White Plague Initial Trait Space Model

2019-2020 NSF EAGER Annual Report

Code for the initial trait space model and implementation.

Methods:

Our trait-space model applies an environmental filter (i.e. a white plague outbreak) to a reef community and then uses a hierarchical Bayesian model that incorporates intraspecific trait variation to predict the community assemblage after the environmental filter is applied. First, an ideal trait distribution for being resistant to the disease is created by taking the trait averages of the resistant species. Initially, only M. cavernosa values of Thalassobius mediterraneus abundance and Prophenyloxidase concentration were used to create the ideal trait distribution. This ideal, target trait distribution is essentially then environmental filter. Mclust, an R package for normal mixture modeling for model based clustering, classification, and density estimation, is then used to compute the probability of a trait given a species. Traits are drawn from the ideal traits distribution. This is the probability of a trait given the environment where the environment is the target traits. The posterior for the probability of a species given its traits and environment (target traits filter) is calculated. Finally, the posterior probability of a species given the ideal traits is determined by integrating out the traits. The model produces post-outbreak relative abundances of the species based on the resistant trait targets.

Code:

Packages needed

```
library(FD)          # gower dissimiliarity matrix and
#some other clustering stuff
library(ggplot2)     # plotting
library(vegan)       # stats
library(tidyverse)   # data manipulation
library(cluster)     # clustering algorithms
library(factoextra) # clustering visualization
library(dendextend) # for making dendograms fancy
library(clValid)
# has the dunns index function for comparing cluster numbers
library(mclust)
# normal mixture modeling for model based clustering,
#classification, and density estimation
library(randomForest) # Random Forests functions
library(MASS)
```

Load in data

```
trait.data<-read.csv("traits_loggenes_env.csv",row.names=1)
#trait dataframes where each column is a different species
propheno<-read.csv("propheno.csv") #Prophenyloxidase concentration
thalmed<-read.csv("thalmed.csv") #Thalassobius mediterraneus
```

Step 1a.

Instead of an environmental gradient, we want an ideal trait distribution for being resistant. Initially using M. cavernosa traits as the ideal because it is the most resistant species

Get data for creating target traits distribution

```
#covariance
cov(propheno$Mcav.AntOx_Prophenyloxidase,thalmed$Mcav.Bac_thalassobius_mediterraneus)
```

```
## [1] 0.0005296897
```

```
#propheno mean and variance
mean(propheno$Mcav.AntOx_Prophenyloxidase)
```

```
## [1] 4.81985
```

```
var(propheno$Mcav.AntOx_Prophenyloxidase)
```

```
## [1] 3.397597
```

```
#thalmed mean and variance
mean(thalmed$Mcav.Bac_thalassobius_mediterraneus)
```

```
## [1] 0.005633363
```

```
var(thalmed$Mcav.Bac_thalassobius_mediterraneus)
```

```
## [1] 5.477145e-06
```

Create the target traits distribution

```
reps<-c(1:10)
# defined early, but this is basically the number of simulations being run.
#propheno values first
covariance<-matrix(c(3.3976,0.0005,0.0005,0.000005),ncol=2)
means<-c(4.8199,0.0056)
targets <-mvrnorm(n=1000,mu=means,Sigma=covariance)
#### Fit multivariate normal mixture model
pdf_targets <- mvn(modelName="XXX",targets,warn=TRUE)
#Computes the mean, covariance, and log-likelihood
#from fitting a single Gaussian to given data (univariate or multivariate normal).
#"XXX" for a general ellipsoidal Gaussian.
### Might want to change this eventually,
#try seeing what using Mclust instead looks like.

par_targets <-  pdf_targets$parameters
```

Step 1b:

Use Mclust to compute the probability of a trait given the species

```
trait1<-trait.data$AntOx_Prophenyloxidase #set trait 1
trait2<-trait.data$Bac_thalassobius_mediterraneus #set trait 2
species<-trait.data$Species
Sp_data <- cbind(species,trait1,trait2)

cnat<-as.data.frame(Sp_data)%>%
  filter(species==1)
pdf_cnat<-Mclust(cnat[,2:3],warn=TRUE)
par_cnat<-pdf_cnat$parameters

mcav<-as.data.frame(Sp_data)%>%
  filter(species==2)
pdf_mcav<-Mclust(mcav[,2:3],warn=TRUE)
```

```
par_mcav<-pdf_mcav$parameters

oann<-as.data.frame(Sp_data)%>%
  filter(species==3)
pdf_oann<-Mclust(oann[,2:3],warn=TRUE)
par_oann<-pdf_oann$parameters

ofav<-as.data.frame(Sp_data)%>%
  filter(species==4)
pdf_ofav<-Mclust(ofav[,2:3],warn=TRUE)
par_ofav<-pdf_ofav$parameters

past<-as.data.frame(Sp_data)%>%
  filter(species==5)
pdf_past<-Mclust(past[,2:3],warn=TRUE)
par_past<-pdf_past$parameters

ppor<-as.data.frame(Sp_data)%>%
  filter(species==6)
pdf_ppor<-Mclust(ppor[,2:3],warn=TRUE)
par_ppor<-pdf_ppor$parameters

ssid<-as.data.frame(Sp_data)%>%
  filter(species==7)
pdf_ssid<-Mclust(ssid[,2:3],warn=TRUE)
par_ssid<-pdf_ssid$parameters
```

Step 2a:

Draw traits from the target traits distribution

```
#drawing samples from mixture densities fitted in Step 1A
N<- 100
trt_sample <-  matrix(0,length(reps)*N,2)
trt_sample[1:N,] <- sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(N+1):(2*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(2*N+1):(3*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(3*N+1):(4*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(4*N+1):(5*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(5*N+1):(6*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(6*N+1):(7*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(7*N+1):(8*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(8*N+1):(9*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
trt_sample[(9*N+1):(10*N),]<-sim(pdf_targets$modelName,par_targets,N)[,2:3]
```

Probability of a trait given the "environment," where the environment is the target traits distribution

```
## computing(P(T/E))
P_T_E <- rep(0,length(reps)*N)
P_T_E[1:N] <- dens(pdf_targets$modelName,trt_sample[1:N,],parameters=par_targets)
P_T_E[(N+1):(2*N)] = dens(pdf_targets$modelName,trt_sample[(N+1):(2*N),],
                          parameters=par_targets)
P_T_E[(2*N+1):(3*N)] = dens(pdf_targets$modelName,trt_sample[(2*N+1):(3*N),],
                            parameters=par_targets)
P_T_E[(3*N+1):(4*N)] = dens(pdf_targets$modelName,trt_sample[(3*N+1):(4*N),],
```

```
                              parameters=par_targets)
P_T_E[(4*N+1):(5*N)] = dens(pdf_targets$modelName,trt_sample[(4*N+1):(5*N),],
                              parameters=par_targets)
P_T_E[(5*N+1):(6*N)] = dens(pdf_targets$modelName,trt_sample[(5*N+1):(6*N),],
                              parameters=par_targets)
P_T_E[(6*N+1):(7*N)] = dens(pdf_targets$modelName,trt_sample[(6*N+1):(7*N),],
                              parameters=par_targets)
P_T_E[(7*N+1):(8*N)] = dens(pdf_targets$modelName,trt_sample[(7*N+1):(8*N),],
                              parameters=par_targets)
P_T_E[(8*N+1):(9*N)] = dens(pdf_targets$modelName,trt_sample[(8*N+1):(9*N),],
                              parameters=par_targets)
P_T_E[(9*N+1):(10*N)] = dens(pdf_targets$modelName,trt_sample[(9*N+1):(10*N),],
                               parameters=par_targets)
```

Step 2b.

Get the probability that a species has the sampled traits

```
#computing P(T/Sk) using Mclust done earlier
#note that in 2014 laughlin does the exp(trt_sample)
P_T_cnat<-dens(pdf_cnat$modelName,trt_sample,parameters=par_cnat)

P_T_oann<-dens(pdf_oann$modelName,trt_sample,parameters=par_oann)

P_T_ofav<-dens(pdf_ofav$modelName,trt_sample,parameters=par_ofav)

P_T_ppor<-dens(pdf_ppor$modelName,trt_sample,parameters=par_ppor)

P_T_past<-dens(pdf_past$modelName,trt_sample,parameters=par_past)

P_T_ssid<-dens(pdf_ssid$modelName,trt_sample,parameters=par_ssid)

P_T_mcav<-dens(pdf_mcav$modelName,trt_sample,parameters=par_mcav)

P_T_S <- cbind(P_T_cnat,P_T_mcav,P_T_oann,P_T_ofav,P_T_past,P_T_ppor,P_T_ssid)
summary(P_T_S)
```

```
##     P_T_cnat    P_T_mcav          P_T_oann          P_T_ofav
##  Min.   :0   Min.   :   0.00   Min.   : 0.0000   Min.   :0
##  1st Qu.:0   1st Qu.:   0.00   1st Qu.: 0.0000   1st Qu.:0
##  Median :0   Median :   0.00   Median : 0.0000   Median :0
##  Mean   :0   Mean   :  19.47   Mean   : 0.0248   Mean   :0
##  3rd Qu.:0   3rd Qu.:   0.00   3rd Qu.: 0.0000   3rd Qu.:0
##  Max.   :0   Max.   :4091.43   Max.   :23.4666   Max.   :0
##     P_T_past          P_T_ppor          P_T_ssid
##  Min.   : 0.0000   Min.   :0.00000   Min.   :0.000e+00
##  1st Qu.: 0.0000   1st Qu.:0.00000   1st Qu.:0.000e+00
##  Median : 0.0000   Median :0.00000   Median :0.000e+00
##  Mean   : 0.1513   Mean   :0.00142   Mean   :3.826e-32
##  3rd Qu.: 0.0000   3rd Qu.:0.00000   3rd Qu.:0.000e+00
##  Max.   :14.6549   Max.   :0.78667   Max.   :3.780e-29
```

Multiply likelihood by a prior to get the posterior

```
#flat prior
P_T_S_pr <-  P_T_S/7
#sps abundancesfrom the miller paper
P_T_S_pr <- cbind(P_T_cnat*0.032,P_T_mcav*0.035,P_T_oann*0.4,
                  P_T_ofav*0.4,P_T_past*0.05,P_T_ppor*0.05,P_T_ssid*0.033)
P_T_S_pr_sum<-apply(P_T_S_pr,1,sum)
summary(P_T_S_pr_sum)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   0.0000   0.0000   0.0000   0.6992   0.0000 143.2000
```

Step 2C.

Get the Posterior for the probability of a species given traits and environment

```
#computing P(Sk/T,E) using Bayes theorem

P_S_T_E = matrix(0,dim(P_T_S)[1],7)

for (i in 1:dim(P_T_S)[1]){
  P_S_T_E[i,] = exp(log(P_T_S_pr[i,]) - log(P_T_S_pr_sum[i]))
  } #using log
P_S_T_E[is.nan(P_S_T_E)] <- 0
```

Step 2D

Get the posterior probability of a species given the target resistant traits, integrate out the traits.

```
#P(Sk/T) by integrating out T's
P_S_E_all = matrix(0,length(reps)*N,7)
P_S_E_unnorm = matrix(0,length(reps),7) #unnormalised P_S_E
P_S_E = matrix(0,length(reps),7)

#before MC integration (with log)
for (i in 1:dim(P_S_E_all)[1]){
  P_S_E_all[i,]=exp(log(P_T_E[i])+log(P_S_T_E[i,]))}
#MC integration and normalisation
c=1
for (k in 1:length(reps)){
  c=(k-1)*N+1
  P_S_E_unnorm[k,]=apply(P_S_E_all[c:(c+(N-1)),],2,mean) #MC
  P_S_E[k,]=P_S_E_unnorm[k,]/sum(P_S_E_unnorm[k,]) #normalisation
  }
apply(P_S_E,1,sum) ### check that probs sum to one
```

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

```
P_S_E
```

```
##       [,1]      [,2]         [,3] [,4]      [,5]         [,6]       [,7]
##  [1,]    0 0.5567918 2.474304e-03    0 0.3086113 2.052332e-03 0.13007030
##  [2,]    0 0.7598679 1.855619e-13    0 0.1471057 3.572541e-03 0.08945382
##  [3,]    0 0.6304456 3.028299e-05    0 0.1764152 4.808568e-03 0.18830034
##  [4,]    0 0.6570271 1.268935e-03    0 0.2131949 1.141970e-02 0.11708942
##  [5,]    0 0.7278021 1.835150e-09    0 0.1168261 8.130260e-03 0.14724158
##  [6,]    0 0.6110141 8.837922e-17    0 0.1857177 2.443964e-03 0.20082419
```

```
##  [7,]    0 0.5459131 1.450701e-83    0 0.2540268 3.703458e-03 0.19635664
##  [8,]    0 0.6635370 1.114930e-03    0 0.2199744 7.734833e-24 0.11537373
##  [9,]    0 0.6908051 4.256052e-04    0 0.1997496 6.135968e-03 0.10288374
## [10,]    0 0.6908998 0.000000e+00    0 0.2262230 1.262324e-03 0.08161490
```

```
### probs
```

Post-Outbreak abundances based on 'resistant trait targets'

```
### Boxplots
boxplot(ylab="Relative abundances",
        main="Sp. abundances post-outbreak, mcav is target",
        ylim=c(0,1),P_S_E[,1],P_S_E[,2],
        P_S_E[,3],P_S_E[,4],P_S_E[,5],P_S_E[,6],P_S_E[,7],
      names=c("Cnat","Mcav","Oann","Ofav","Past","Ppor","Ssid"),
      range=0,cex.main=1,font.lab=1, cex.lab=1.5,
      border=c("darkmagenta","cornflowerblue","blue",
               "darkslategray4","green","orange","orangered","red"),
      col=c("darkmagenta","cornflowerblue","blue","darkslategray4",
            "green","orange","orangered","red"))
```