most mutual information with presence, so bio1 becomes X1. MI = 0.23



mutin2 <- rep(NA, 19)
for (i in 1:19){
  mutin2[i] <- mutinformation(cbind(discretize(hammy[,1]), discretize(hammy[,i])), discretize(hammy[,20]
}

mutin2 # bio12 increases the information the most, so bio12 becomes X2. MI = 0.3576


## Then I suppose, compare the third most-informative variable (X3) determined from permuted medians to
## Here: permuted best X3 is bio2
mutin3 <- rep(NA, 19)
for(i in 1:19){
    mutin3[i] <- mutinformation(cbind(discretize(hammy[,1]), discretize(hammy[,12]), discretize(hammy[,
  }
mutin
rank(mutin) # Using all the points, the best X3 is bio2; model MI(3) = 0.49495: bio1, bio12, bio2

## Let's just continue using all the points for now...

### Determining X4:
```

```
mutin4 <- rep(NA, 19)
for(i in 1:19){
    mutin4[i] <- mutinformation(cbind(discretize(hammy[,1]),
                                      discretize(hammy[,12]),
                                      discretize(hammy[,2]),
                                      discretize(hammy[,i])),
                                discretize(hammy[,20]))
  }
mutin4
rank(mutin4) # bio17 is X4; MI = 0.5748


### Determining X5:
mutin5 <- rep(NA, 19)
for(i in 1:19){
    mutin5[i] <- mutinformation(cbind(discretize(hammy[,1]),
                                      discretize(hammy[,12]),
                                      discretize(hammy[,2]),
                                      discretize(hammy[,17]),
                                      discretize(hammy[,i])),
                                discretize(hammy[,20]))
  }
mutin5
rank(mutin5) # bio4 is X5; MI = 0.6030



mutin6 <- rep(NA, 19)
for(i in 1:19){
    mutin6[i] <- mutinformation(cbind(discretize(hammy[,1]),
                                      discretize(hammy[,12]),
                                      discretize(hammy[,2]),
                                      discretize(hammy[,17]),
                                      discretize(hammy[,4]),
                                      discretize(hammy[,i])),
                                discretize(hammy[,20]))
  }
mutin6
rank(mutin6) # bio15 and bio16 are tied for X6; MI = 0.6155... maybe bootstrap to pick one??

mutin6boot <- matrix(nrow=1000, ncol=19)
for(j in 1:1000) {
  hamboot <- rbind(hammy[sample(nrow(hammy[hammy$pres==1,]), 88, replace=F),],
                   hammy[sample(nrow(hammy[hammy$pres==0,]), 88, replace=F),])
  mutin <- rep(NA, 19)
  for(i in 1:19){
    mutin[i] <- mutinformation(cbind(discretize(hamboot[,1]),
                                      discretize(hamboot[,12]),
                                      discretize(hamboot[,2]),
                                      discretize(hamboot[,17]),
                                      discretize(hamboot[,4]),
                                      discretize(hamboot[,i])),
                                discretize(hamboot[,20]))
```

```
  }
  mutin6boot[j,] <- mutin
}
mutin6CI <- apply(mutin6boot, 2, medianCI)
mutin6CI # well this didn't help... bio10 has the highest median here... could this be more informative
rank(mutin6CI[1,]) # but then these results become incomparable to the other runs! fuuuuuck. So skip bo


mutin7 <- rep(NA, 19)
for(i in 1:19){
    mutin7[i] <- mutinformation(cbind(discretize(hammy[,1]),
                                      discretize(hammy[,12]),
                                      discretize(hammy[,2]),
                                      discretize(hammy[,17]),
                                      discretize(hammy[,4]),
                                      discretize(hammy[,15]),
                                      discretize(hammy[,i])),
                                discretize(hammy[,20]))
  }
mutin7
rank(mutin7) # bio16; MI = 0.626



mutin8 <- rep(NA, 19)
for(i in 1:19){
    mutin8[i] <- mutinformation(cbind(discretize(hammy[,1]),
                                      discretize(hammy[,12]),
                                      discretize(hammy[,2]),
                                      discretize(hammy[,17]),
                                      discretize(hammy[,4]),
                                      discretize(hammy[,15]),
                                      discretize(hammy[,16]),
                                      discretize(hammy[,i])),
                                discretize(hammy[,20]))
  }
mutin8
rank(mutin8) # tie: bio11 and bio18; MI = 0.6313. I'll go with bio18 because maxent suggests it is actu



mutin9 <- rep(NA, 19)
for(i in 1:19){
    mutin9[i] <- mutinformation(cbind(discretize(hammy[,1]),
                                      discretize(hammy[,12]),
                                      discretize(hammy[,2]),
                                      discretize(hammy[,17]),
                                      discretize(hammy[,4]),
                                      discretize(hammy[,15]),
                                      discretize(hammy[,16]),
                                      discretize(hammy[,18]),
                                      discretize(hammy[,i])),
                                discretize(hammy[,20]))
```

```
  }
mutin9
rank(mutin9) # no additional information... so the current model should be the best(?)/most parsimoniou

# best model using MI: pres ~ bio1, bio2, bio4, bio12, bio15, bio16, bio17, bio18
# This model has as much information as the model with all 19 variables!
```

Using mutual information, I identified eight predictors that have as much information as all nineteen! (MI = 0.631 eith these eight and remains at 0.631 with the addition of any other predictors)

The chosen predictors differ from those chosen by step(), interestingly (evaluated below).

GLM models:

```
hammy.glm <- glm(pres ~ bio1+bio2+bio3+bio4+bio5+bio6+bio7+bio8+bio9+bio10+bio11+bio12+bio13+bio14+bio1
summary(hammy.glm)
step(hammy.glm, direction="backward")

# using step() with glm, get lowest AIC with pres ~ bio1 + bio4 + bio9 + bio14 + bio15 + bio17 + bio18
hammy.glm.step <- glm(pres ~ bio1 + bio4 + bio9 + bio14 + bio15 + bio17 + bio18 + bio19, family=binomia
summary(hammy.glm.step)

# best model using MI: pres ~ bio1, bio2, bio4, bio12, bio15, bio16, bio17, bio18
# as an aside: the order you put variables into the glm with + doesn't matter...
hammy.glm.mi <- glm(pres ~ bio1 + bio12 + bio2 + bio17 + bio4 + bio15 + bio16 + bio18, family=binomial(
summary(hammy.glm.mi)

# are there still highly correlated variables in the two reduced models? Yes!
# mine(hammy[,c(1,2,4,12,15,16,17,18)])$MIC # yes: bio1+bio17 (MIC=0.710); bio12+bio16 (MIC=0.918). Try
# mine(hammy[,c(1,4,9,14,15,17,18,19)])$MIC # yes: bio1+bio17 again; bio14+bio17 (MIC=0.790). Try the s

# summary(glm(pres ~ bio1 + bio4 + bio9 + bio14 + bio15 + bio18 + bio19, family=binomial(link = "logit"
# summary(glm(pres ~ bio1 + bio12 + bio2 + bio17 + bio4 + bio15 + bio18, family=binomial(link = "logit"
```

Bootstrap functions:

These functions bootstrap resample from the presence points and from the absence points separately (so each iteration keeps the same number of presence and absence points), and the model is re-run with the resampled datapoints. The output is the map of the median probability of all bootstrapped predictions:

```
## bootstrapped model
```

```
bootPredict <- function(rasterbrick, model, present=spea2, absent=absent2, reps=10){ # will use spea2 a
  bootpreds <- replicate(reps, {
    pp <- sample(nrow(present), replace=T)
    aa <- sample(nrow(absent), replace=T)
    bootpres <- present[pp,]
    bootabsent <- absent[aa,]
    bootallpts <- rbind(bootpres, bootabsent) # combines bootstrapped present and absence points togeth
    newmodel <- update(model, data=bootallpts)
    newpredict <- predict(rasterbrick, newmodel, type="response")

  })
  medianpred <- calc(brick(bootpreds), fun=median)
  madpred <- calc(brick(bootpreds), fun=mad)
  #confintpred <- calc(brick(bootpreds), fun=quantile, probs=c(0.025, 0.975))
  invisible(stack(medianpred, madpred))
}

# bootstrapped NHST
bootPredict.nh <- function(rasterbrick, model, locs=hammy, reps=10){ # will use spea2 and absent2
  bootpreds.nh <- replicate(reps, {
    bootallpts <- apply(locs, 2, sample, replace=F) # shuffles data within columns for doing NHST
    bootallpts <- data.frame(bootallpts)
    #colnames(bootallpts) <- colnames(locs)
    newmodel <- update(model, data=bootallpts)
    newpredict <- predict(rasterbrick, newmodel, type="response")

  })
  medianpred.nh <- calc(brick(bootpreds.nh), fun=median)
  madpred.nh <- calc(brick(bootpreds.nh), fun=mad)
  #confintpred <- calc(brick(bootpreds), fun=quantile, probs=c(0.025, 0.975))
  invisible(stack(medianpred.nh, madpred.nh))
}
```

Model predictions:

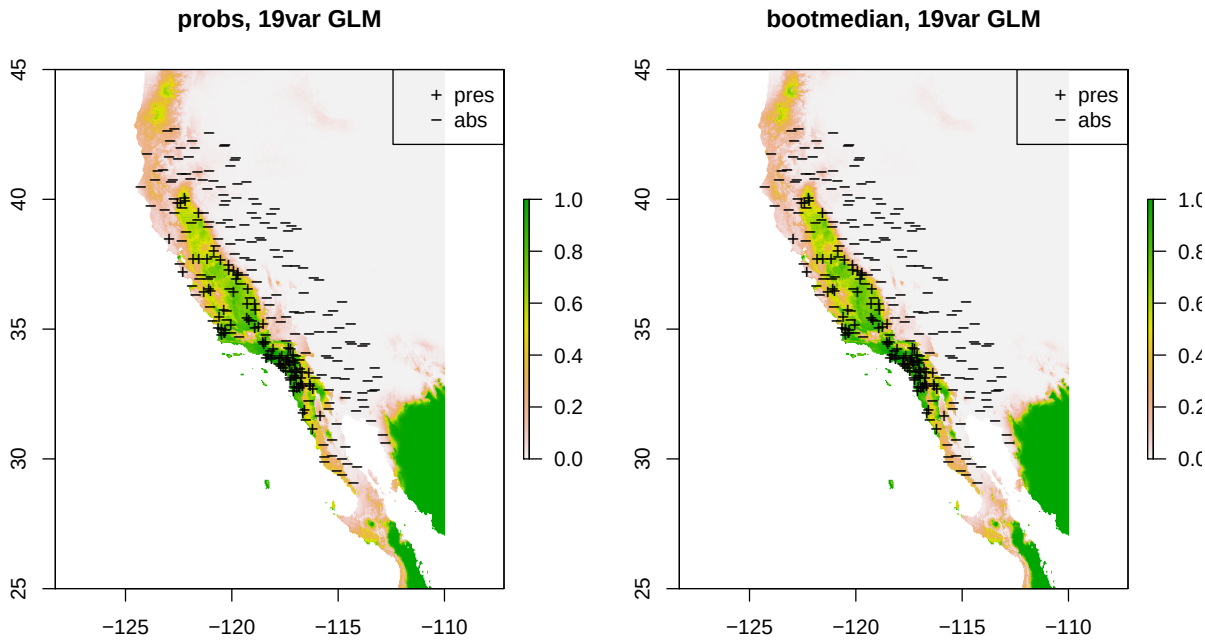
the package "raster" allows for explicit spatial prediction using the values in the variable rasters/maps

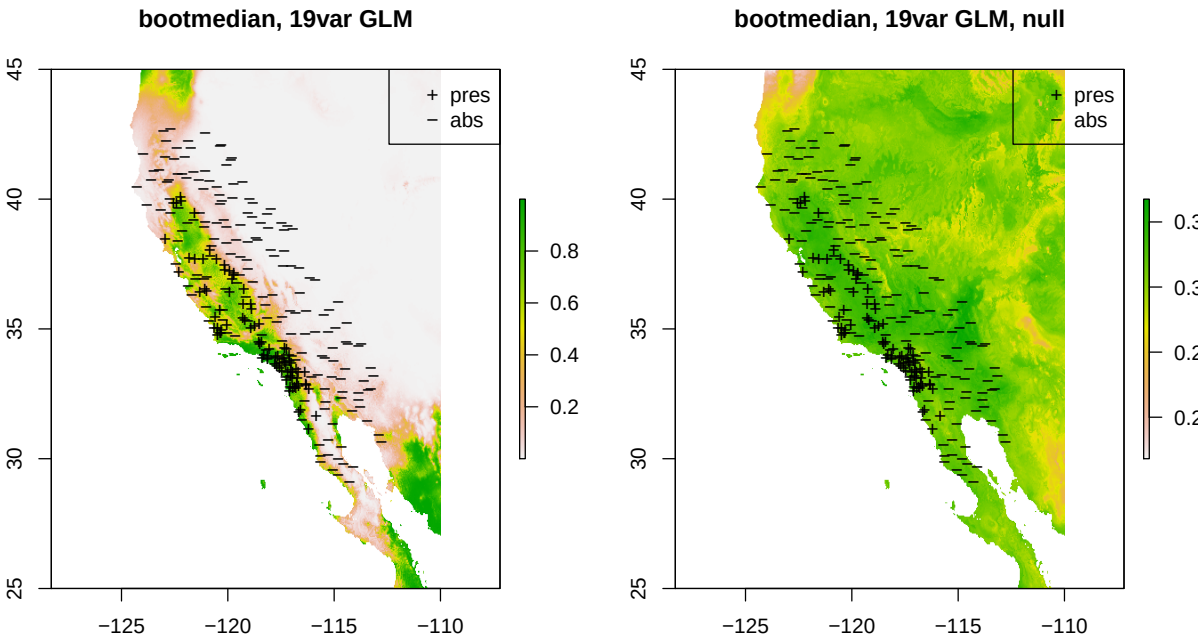
```
# ?predict (use it as implemented in raster package, not dismo)
# bootstrap the predictions and plot the median

bclimRaster <- brick("C:/Users/Kevin/Google Drive/UCLA Courses or Lab meetings etc/EEB 234/Final Projec

bclimRaster.mi <- bclimRaster[[c(1,2,4,12,15,16,17,18)]] #bio1, bio2, bio4, bio12, bio15, bio16, bio17,
bclimRaster.glm.step <- bclimRaster[[c(1,4,9,14,15,17,18,19)]] #bio1 + bio4 + bio9 + bio14 + bio15 + bi

hammyPredict.glm.step <- predict(bclimRaster.glm.step, hammy.glm.step, type="response") # type="respons
hammyPredict.mi <- predict(bclimRaster.mi, hammy.glm.mi, type="response")
hammyPredict.full <- predict(bclimRaster, hammy.glm, type="response")
```

```
# bootstrap the models and take the median over all predictions. Print MAD as well.
hammyPredict.full.boot <- bootPredict(bclimRaster, hammy.glm, reps=100)
hammyPredict.full.nh <- bootPredict.nh(bclimRaster, hammy.glm, reps=100)

hammyPredict.glm.step.boot <- bootPredict(bclimRaster.glm.step, hammy.glm.step, reps=100)
hammyPredict.glm.step.nh <- bootPredict.nh(bclimRaster.glm.step, hammy.glm.step, reps=100)

hammyPredict.mi.boot <- bootPredict(bclimRaster.mi, hammy.glm.mi, reps=100)
hammyPredict.mi.nh <- bootPredict.nh(bclimRaster.mi, hammy.glm.mi, reps=100)
```

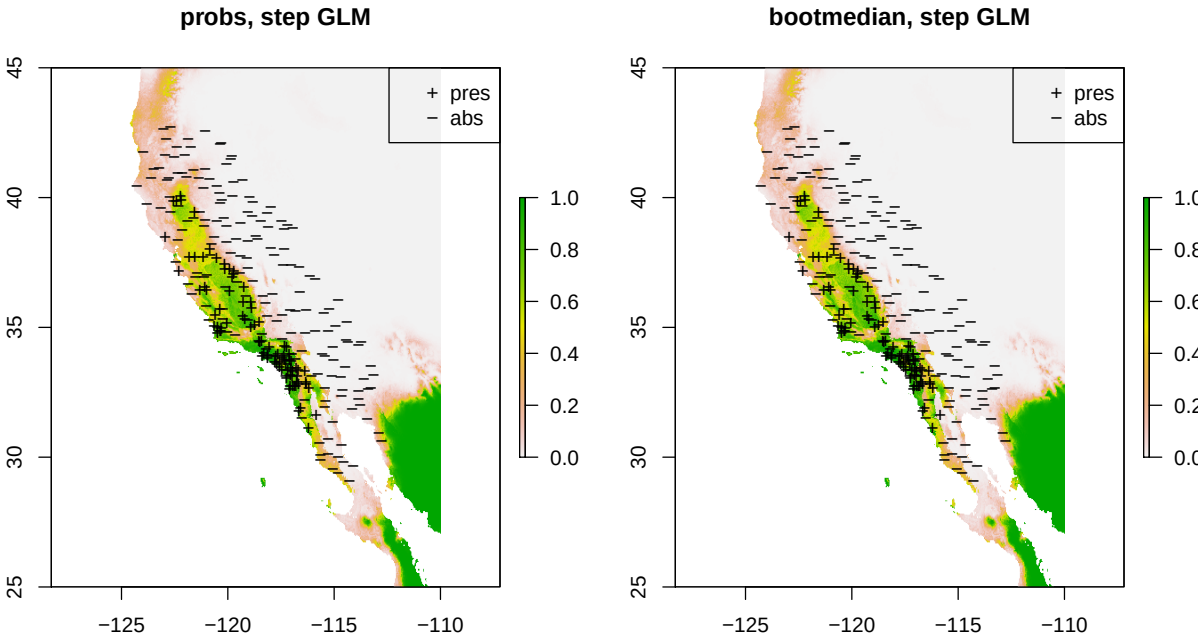
The following maps show probability of occurrence projected across the study area. Because absences/pseudoabsences were not sampled from the entire study area, the model often stumbles in areas of novel climate (see e.g. lower right in all maps).

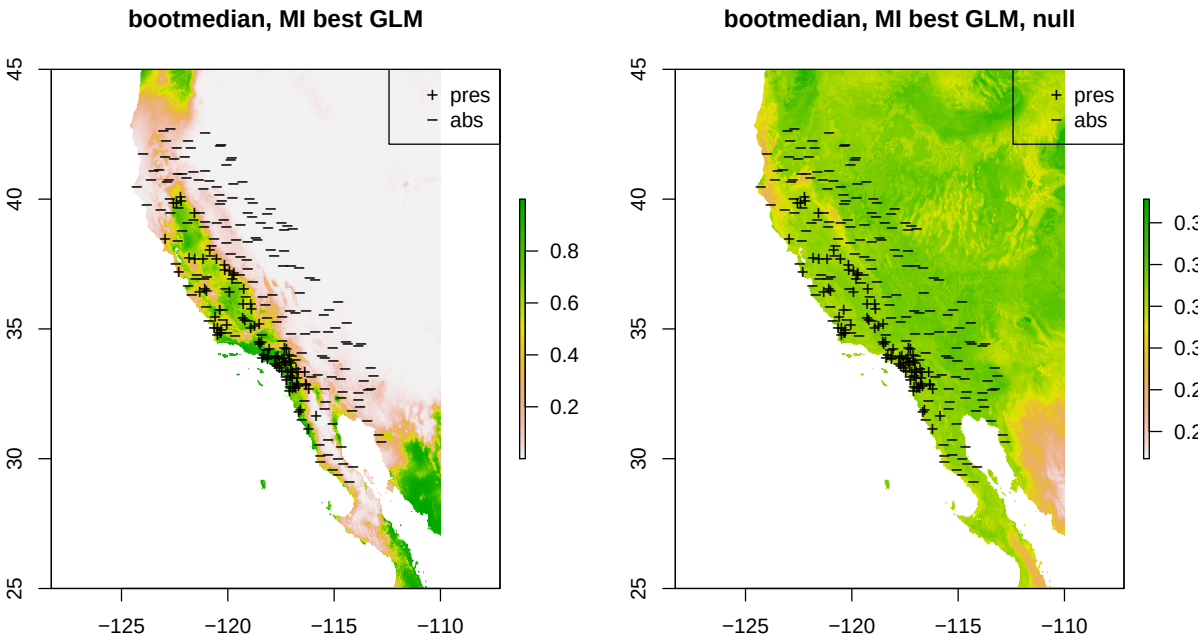
GLM model with all 19 predictors:



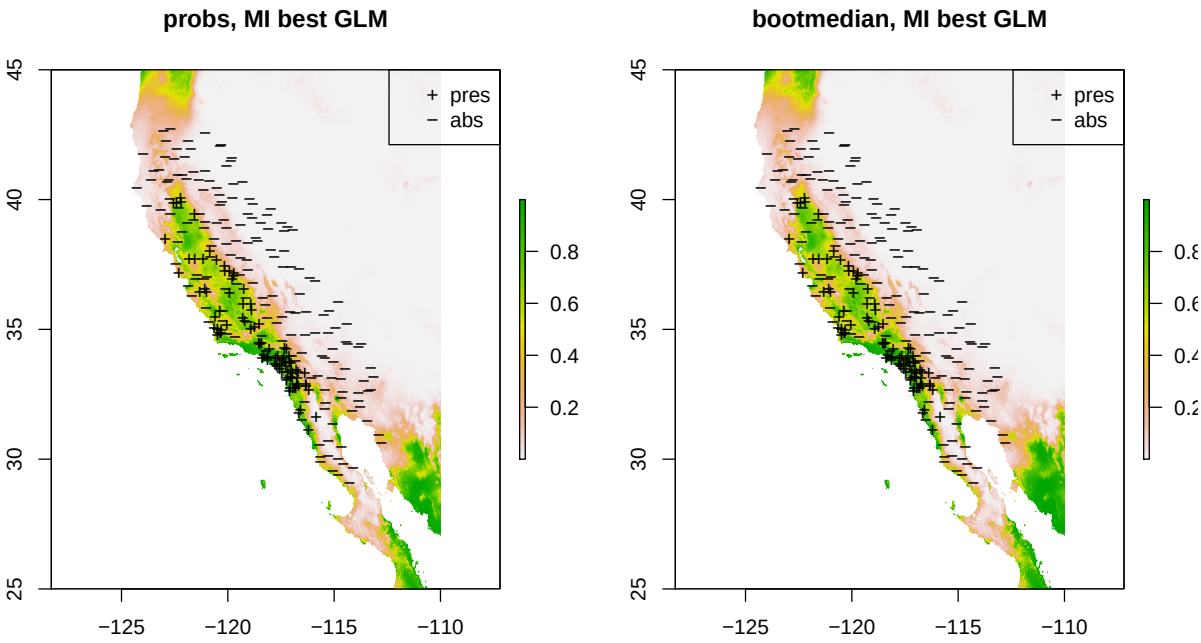


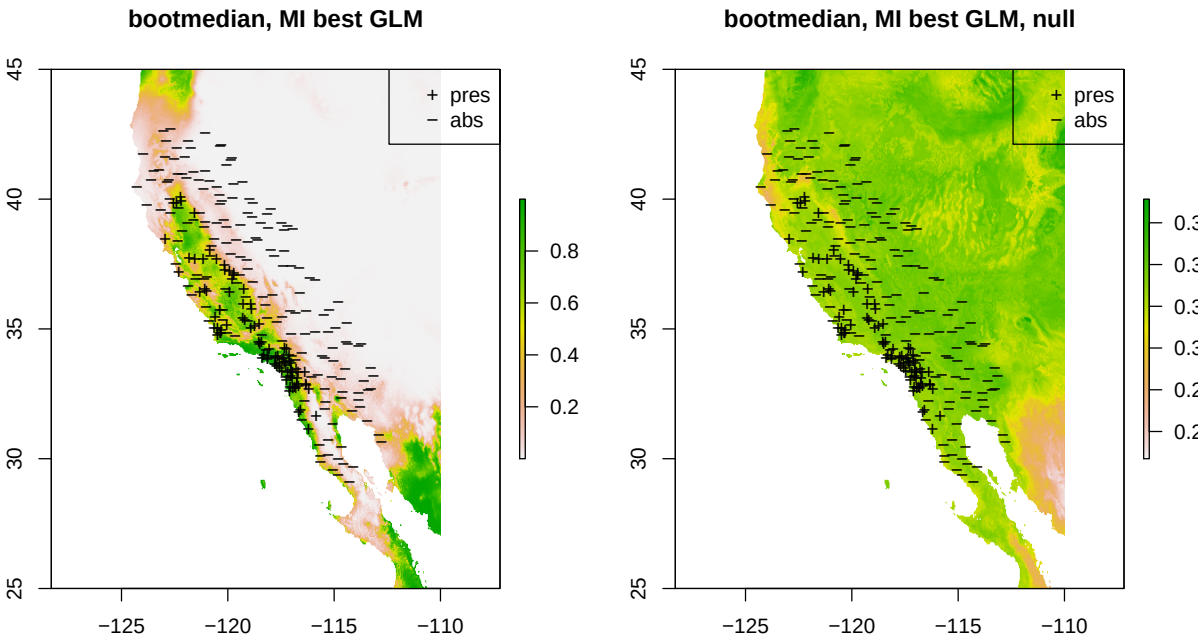
GLM model with stepwise-determined predictors:



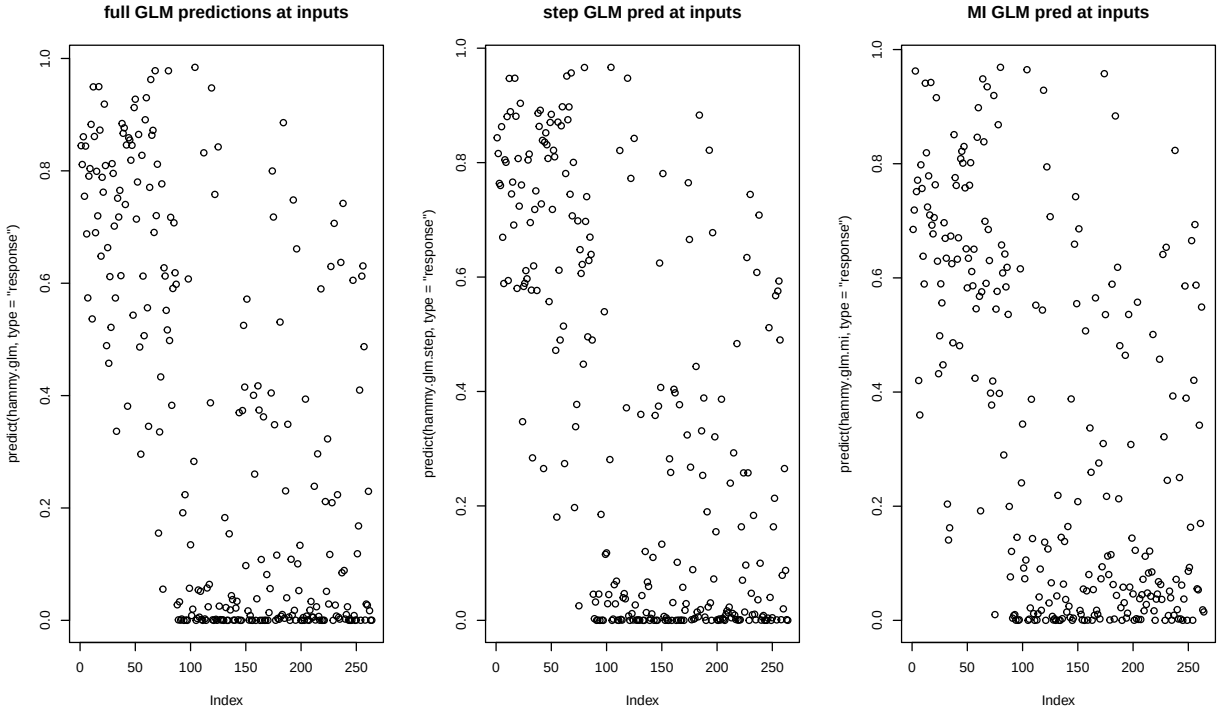


GLM model with MI-selected predictors:

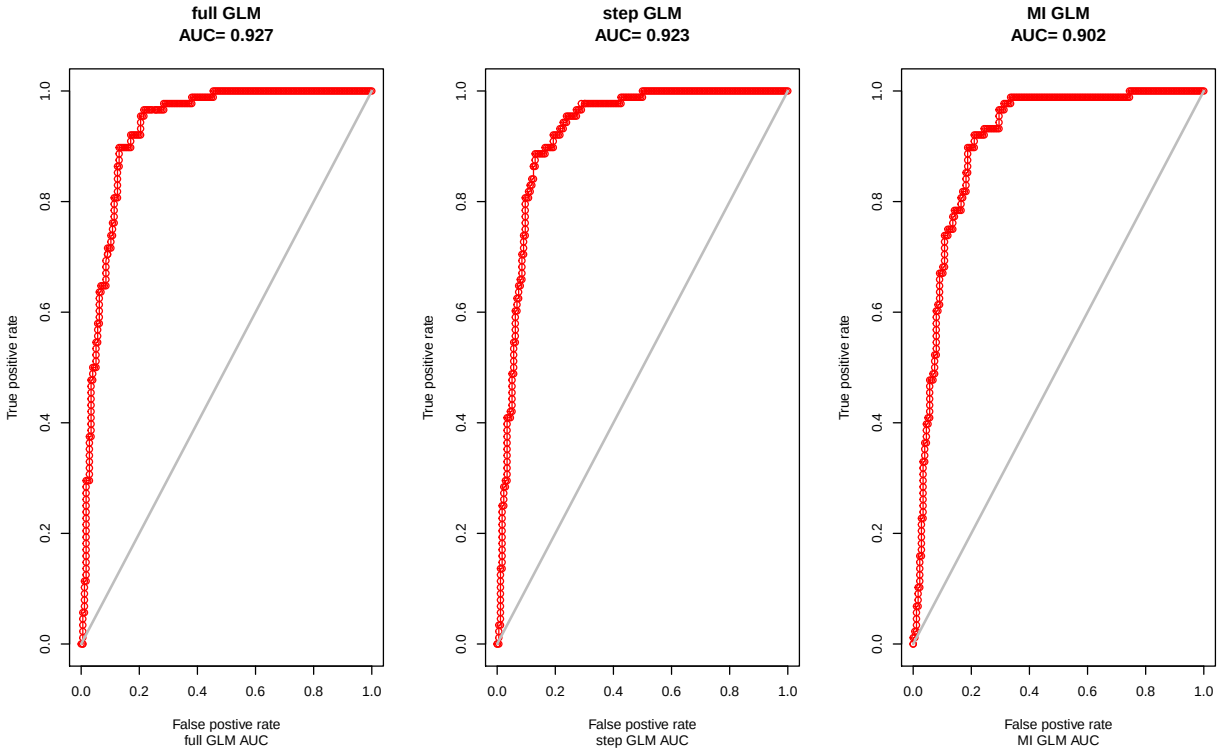



bootmedian, MI best GLM
+ pres
− abs
0.8
0.6
0.4
0.2
45
40
35
30
25
−125
−120
−115
−110
bootmedian, MI best GLM, null
+ pres
− abs
0.3
0.3
0.3
0.3
0.2
0.2
45
40
35
30
25
−125
−120
−115
−110

Model probabilities of occurrence at the data used to generate the model itself. Index 1-88 are true presences (actual prob=1) and 89+ are pseudoabsences (actual prob=0):



Area under ROC curve for all three models:



Receiver operating characteristic curves of the three models examined. The area under these curves is assumed to be a measure of model performance, with higher AUC indicating a better model. AUC are only comparable across models using same presence and absence points and the same areal extent. With the highest AUC of 0.927, the GLM model with all 19 bioclim variables performs the best, though this may come at a cost of generalizability to novel climate spaces.
