Exercise 8: Bias Correction

Adam B. Smith, Danielle Svehla, & Camilo Sanín

July 24, 2017

Biased data produces biased model. Records can display bias because they represent the haphazard collation of collectors who were conducting their research for their--not your--purposes. As a result, specimens tend to be located near roads, universities, and cities. This geographic bias can cause records to be biased in environmental space where the model actually operates. Likewise, collectors might bias collections towards environments that are nice to work in or offer a higher chance of supporting their species of interest. This is a direct form of bias that also affects model output.

In this exercise you will learn:

1. How to **inspect** for potential bias in geographic and environmental space.
2. Four methods for correcting for bias in sampling of presence sites. These include

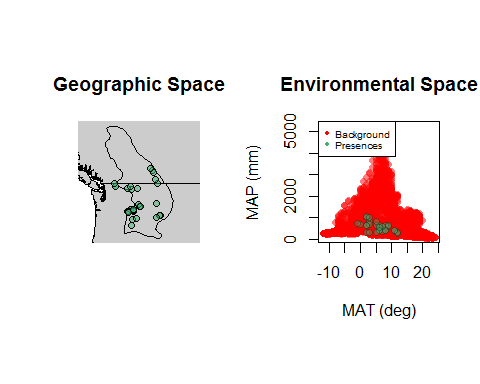
* Using **target background sites** in place of random background sites
* Using a **bias grid** that reflects probability of sampling to generate background sites
* **Geographic thinning** of presence sites
* **Environmental thinning** of presence sites
* How to implement **inverse P-weighting** to correct for bias
* Issues related to inclusion of presence sites in which the species inhabited a **microsite** favorable to it in an otherwise unfavorable environment

Personally we have found that different methods work differently for different situations, so we can't recommend any one in general.

# Exploring bias

Let's look at the distribution of records in geographic and environmental space.

par(mfrow=c(1, 2), pty='s')  
  
# plot in geographic space  
plot(rangeMap, main='Geographic Space')  
plot(countries, add=TRUE, col='gray80')  
plot(rangeMap, add=TRUE)  
points(records$longitude,  
 records$latitude,  
 pch=21,  
 bg=alpha('mediumseagreen', 0.5)  
)  
  
# plot in environmental space  
plot(randomBg$WC01,  
 randomBg$WC12,  
 pch=16,  
 col=alpha('red', 0.5),  
 xlab='MAT (deg)',  
 ylab='MAP (mm)',  
 main='Environmental Space'  
)  
  
points(records$WC01,  
 records$WC12,  
 col=alpha('black', 0.35),  
 bg=alpha('mediumseagreen', 0.35),  
 pch=21  
)  
  
legend('topleft',  
 legend=c('Background', 'Presences'),  
 col=c('red', 'mediumseagreen'),  
 pch=16,  
 cex=0.6  
)



Where might the model become bias in *geographic* space? How would this affect the model's ability to capture the species' response to our chosen variables (WC03, WC10, WC15, and WC19--see the [WORLDCLIM](http://www.worldclim.org/bioclim) site for definitions)? Where might the model become biased in *environmental* space? What portions of environmental space will be over-represented and what portions under-represented?

Obviosuly we seem to lacking records from the Canadian portion of the range. This presents an extreme version of bias--it appears the species resides there, but we have very little data to model its response in the area lacking records. Hoenetly, in this case we would suggest 1) looking for more records; and if that didn't help; 2) just interpreting the model output for the US-portion of the range; or 3) giving up! But we're here to learn regardless of how well our models work, so we'll forge ahead.

# Bias correction using target background sites

So far we've used randomly located background sites to train Maxent. However, this presumes that presence sites were sampled such that each real presence had an equal probability of being sampled--i.e., in an unbiased manner. One way to correct for bias in presences is to include the same kind of bias in background sites. The two biases then cancel one another out.

Most relevant citation: [Phillips, S.J., Dud?k, M., Elith, J., Graham, C.H., Lehmann, A., Leathwick, J., and Ferrier, S. 2009. Sample selection bias and presence-only distribution models: Implications for background and pseudo-absence data. Ecological Applications 19:181-197.](http://dx.doi.org/10.1890/07-2153.1)

Phillips et al. (2009) propose using **target background** sites, which are presences of other species that could have been recorded give the methods used to observe the focal species. We'll download these sites from GBIF. In our case we'll use presence sites of other ground squirrels in western North America. This includes genera from *Urocitellus* plus *Ammospermophilus*, *Callospermophilus*, *Poliocitellus*, *Xerospermophilus*, *Ictidomys*, and *Otospermophilus*. In this case we'll use a cheat--in the recent past all of these used to be simply *Spermophilus*. Searching for GBIF automatically returns synonyms so long as the synonyms are part of the GBIF backbone taxonomy. So we'll search for "*Spermophilus*" and capture all of these genera. Normally we would also want to subject these sites to the same data-cleaning procedures we used for our focal species, but here we'll just use them as-is. We'll also use the rgbif package to download them.

Note that the download process can take quite a few minutes. To speed the process I suggest using the records I pre-downloaded in the Background Sites folder. I've included the code below to get these but feel free to skip it and go to the next chunk of code.

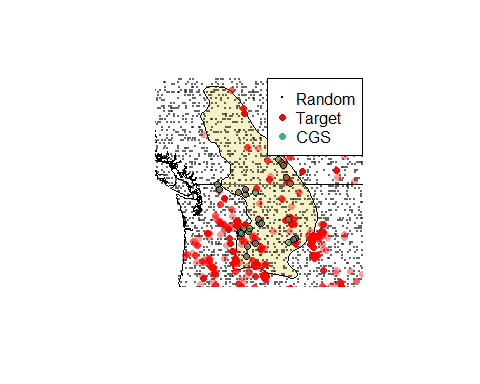
library(rgbif)  
  
# obtain GBIF data for other ground squirrels  
targetRecords <- occ\_search(  
 scientificName=c(  
 'Urocitellus',  
 'Ammospermophilus',  
 'Callospermophilus',  
 'Poliocitellus',  
 'Xerospermophilus',  
 'Ictidomys',  
 'Otospermophilus'  
 ),  
 hasCoordinate=TRUE,  
 basisOfRecord='PRESERVED\_SPECIMEN',  
 year='1970,2000',  
 hasGeospatialIssue=FALSE,  
 fields='minimal',  
 decimalLongitude='-157,-94',  
 decimalLatitude='30,69',  
 limit=10000  
)  
  
# cycle through each genus and get just coordinates  
if (exists('targetSites')) rm(targetSites)  
  
for (i in seq\_along(targetRecords)) {  
  
 # get this genus' coordinates  
 thisGenusCoords <- data.frame(  
 longitude=targetRecords[[i]]$data$decimalLongitude,  
 latitude=targetRecords[[i]]$data$decimalLatitude  
 )  
   
 # add to data frame with all target coordinates  
 targetSites <- if (exists('targetSites')) {  
   
 rbind(targetSites, thisGenusCoords)  
   
 } else {  
   
 thisGenusCoords  
   
 }  
   
}  
  
head(targetSites)  
  
# extract environmental data at target background sites  
targetEnv <- as.data.frame(extract(climate, targetSites))  
  
# remove target background sites with no environmental data (fall in ocean)  
outside <- which(is.na(rowSums(targetEnv))) # if a record has missing data (NA) then the sum of its rows is also NA... find these (if any)  
if (length(outside) > 0) {  
 targetSites <- targetSites[-outside, ]  
 targetEnv <- targetEnv[-outside, ]  
}  
  
# combine coordinates and environmental data  
targetBg <- cbind(targetSites, targetEnv)  
names(targetBg)[1:2] <- c('longitude', 'latitude')  
  
# how many target records do we have?  
# if we had >10000 we could use a random sample of 10000 for expediency  
nrow(targetBg)  
  
# save target background sites  
save(targetBg, file='./Background Sites/Target Background Sites Drawn from All Spermophilus Downloaded from GBIF.Rdata', compress=TRUE)

**Restart here if you did not execute the previous chunk of code.**

load('./Background Sites/Target Background Sites Drawn from All Spermophilus Downloaded from GBIF.Rdata')

Let's compare the random versus target background in geographic space.

# plot background sites on map  
par(pty='s')  
plot(rangeMap)  
sp::plot(countries, col='white', add=TRUE)  
plot(rangeMap, add=TRUE, col=alpha('khaki', 0.5))  
  
# random background  
points(randomBg$longitude,  
 randomBg$latitude,  
 col=alpha('black', 0.6),  
 pch=16,  
 cex=0.1  
)  
  
# target background  
points(targetBg$longitude,  
 targetBg$latitude,  
 col=alpha('red', 0.4),  
 pch=16  
)  
  
# presences  
points(records$longitude,  
 records$latitude,  
 pch=21,  
 bg=alpha('mediumseagreen', 0.5)  
)  
  
legend('topright',  
 legend=c('Random', 'Target', 'CGS'),  
 col=c('black', 'red', 'mediumseagreen'),  
 pch=c(16, 16, 16),  
 pt.cex=c(0.3, 1, 1),  
 bg='white'  
)

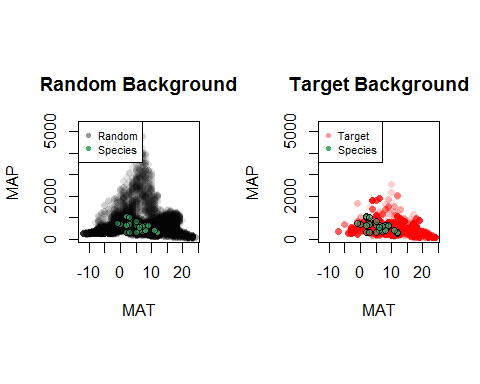


## Reflection

1. Where does most collection activity tend to occur? Does it seem to reflect the same bias we think we see in the ground squirrel's records?
2. Can you identify any geographic entities (cities, roads) just by looking at the target sites?

Now let's examine the distribution of target background sites in environmental space.

par(mfrow=c(1, 2), pty='s')  
  
# plot RANDOM background sites in environmental space  
plot(randomBg$WC01,  
 randomBg$WC12,  
 pch=16,  
 col=alpha('black', 0.1),  
 main='Random Background',  
 xlab='MAT',  
 ylab='MAP',  
 xlim=c(min(randomBg$WC01), max(randomBg$WC01)),  
 ylim=c(min(randomBg$WC12), max(randomBg$WC12))  
)  
  
points(records$WC01,  
 records$WC12,  
 pch=21,  
 bg=alpha('mediumseagreen', 0.5)  
)  
  
legend('topleft',  
 legend=c('Random', 'Species'),  
 pch=16,  
 col=c(alpha('black', 0.4), 'mediumseagreen'),  
 bg='white',  
 cex=0.7  
)  
  
# plot TARGET background sites in environmental space  
plot(targetBg$WC01,  
 targetBg$WC12,  
 pch=16,  
 col=alpha('red', 0.1),  
 main='Target Background',  
 xlab='MAT',  
 ylab='MAP',  
 xlim=c(min(randomBg$WC01), max(randomBg$WC01)),  
 ylim=c(min(randomBg$WC12), max(randomBg$WC12))  
)  
  
points(records$WC01,  
 records$WC12,  
 col=alpha('black', 0.4)  
)  
  
points(records$WC01,  
 records$WC12,  
 bg=alpha('mediumseagreen', 0.5),  
 pch=21  
)  
  
legend('topleft',  
 legend=c('Target', 'Species'),  
 pch=16,  
 col=c(alpha('red', 0.4), 'mediumseagreen'),  
 bg='white',  
 cex=0.7  
)

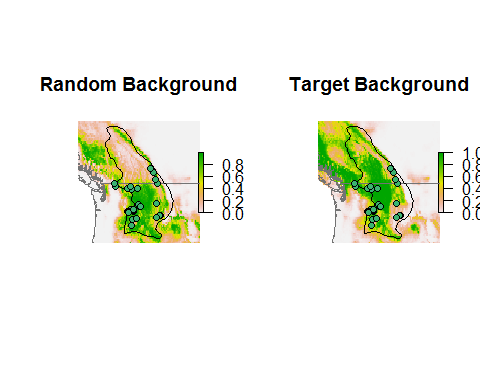


## Reflection

What kinds of climate space are more/less represented in the target background set than the random background set? Knowing how Maxent performs its calculations (previous tutorial), how do you think this will affect the predictions into these environments (random environment vs. target environment)?

Now let's train a model using the target background sites and write the prediction raster.

# make training data frame with predictors and vector of 1/0 for presence/background  
trainData <- rbind(records[ , predictors], targetBg[ , predictors])  
presBg <- c(rep(1, nrow(records)), rep(0, nrow(targetBg)))  
  
# create output directory for model object and rasters  
dir.create('./Models/Model 03 Bias Correction - Target Background',  
 recursive=TRUE, showWarnings=FALSE)  
  
# model species  
targetBgModel <- maxnet(p=presBg, data=trainData)  
  
# save model  
save(targetBgModel,  
 file='./Models/Model 03 Bias Correction - Target Background/Model.Rdata',  
 compress=TRUE)  
  
# write raster  
targetBgMap <- predict(  
 climate[[predictors]],  
 targetBgModel,  
 filename='./Models/Model 03 Bias Correction - Target Background/maxentPrediction1970to2000',   
 format='GTiff',  
 overwrite=TRUE,  
 type='cloglog'  
)  
  
# plot  
par(mfrow=c(1, 2), pty='s')  
plot(rangeMap, main='Random Background')  
plot(manualSelectMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
plot(rangeMap, main='Target Background')  
plot(targetBgMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')



Using the target sites increased the estimated suitability in the northern part of the species' range where we have few records.

## Reflection

Compare your efforts to correct for bias using target background sites.

1. Where do the species' records seem biased in geographic and environmental space? What might be the cause of this bias?
2. Do you think the target background approach was able to correct for the bias?
3. What other effects did using the target background sites have on the model prediction? Were these desirable?

# OPTIONAL: The next three sections present other methods for correcting for bias. If you are making good headway or feel you want to "know it all, now!", then please proceed. Otherwise please proceed to Exercise 9.

# Bias background bias correction

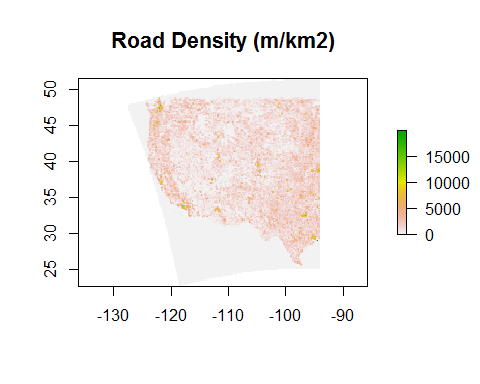
Related to the target background technique is use of a **bias map** from which background sites are drawn. The background sites are drawn with a probability proportional to the cells' values. The values are generated in a manner that presumably reflects the sampling bias in the presence sites. How can you generate a bias map? It depends on the species--often bias maps reflect factors expected to correlate with the probability that an area was surveyed for the species. For example, cell values might reflect distance to roads, human population density, etc.

Very relevant citations: [Elith, J., M. Kearney, and S. Phillips. 2010. The art of modeling range-shifting species. Methods in Ecology and Evolution 1:330-342.](http://dx.doi.org/10.1111/j.2041-210X.2010.00036.x)

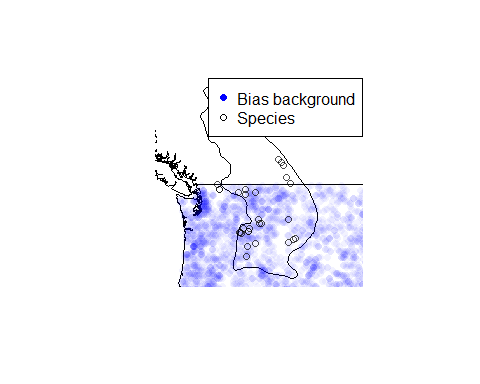
[Meyer, C., Kreft, H., Guralnick, and Jetz, W. 2015. Global priorities for an effective information basis of biodiversity distributions. Nature Communications 6:8221.](http://dx.doi.org/10.1038/ncomms9221)

Here we'll use a road density map made by NOAA with the expectation that survey effort is a function of vehicular access. The map represents length of roads per unit area. Unfortunately the map was only created for the coterminous United States, so our model should only be interpreted within this country.

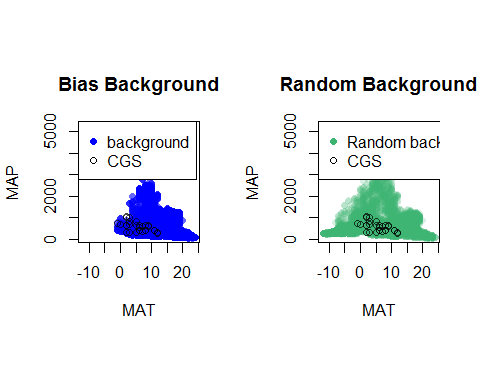
# road density raster  
roadDensity <- raster('./Road Density 1998 (NOAA)/roadDensity.tif')  
plot(roadDensity, main='Road Density (m/km2)', cex.main=1.4)



# Sample background points in proportion to road density.  
# Using the sampleRast() function in the enmSdmpackage  
# because randomPoints() function fails if all  
# cells have been selected... this function samples with replacement.  
biasSites <- sampleRast(roadDensity, 11000, replace=TRUE, prob=TRUE)  
  
# extract environmental data at bias sites  
biasEnv <- extract(climate, biasSites)  
biasEnv <- as.data.frame(biasEnv)  
  
# remove target background sites with no environmental data (fall in ocean)  
outside <- which(is.na(rowSums(biasEnv)))  
  
if (length(outside) > 0) {  
  
 biasSites <- biasSites[-outside, ]  
 biasEnv <- biasEnv[-outside, ]  
  
}  
  
# combine coordinates and environmental data  
biasBg <- cbind(biasSites, biasEnv)  
names(biasBg)[1:2] <- c('longitude', 'latitude')  
  
# save target background sites  
save(biasBg,  
 file='./Background Sites/Bias Backgound Sites Drawn from Road Density.Rdata',  
 compress=TRUE  
)  
  
# map bias-selected background sites  
par(pty='s')  
plot(rangeMap)  
sp::plot(countries, add=TRUE)  
  
points(biasSites, col=alpha('blue', 0.05), pch=16)  
points(records$longitude,  
 records$latitude,  
 pch=1,  
 col=alpha('black', 0.7)  
)  
  
legend('topright',  
 xpd=NA,  
 legend=c('Bias background', 'Species'),  
 col=c('blue', 'black'),  
 pch=c(16, 1),  
 bg='white'  
)



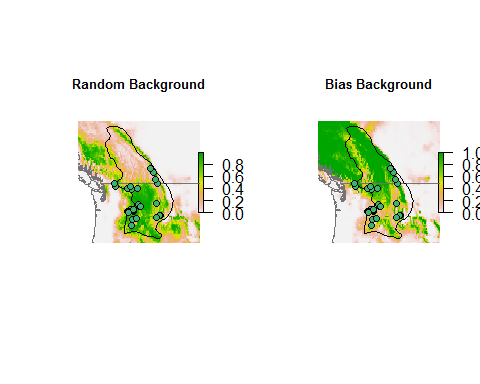
# plot BIAS background sites in environmental space  
par(mfrow=c(1, 2), pty='s')  
plot(biasBg$WC01,  
 biasBg$WC12,  
 pch=16,  
 col=alpha('blue', 0.6),  
 main='Bias Background', xlab='MAT', ylab='MAP',  
 xlim=c(min(randomBg$WC01), max(randomBg$WC01)),  
 ylim=c(min(randomBg$WC12), max(randomBg$WC12))  
)  
  
points(records$WC01,  
 records$WC12,  
 col=alpha('black', 0.7)  
)  
  
legend('topleft',  
 legend=c('background', 'CGS'),  
 pch=c(16, 1),  
 col=c('blue', 'black'),  
 bg='white'  
)  
  
# plot RANDOM background sites in environmental space  
plot(randomBg$WC01,  
 randomBg$WC12,  
 pch=16,  
 col=alpha('mediumseagreen', 0.2),  
 main='Random Background',  
 xlab='MAT',  
 ylab='MAP',  
 xlim=c(min(randomBg$WC01), max(randomBg$WC01)),  
 ylim=c(min(randomBg$WC12), max(randomBg$WC12))  
)  
  
points(records$WC01,  
 records$WC12,  
 col=alpha('black', 0.7)  
)  
  
legend('topleft',  
 legend=c('Random background', 'CGS'),  
 pch=c(16, 1),  
 col=c('mediumseagreen', 'black'),  
 bg='white'  
)



How do the bias background sites compare to the random background sites in environmental space?

Continue modeling using the background sites drawn from the bias (road density) grid:

# make training data frame with predictors and vector of 1/0 for presence/background  
trainData <- rbind(  
 records[ , predictors],  
 biasBg[ , predictors]  
)  
  
presBg <- c(rep(1, nrow(records)), rep(0, nrow(biasBg)))  
  
# create output directory for model object and rasters  
dir.create('./Models/Model 04 Bias Correction - Bias Background', recursive=TRUE,  
 showWarnings=FALSE)  
  
# model species  
biasBgModel <- maxnet(p=presBg, data=trainData)  
  
# save model  
save(biasBgModel,  
 file='./Models/Model 04 Bias Correction - Bias Background/Model.Rdata',  
 compress=TRUE  
)  
  
# write raster  
biasBgMap <- predict(  
 climate[[predictors]],  
 biasBgModel,  
 filename='./Models/Model 04 Bias Correction - Bias Background/maxentPrediction1970to2000',  
 format='GTiff',  
 overwrite=TRUE,  
 type='cloglog'  
)  
  
# plot  
par(mfrow=c(1, 2), pty='s', cex.main=0.8)  
  
plot(rangeMap, main='Random Background')  
plot(manualSelectMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
plot(rangeMap, main='Bias Background')  
plot(biasBgMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')



Interestingly, using the bias map seems to have *over*-corrected for the undersampling of Canadian sites.... but recall that the bias map only covers the US, so predictions outside this area should be ignored. (Still, it's a useful check on your understanding to think about why this occurs.) Within the US estimated environmental suitability seems to be sparser within the range.

### Reflection

1. How do the target background and bias background approaches compare? How do they compare to the model using random background sites? Overall which seems better?
2. What other types of data could you use to reflect bias in sampling?

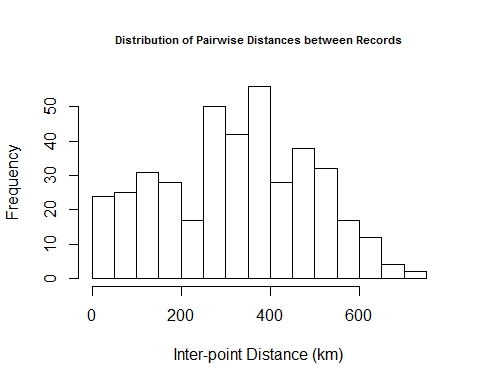
# Geographic thinning of records

Sometimes the only way to remove aggregations of records is to discard some. Records can be thinned in geographic or environmental space. In this portion of the exercise we'll thin records in geographic space.

Very relevant citation: [Aiello-Lammens, M. A., Boria, R. A., Radosavljevic, A., Vilela, B., and Anderson, R. P. 2015. spThin: an R package for spatial thinning of species occurrence records for use in ecological niche models. Ecography 38:541-545 (ver. X).](http://dx.doi.org/10.1111/ecog.01132)

The general idea behind geographic thinning is that if records are far enough from one another on Earth they'll also be more representative in environmental space. But how far apart must they be? We'll manually choose a distance based on the distribution of pairwise distances between records.

# calculate pairwise distances between points... store as matrix  
dists <- matrix(rep(NA, nrow(records)^2), nrow=nrow(records))  
  
for (i in 1:nrow(records)) {  
 dists[i, ] <- distCosine(  
 cbind(records$longitude[i], records$latitude[i]),  
 cbind(records$longitude, records$latitude)  
 )  
}  
  
# force distance from a point to itself to NA  
diag(dists) <- NA  
  
# remove duplicate distances  
dists[lower.tri(dists)] <- NA  
  
hist(dists / 1000,  
 breaks=20,  
 xlab='Inter-point Distance (km)',  
 main='Distribution of Pairwise Distances between Records',  
 cex.main=0.7  
)



Notice that the number of distances initially increases, peaks ~250 km, then declines. Normally we'd stratify by thinning points so they're no closer than half this distance (~125 km). Initially we did indeed try this but found that it removed so many points that the predictions were *much* worse... since our goal is to demonstrate removal of bias (and presumably production of better models), let's use a smaller distance (50 km) so we have more points remaining.

We can thin the records using a script we wrote inspired by the spThin package. The function randomly removes points based on minimum distances, so your and my results may vary.

# thinning distance in meters  
minDist <- 50000  
  
set.seed(12345)  
  
# thin records  
spatiallyThinnedRecords <- geoThinApprox(  
 x=records,  
 minDist=minDist,  
 longLat=c('longitude', 'latitude')  
)  
  
# save records!  
saveRDS(spatiallyThinnedRecords, './Species Records/06 Species Records - Randomly Thinned Presences.rds')  
  
nrow(records)

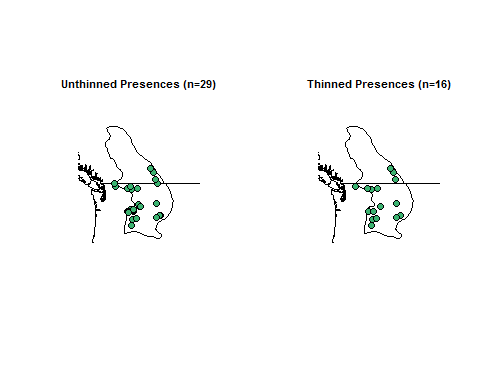
## [1] 29

nrow(spatiallyThinnedRecords)

## [1] 16

How many records did thinning remove?

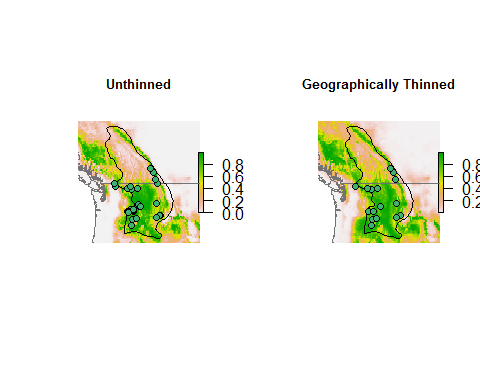
# plot  
par(mfrow=c(1, 2), pty='s', cex.main=0.7)  
plot(rangeMap, main=paste0('Unthinned Presences (n=', nrow(records), ')'))  
sp::plot(countries, add=TRUE, col='white')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
plot(rangeMap, main=paste0('Thinned Presences (n=', nrow(spatiallyThinnedRecords), ')'))  
sp::plot(countries, add=TRUE, col='white')  
plot(rangeMap, add=TRUE)  
points(spatiallyThinnedRecords$longitude, spatiallyThinnedRecords$latitude, pch=21, bg='mediumseagreen')



After thinning we're left with just a handful of records... Nonetheless, let's proceed on the assumption that the thinned data best reflects the overall unbiased environmental preferences of the Columbian ground squirrel.

Now, let's model. Note that we'll be using the random background sites (i.e., not the target sites or sites from the bias grid) since the presences have presumably had their bias removed.

# make training data frame with predictors and vector of 1/0 for presence/background  
trainData <- rbind(  
 spatiallyThinnedRecords[ , predictors],  
 randomBg[ , predictors]  
)  
  
presBg <- c(rep(1, nrow(spatiallyThinnedRecords)), rep(0, nrow(randomBg)))  
  
# create output directory for model object and rasters  
dir.create('./Models/Model 05 Bias Correction - Spatially Thinned Presences',  
 recursive=TRUE, showWarnings=FALSE)  
  
# model species  
spatialThinModel <- maxnet(p=presBg, data=trainData)  
  
# save model  
save(spatialThinModel,  
 file='./Models/Model 05 Bias Correction - Spatially Thinned Presences/Model.Rdata',   
 compress=TRUE  
)  
  
# write raster  
spatialThinMap <- predict(  
 climate[[predictors]],  
 spatialThinModel,  
 filename='./Models/Model 05 Bias Correction - Spatially Thinned Presences/maxentPrediction1970to2000',  
 format='GTiff',  
 overwrite=TRUE,  
 type='cloglog'  
)  
  
  
# plot  
par(mfrow=c(1, 2), pty='s', cex.main=0.8)  
  
plot(rangeMap, main='Unthinned')  
plot(manualSelectMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
plot(rangeMap, main='Geographically Thinned')  
plot(spatialThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(spatiallyThinnedRecords$longitude, spatiallyThinnedRecords$latitude, pch=21, bg='mediumseagreen')



The estimated environmental suitability is now much more evenly spread across the species' range. Unfortunately, removing points reduces sample size. Maxent has some default limits to the complexity with which it can model a species' response to the environment, and these limits are set by sample size. Below 15 records Maxent uses the simplest possible model (a linear model). Hence, this may be why Maxent predicts higher suitability going south using the thinned records.

### Reflection

1. Thinning records costs you data--was it worth it?
2. What effect does record thinning have on the model output? Across the three methods explored here, which seemed to have work best, if any?

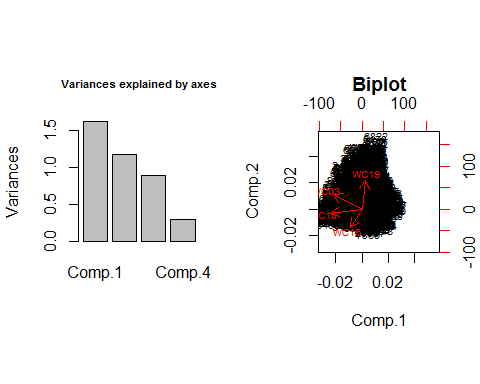
# Environmental thinning of records

A third option is to thin in environmental space. Most bias probably exists because of geographic processes--collectors tend to collect near their place of abode, near roads, etc. In turn, this leads to bias in environmental space. Thus, geographic bias is the original bias creating the bias in the space the model actually works with. However, sometimes bias can be more environmentally based than geographically based. For example, collectors might target climates that are easier to work in (less hot, less cold, etc.). I have seen this occur for specimens of small mammals of California which have very few records--yet definitely occur--in the agriculturally-dominated Central Valley. Even if the original bias is in geographic space, thinning environmentally might make more sense than geographic thinning since the model actually works in environmental, not geographic space.

Most relevant citation: [Varela, S., Anderson, R.P., Garcia-Valdis, R., and Fernandez-Gonz?lez. 2014. Environmental filters reduce the effects of sampling bias and improve predictions of ecological niche models. Ecography 37:1084-1091.](http://dx.doi.org/10.1111/j.1600-0587.2013.00441.x)

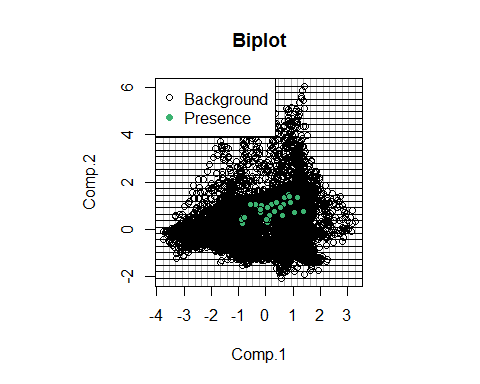
Sara Varela and colleagues (2014) use simulated species to demonstrate their technique, for which they chose just two variables to define an environmental space for thinning. Here we'll thin in multidimensional environmental space by condensing the predictors to two axes using principal components analysis. We'll then divide this space into cells then choose just one record per bin to represent the species. Other ways to do this may also work (e.g., using a set minimum pairwise environmental distance between records, as we used with geographic distance).

# Calculate PCA on environmental variables  
# using "cor=TRUE" because predictors aren't on same scale.  
pca <- princomp(randomBg[ , predictors], cor=TRUE)  
  
# plot  
par(mfrow=c(1, 2), pty='s')  
plot(pca, main='Variances explained by axes', cex.main=0.7)  
biplot(pca, main='Biplot', cex=0.6)



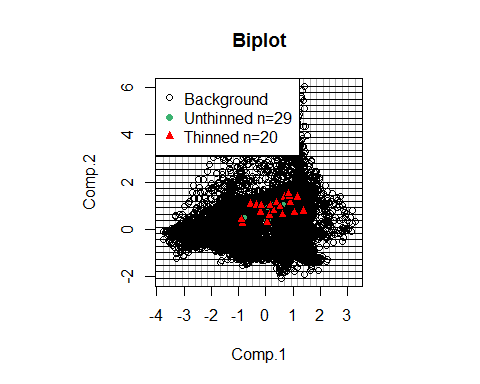
You can see most of the variation in our predictors is explained by the first two axes. Now let's examine the distribution of records in PCA (environmental) space.

# rotate (predict) environment at species' presences into PCA space  
speciesPca <- predict(pca, newdata=records[ , predictors])  
  
# plot  
par(pty='s')  
plot(pca$scores[ , 1:2], main='Biplot')  
  
# demarcate "bin" lines on plot  
numBins <- 30  
  
for (x in seq(min(pca$scores[ , 1]), max(pca$scores[ , 1]), length.out=numBins + 1)) {  
 abline(v=x, lwd=0.5, col=alpha('black', 0.3)) # vertical line  
 for (y in seq(min(pca$scores[ , 2]), max(pca$scores[ , 2]), length.out=numBins + 1)) {  
 abline(h=y, lwd=0.5, col=alpha('black', 0.3)) # horizontal line  
 }  
}  
  
points(speciesPca[ , 1:2],  
 pch=21,  
 bg='mediumseagreen'  
)  
  
legend('topleft',  
 legend=c('Background', 'Presence'),  
 col=c('black', 'mediumseagreen'),  
 pch=c(1, 16),  
 bg='white'  
)



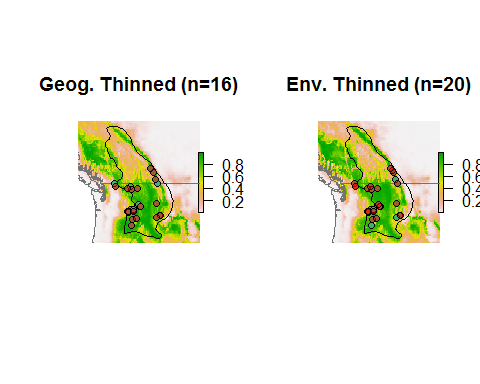
The records are not randomly distributed in environmental space even within the environmental range of the species. This might reflect real differences in preference, or it might reflect collecting bias. We'll assume the clustering is due to sampling bias. Now we'll create a 2D histogram (essentially, the cells in the graph below), tally the number of records in each cell, then randomly select one from each cell for modeling.

# matrix to store number of points in each environmental "cell"  
env <- matrix(rep(0, 40^2), nrow=40)  
  
# calculate bin boundaries  
pc1Bins <- seq(  
 min(pca$scores[ , 1]) - 0.00001,  
 max(pca$scores[ , 1]),  
 length.out=numBins + 1  
)  
pc2Bins <- seq(  
 min(pca$scores[ , 2]) - 0.00001,  
 max(pca$scores[ , 2]),  
 length.out=numBins + 1  
)  
  
# data frame for new records  
envThinnedRecords <- data.frame()  
  
# find records in each environmental cell and randomly select one  
for (pc1 in 1:numBins) {  
 for (pc2 in 1:numBins) {  
   
 # get index of record(s) in this cell  
 recordsInCell <- which(speciesPca[ , 1] > pc1Bins[pc1] &  
 speciesPca[ , 1] <= pc1Bins[pc1 + 1] & speciesPca[ , 2] > pc2Bins[pc2] &  
 speciesPca[ , 2] <= pc2Bins[pc2 + 1])  
   
 # if there is >1 record, select one  
 if (length(recordsInCell) > 0) {  
 chosenRecord <- if (length(recordsInCell)==1) {  
 recordsInCell  
 } else {  
 sample(recordsInCell, 1)  
 }  
   
 envThinnedRecords <- rbind(envThinnedRecords, records[chosenRecord, ])  
 }  
   
 }  
}  
  
# predict PCA scores at thinned presence sites  
thinnedPca <- predict(pca, envThinnedRecords)  
  
# plot  
par(pty='s')  
plot(pca$scores[ , 1:2], main='Biplot')  
  
# show "bin" lines  
for (x in seq(min(pca$scores[ , 1]), max(pca$scores[ , 1]), length.out=numBins + 1)) {  
   
 abline(v=x, lwd=0.5, col=alpha('black', 0.3)) # vertical line  
   
 for (y in seq(min(pca$scores[ , 2]), max(pca$scores[ , 2]), length.out=numBins + 1)) {  
   
 abline(h=y, lwd=0.5, col=alpha('black', 0.3)) # horizontal line  
   
 }  
   
}  
  
points(speciesPca[ , 1:2], pch=21, bg='mediumseagreen')  
points(thinnedPca[ , 1:2], pch=24, bg='red', cex=1)  
legend('topleft',  
 legend=c(  
 'Background',  
 paste0('Unthinned n=', nrow(records)),  
 paste0('Thinned n=', nrow(envThinnedRecords))  
 ),  
 col=c('black', 'mediumseagreen', 'red'),  
 pch=c(1, 16, 17),  
 bg='white'  
)



You can see that the thinned records more evenly represent the conditions experienced by the species within its range. Now let's train a model using the environmentally-thinned records.

# make training data frame with predictors and vector of 1/0 for presence/background  
trainData <- rbind(  
 envThinnedRecords[ , predictors],  
 randomBg[ , predictors]  
)  
  
presBg <- c(rep(1, nrow(envThinnedRecords)), rep(0, nrow(randomBg)))  
  
# create output directory for model object and rasters  
dir.create('./Models/Model 06 Bias Correction - Env Thinned Presences',  
 recursive=TRUE, showWarnings=FALSE)  
  
# model species  
envThinModel <- maxnet(p=presBg, data=trainData)  
  
save(envThinModel,  
 file='./Models/Model 06 Bias Correction - Env Thinned Presences/Model.Rdata',  
 compress=TRUE)  
  
# write current raster  
envThinMap <- predict(  
 climate[[predictors]],  
 envThinModel,  
 filename='./Models/Model 06 Bias Correction - Env Thinned Presences/maxentPrediction1970to2000',   
 format='GTiff', overwrite=TRUE, type='cloglog')  
  
# plot  
par(mfrow=c(1, 2), pty='s')  
  
plot(rangeMap, main=paste0('Geog. Thinned (n=', nrow(spatiallyThinnedRecords), ')'))  
plot(spatialThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
points(spatiallyThinnedRecords$longitude,  
 spatiallyThinnedRecords$latitude,  
 pch=21,  
 bg=alpha('red', 0.5)  
)  
  
plot(rangeMap, main=paste0('Env. Thinned (n=', nrow(envThinnedRecords), ')'))  
plot(envThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
points(envThinnedRecords$longitude,  
 envThinnedRecords$latitude,  
 pch=21,  
 bg=alpha('red', 0.5)  
)



### Reflection

1. Compare the maps of points thinned geographically to the points thinned environmentally. How are they different and how the same?
2. Why do the environmentally thinned points seem to be more spatially clustered? Which thinning process better reflects the environmental preferences of the species in the northern part of its range?
3. Which do you think did a better job, thinning geographically or environmentally?

# Inverse p-weighted bias correction

Some modeling methods allow users to specify weights of presences and absences/background sites. The inverse p-weighted technique uses this to down-weight presence sites in locations that are more likely to have been sampled. We won't explore this here since Maxent doesn't allow weighting of points (directly), but it is worth considering for other types of models.

Most relevant citation: [Stolar, J. and Seton, S.E. 2015. Accounting for spatially biased sampling effort in presence-only species distribution models. Diversity and Distributions 21:595-608.](http://dx.doi.org/10.1111/ddi.12279)

# Microhabitat outliers

An underlying assumption of distribution/niche modeling is that the values of the environmental variables in each cell where a species has a presence reflect the conditions actually experienced by the population in that cell (or at least are highly correlated with relevant conditions). For example, a presence might lie in a cell in which the mean annual precipitation is just 10 mm. The model would this assume that the species is able to persist in very dry conditions. However, maybe the species is really able to persist in that cell because of microhabitat conditions not reflected by the coarse-scale conditions of the cell--maybe there is a stream running through an otherwise dry cell where the species is able to persist, or maybe talus creates cool conditions allowing the species to escape otherwise high temperatures.

It's not possible to explore this situation with a general approach--it depends a lot on your knowledge of the species and the landscape. Nonetheless, removing just a small number of presences (<= ~5%) that fall within these "outlier" cells can have a profound effect on your model predictions.

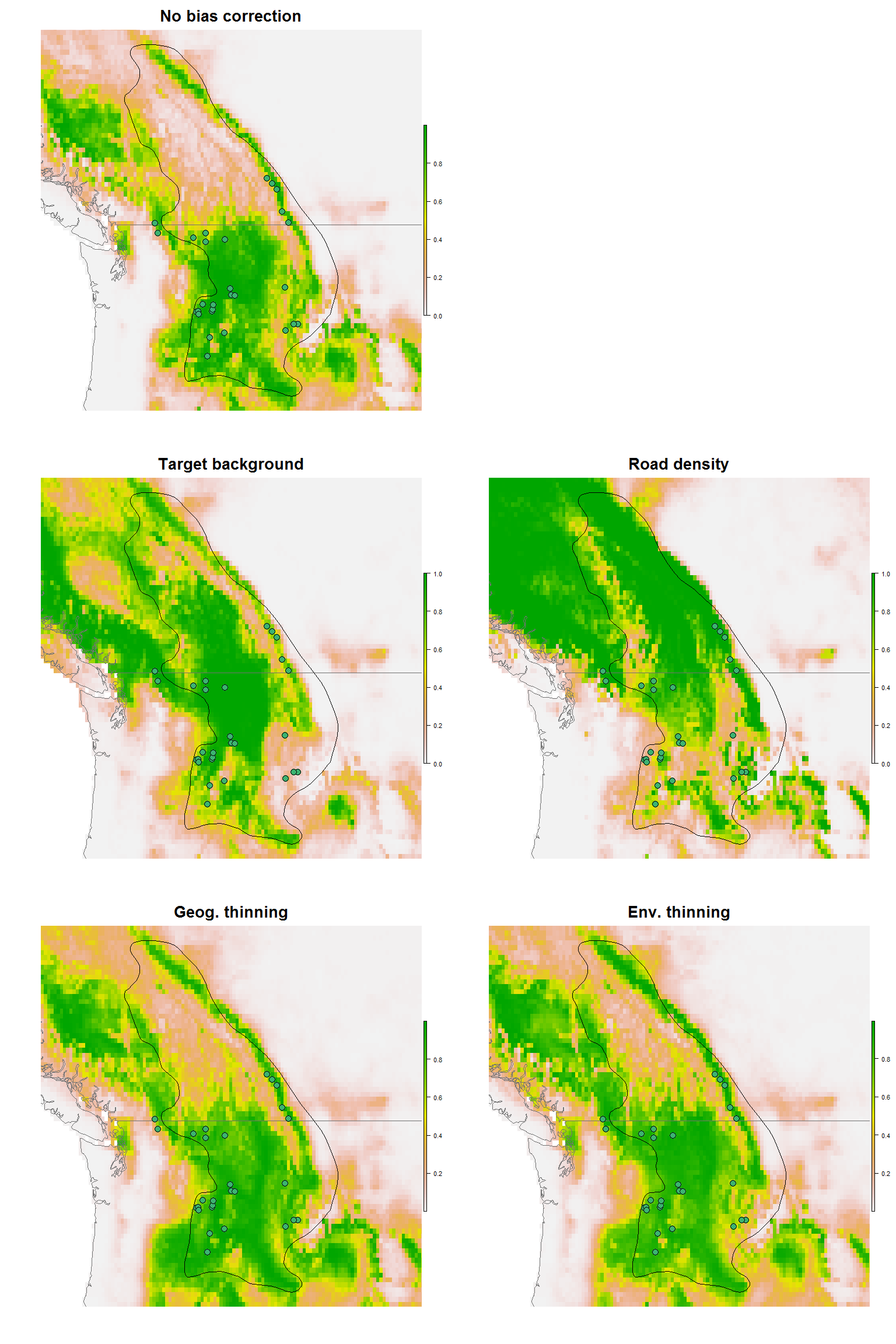
Most relevant citation: [Soley-Guardia, M., Guitirrez, E.E., Thomas, D.M., Ochoa-G, J., Aguilera, M., and Anderson, R.P. 2016. Are we overestimating the niche? Removing marginal localities helps ecological niche models detect environmental barriers. Ecology and Evolution 6:1267-1279.](http://dx.doi.org/10.1002/ece3.1900)

# Underview

Let's take a look at all of the bias-correction maps at once, plus the original "uncorrected" map we made in a previous tutorial.

par(mfrow=c(3, 2), pty='s')  
  
# uncorrected map  
plot(rangeMap, main='No bias correction')  
plot(manualSelectMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
# blank plot  
frame()  
  
# correction with target background  
plot(rangeMap, main='Target background')  
plot(targetBgMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
# correction with background reflecting road density  
plot(rangeMap, main='Road density')  
plot(biasBgMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
# correction by geographic thinning  
plot(rangeMap, main='Geog. thinning')  
plot(spatialThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')  
  
# correction by environmental thinning  
plot(rangeMap, main='Env. thinning')  
plot(envThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen')

par(mfrow=c(3, 2), pty='s')  
  
# uncorrected map  
plot(rangeMap, main='No bias correction', cex.main=2.6, cex=2)  
plot(manualSelectMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen', cex=2.4)  
  
# blank plot  
frame()  
  
# correction with target background  
plot(rangeMap, main='Target background', cex.main=2.6, cex=2)  
plot(targetBgMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen', cex=2.4)  
  
# correction with background reflecting road density  
plot(rangeMap, main='Road density', cex.main=2.6, cex=2)  
plot(biasBgMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen', cex=2.4)  
  
# correction by geographic thinning  
plot(rangeMap, main='Geog. thinning', cex.main=2.6, cex=2)  
plot(spatialThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen', cex=2.4)  
  
# correction by environmental thinning  
plot(rangeMap, main='Env. thinning', cex.main=2.6, cex=2)  
plot(envThinMap, add=TRUE)  
sp::plot(countries, add=TRUE, border='gray45')  
plot(rangeMap, add=TRUE)  
points(records$longitude, records$latitude, pch=21, bg='mediumseagreen', cex=2.4)



So which technique is best? It is hard to say, though there may be indications. In following tutorials we'll explore how to evaluate the performance of models. Mainly people rely on these performance statistics to tell them which bias correction technique is best. However, I firmly believe that visual inspection of maps--just like visual inspection of residuals--is a very important part of model evaluation. Thus we've explored here the effects of bias correction using just our eyes.

There have been very few systematic explorations of the ability of different bias correction techniques for correcting bias. Of note, [Fourcade et al. (2014)](http://dx.doi.org/10.1371/journal.pone.0097122) has conducted the most comprehensive evaluation, albeit with just three species (two real and one simulated). They induced four kinds of bias in their data then tested the ability of five bias correction techniques, including spatial filtering and environmental filtering as we explored here. If you read the abstract and even the paper you might be left with the impression that no technique works well more than 50% of the time, but geographic filtering is still the most dependable method. However, if you carefully examine the large table in that paper you'll see that the real message should be that spatial filtering makes models *worse* most of the time! (And on average other techniques are even worse than that!)

So what should you do? Bias is obviously a problem, so it should be addressed. But the field of distribution and niche modeling has yet to develop universally applicable techniques for bias correction. While unsatisfactory, this is the nature of science--we find what we know and along the way we don't know.