Parasite species distribution modeling

Tad Dallas, Andrew Park, and John M. Drake

Introduction

Species distirbution models and the importance of them

Knowledge gap

What constrains the range of hosts that a parasite can infect? Is there a simple range of host functional traits that can determine the likelihood that a parasite infects a given host species? How well can we predict parasite occurrences given *only* some host life history traits?

Thesis paragraph

Here, I apply a series of predictive models in order to predict parasite occurrence across a range of potential host species for a large set of parasites of freshwater fish, using host functional traits, and geographic location.

thorns in my side:

- Absence data aren't true absences. Should I even train on these data if the model treats them as true absences?
- How much time to invest reading density estimation literature?

Methods

Data and processing

We use an existing global database of fish-parasite associations (Strona et al. 2013) consisting of over 38000 parasite records spanning a large diversity of parasites (Acanthocephala, Cestoda, Monogenea, Nematoda, Trematoda). In order to allow for cross-validation and accurate prediction, we constrained our ananlyses to parasites with a minimum of 50 host records. In other words, we only examined parasites that had been recorded more than 50 times, but these occurrences could be on fewer than 50 host species. The inclusion of duplicate occurrences was only permitted if the parasite was recorded on a host in a different location, based on latitude and longitude values. Our response variable was parasite occurrence (binary), and was predicted using only host life history traits, and geographic location of host capture. Host trait information was obtained through the FishPest database (Strona and Lafferty 2012; Strona et al. 2013), and FishBase (Froese and Pauly 2010). Host traits descriptions are provided in Table 1

Model formulation

We trained a series of models in order to compare predictive performance of different techniques. Each model was trained on 70% of the data, and accuracy was determined from the remaining 30%. This process was repeated z times ($z = 20$). We generated background data by randomly sampling host species where parasite i was not recorded. To maintain proportional training data, the number of random samples was selected to be five times greater than the occurrence records.

Models used

discuss null predictions scenario, and then go into other algorithms used (brt, svm, lr, rf)

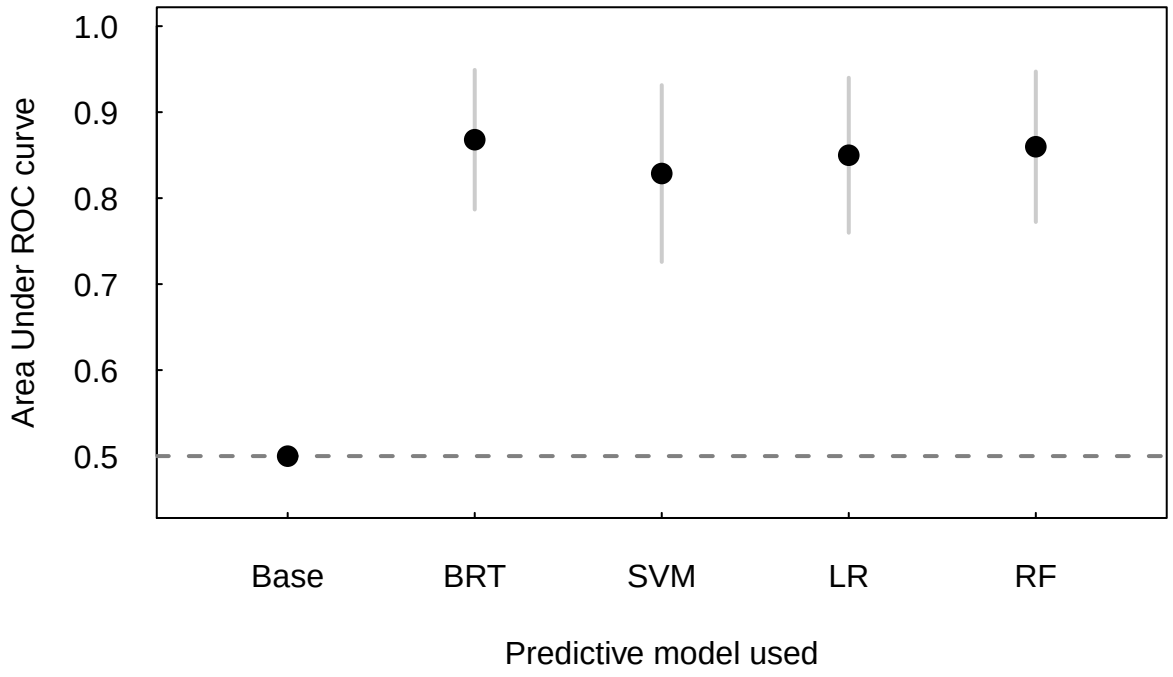
Strona's FishPest database

```
#load('/media/drakelab/Lexar/8910Project/Analysis/pest.Rdata')

## Creating a `fishMatrix` equiv.
#edgePest=edge2matrix(cbind(pest[,'H_SP'], pest[,'P_SP']))
#colnames(edgePest)=unique(pest[,'P_SP'])
#rownames(edgePest)=unique(pest[,'H_SP'])
#edgePest=edgePest[,-which(colSums(edgePest) < 50)]


library(rfishbase); library(gbm); library(ROCR); library(e1071); library(randomForest)
# storage for model outputs
baseline.auc=vector();
brtModel=list(); brt.best.iter=list(); brt.preds=list(); brt.perf=list(); brt.perfAUC=vector()
svmModel=list(); svm.preds=list(); svm.perf=list(); svm.perfAUC=vector()
lrModel=list(); lr.preds=list(); lr.perf=list(); lr.perfAUC=vector()
rfModel=list(); rf.preds=list(); rf.perf=list(); rf.perfAUC=vector()


## Creating a pointsObject
#hostPest = unique(pest[,-c(1:3)], MARGIN=1)
hostPest = pest[,-c(1:3)]
parPest = names(summary(pest[,'P_SP'], maxsum=500)>20);
parPest=parPest[-max(length(parPest))]



ret=list()
for(i in 1:length(parPest)){
#create presence vector
  presence=rep(0, nrow(hostPest))
  presence[which(hostPest[,'P_SP'] == parPest[i])]=1

#make some 'na' into actual NAs
  ugh=which(hostPest=='na', arr.ind=TRUE)
  hostPest[ugh]=NA
  hostPest[,7:17]=apply(hostPest[,7:17],2, as.numeric)

#Only train on some of the absences, since they're totally not true absences
  cutDown=c(which(presence==1), sample(which(presence==0), 5*sum(presence)))
  presence1=presence[cutDown]
  dat=hostPest[cutDown, -c(1,5)]

#Impute the data
  impDat=rfImpute(dat[,-c(1:3)], presence1)
  dat=impDat[,-1]
```

```
flag=0
 while(flag == 0){
  # This makes sure that the test set contains at least 4 hosts on which the parasite actually occurs
  inds=sample(1:nrow(dat), 0.7*nrow(dat))
  if(sum(presence1[inds]) < 4){inds[1:4] = which(presence1 == 1)[1:4]}

  #Set up a prelim train set and a test set
  train = dat[inds,]
  test = dat[-inds,]
  if(all(unique(train$GEO) %in% unique(test$GEO))){flag=1}
 }

 #Presences
  prezTR=presence1[inds]
  prezTE=presence1[-inds]


  #baseline expectations and null models
  baseline.auc[i] = performance(prediction(sample(presence[-inds], length(presence[-inds])), presence[-

 ##trained models
  #boosted regression trees
    brtModel[[i]] = gbm(prezTR ~ ., data=train, n.trees=50000, interaction.depth=4, distribution='bernou
    brt.best.iter[[i]] = gbm.perf(brtModel[[i]], method="OOB")
    brt.preds[[i]] = prediction(predict(brtModel[[i]], newdata=test, n.trees=brt.best.iter[[i]]), prezTE
    brt.perf[[i]] = performance(brt.preds[[i]],"tpr","fpr")
    brt.perfAUC[i]=unlist(performance(brt.preds[[i]], 'auc')@y.values)

  #support vector machines
    svmModel[[i]] = svm(prezTR ~ ., data=train, probability=TRUE)
    svm.preds[[i]] = prediction(predict(svmModel[[i]], test), prezTE)
    svm.perf[[i]] = performance(svm.preds[[i]],"tpr","fpr")
    svm.perfAUC[i] = unlist(performance(svm.preds[[i]], 'auc')@y.values)

  #logistic regression
    lrModel[[i]] = glm(prezTR ~ ., data=train, family=binomial)
    lr.preds[[i]] = prediction(predict(lrModel[[i]], test), prezTE)
    lr.perf[[i]] = performance(lr.preds[[i]],"tpr","fpr")
    lr.perfAUC[i] = unlist(performance(lr.preds[[i]], 'auc')@y.values)

  #random forest
    rfModel[[i]] = randomForest(prezTR ~ ., data=train)
    rf.preds[[i]] = prediction(predict(rfModel[[i]], test), prezTE)
    rf.perf[[i]] = performance(rf.preds[[i]],"tpr","fpr")
    rf.perfAUC[i] = unlist(performance(rf.preds[[i]], 'auc')@y.values)
  print(i)
  }

 ret=cbind(baseline.auc[1:241], brt.perfAUC[1:241], svm.perfAUC[1:241], lr.perfAUC[1:241], rf.perfAUC[1
 colnames(ret)=c('BASE', 'BRT', 'SVM', 'LR', 'RF')
```

```
load('/media/drakelab/Lexar/8910Project/Analysis/pest(reduced).RData')
testRet=as.matrix(ret)
```

```
testRet=apply(testRet,2, as.numeric )
meanAUC = colMeans(testRet)
meanSE = apply(testRet, 2, sd) / sqrt(nrow(testRet))
meanSD = apply(testRet, 2, sd)

plot(1:5, meanAUC, ylim=c(0.45, 1), xlim=c(0.5,5.5), las=1, pch=16, tck=0.01, xaxt='n', xlab='Predictive
abline(h=0.5, col=grey(0.5), lty=2, lwd=2)
segments(x0=1:5, y0=meanAUC+meanSD, y1=meanAUC-meanSD, col=grey(0.8),lwd=2)
points(1:5, meanAUC, pch=16, cex=1.5)
axis(1, at=1:5, labels=c('Base', 'BRT', 'SVM', 'LR', 'RF'), tck=0.01, srt=30)
```

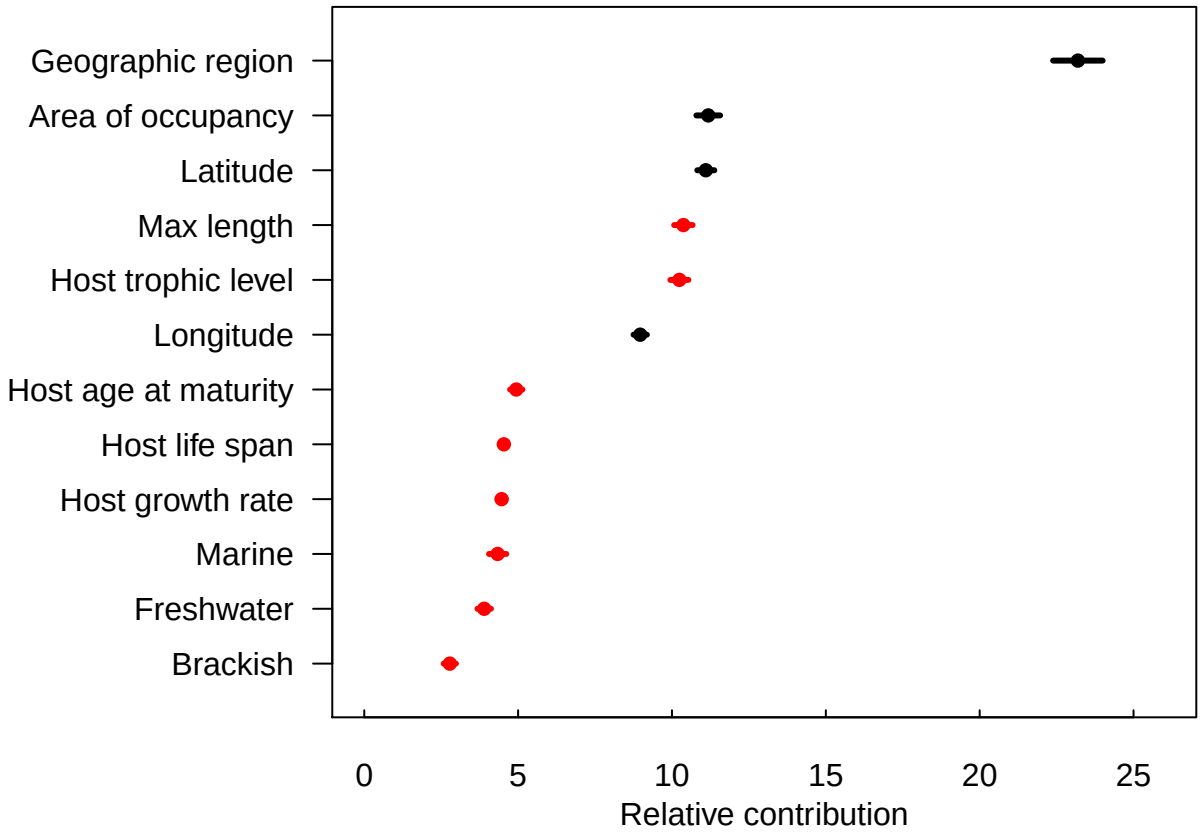


Does predictive power differ systematically among parasites?

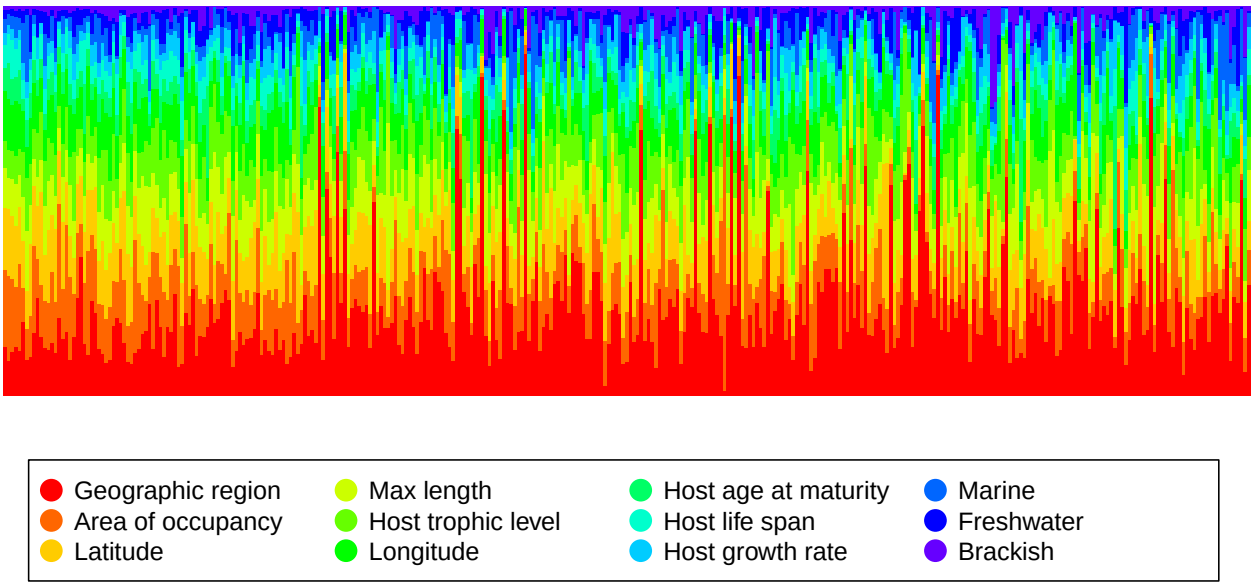
```
rownames(testRet) = parPest[1:241]

#Number of hosts each parasite infects
hostNum = summary(pest[,'P_SP'])[1:99]

#Beating the data up to get parasite 'type'
parType = unique(pest[which(pest[,'P_SP']%in% parPest), 2:4], MARGIN=1)
parType[,1] = gsub('[*]','',parType[,1])
parType = unique(parType, MARGIN=1)
parType = parType[order(parType[,3]),]
```

Are parasite distributions shaped by different factors?

```
varNames=c('Geographic region', 'Area of occupancy', 'Latitude', 'Max length', 'Host trophic level', 'L
#rownames(brtResults[rev(inds),])


brtMean = rowMeans(brtResults)
brtSE = apply(brtResults, 1, sd) / sqrt(ncol(brtResults))
brtSD = apply(brtResults, 1, sd)

inds=order(brtMean)
cols=rev(c(1,1,1,2,2,1,2,2,2,2,2,2))
par(mar=c(3,9,1,1))
plot(brtMean[inds], 1:12, pch=16, xlim=c(0,26), ylim=c(0.5,12.5), las=1, xlab='', ylab='', yaxt='n', co
segments(x0=brtMean[inds] - brtSE[inds], x1=brtMean[inds] + brtSE[inds], y0=1:12, col=cols,lwd=3)
mtext("Relative contribution", side=1, line=2)
axis(2, at=1:12, labels=rev(varNames), las=1)
```




```
layout(matrix(c(1,2),ncol=1), heights=c(1.5, 1))
par(mar=c(0.25,0.25,0.25,0.25))
barplot(brtResults[rev(inds),], col=rainbow(15), xaxt='n', axes=FALSE, border = NA)
plot.new()
legend(0.02,0.85,varNames , pch=16, col=rainbow(15), ncol=4, pt.cex=2, text.width=0.2)
```



Results

Discussion

Tables

Table 1: Host traits examined and their corresponding units

Table 2: Details for parasites modeled, including number of occurrences, and life history information.

Figures

References

Froese, Rainer, and Daniel Pauly. 2010. "FishBase." International Center for Living Aquatic Resources Management.

Strona, Giovanni, and Kevin D Lafferty. 2012. "FishPEST: an Innovative Software Suite for Fish Parasitologists." *Trends in Parasitology* 28 (4): 123.

Strona, Giovanni, Maria Lourdes D Palomares, Nicolas Bailly, Paolo Galli, and Kevin D Lafferty. 2013. "Host Range, Host Ecology, and Distribution of More Than 11 800 Fish Parasite Species: Ecological Archives E094-045." *Ecology* 94 (2): 544–544.