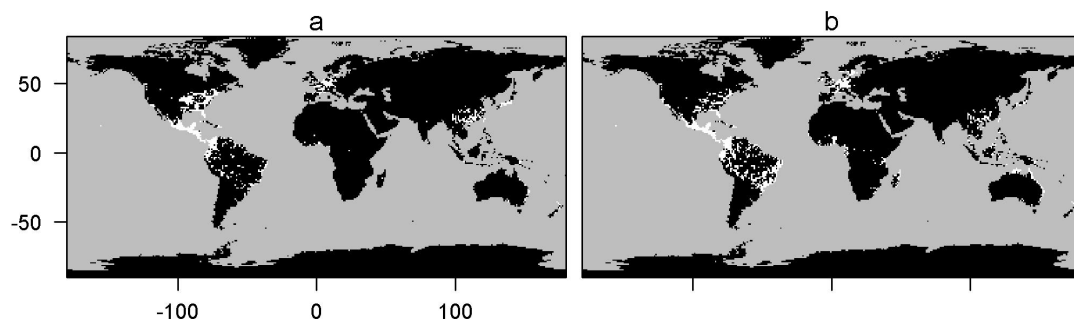


# ***Supplementary Material:*** **Combining phylogenetic and occurrence** **information for risk assessment of pest and** **pathogen interaction with host plants**

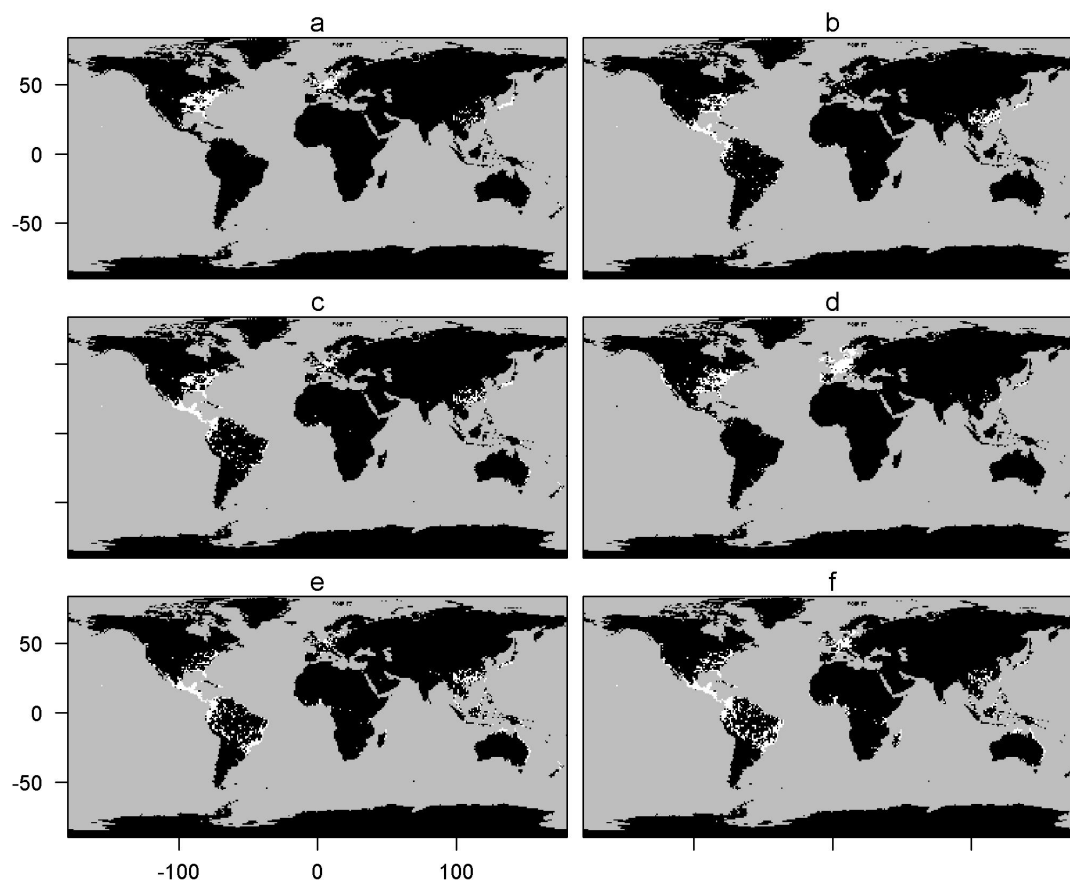
**Angel L. Robles-Fernandez Andres Lira-Noriega \***

\*Correspondence:  
 Corresponding Author  
 aliranoriega@gmail.com

## **1 SUPPLEMENTARY FIGURES**



**Figure S1.** Interaction-or-infection spatial index and risk binary maps for  $M_1$  (a) and  $M_2$  (b) in logistic scale with values over 0.5. Maps summarize information at 1-degree spatial resolution.



**Figure S2.** Spatial interaction-or-infection index binary raster in logistic scale with values over 0.5, for three cases of host range size and different level of phylogenetic constraint. Maps in a, c and e correspond to the interaction-or-infection index for host plant genera from the original host-beetle database ( $M_1$ ), and maps in b, d, and f correspond to the interaction-or-infection index for host plant genera from the phylogenetic tree ( $M_2$ ). The three cases correspond to a narrow host range phylogenetically constrained species (*Xyleborus xylographus*), a wide host range vector phylogenetically constrained species (*Xyleborus glabratus*), and a wide host range phylogenetically dispersed species (*Xylosandrus crassiusculus*). Maps summarize information at 1-degree spatial resolution.

## 2 R CODE

### 2.1 Introduction

All the results in the article were calculated using R package geotax available through GitHub<sup>1</sup>. This package contains tools to spatialize biodiversity information over geographic space. R code file and supplies to run the code are attached as supplementary zip file and also are available in the GitHub package repository. Suggestions are welcome.

The code needs the follow files:

- full\_taxa\_beetle\_plant.csv (contains interaction information as table)
- M1.csv (presence-absence matrix of host by sites)
- M2.csv (presence-absence matrix of host by sites)
- m1\_pts.csv (host presence points)
- m2\_pts.csv (host presence points) and
- host.tree.tre (host phylogenetic tree)

Briefly, the code reads a phylogenetic tree to get the distance  $D$ . Then reads the interaction table to convert it in  $I$  interaction matrix. Next it performs the logistic regression to obtain coefficients  $\beta_0$  and  $\beta_1$ . Using these coefficients it transforms  $D$  in  $P$ . After this it reads  $M_1$  and  $M_2$  matrices and using  $P$  and  $M$  gets  $G$  for each case (matrix). Finally, it correlates the row sums of  $G$  and  $M_{1,2}$  and obtains the linear regression between correlation coefficients and host range (number of host)  $H$ . The code uses non-standard evaluation syntax with use of magrittr package pipe.

```

20
21 #Support material
22 #path the the material directory
23 setwd("~/material")
24
25 devtools::install_github("alrobles/geotax")
26 require(geotax)
27
28 pack <- c("caper","phytools","geiger",
29           "dismo","randomForest","SDMTools")
30
31 lapply(pack, install.packages)
32 lapply(pack, require, character.only = TRUE)
33
34 ####host phylogenetical tree build with phylomatic
35 tree <- read.tree("host.tree")
36
37 #distance matrix
38 d <- tree %>% cophenetic.phylo()
39 d <- log10(d+1)
40

```

<sup>1</sup> <https://github.com/alrobles/geotax>

```

41 #not run
42 #d <- log10(phy_dist + 1) # using data object
43
44 #####
45 # create incidence matrix from database      #
46 # and filter with distance matrix          #
47 #####
48
49 db <- read.csv("full_taxa_beetle_plant.csv", header = T)
50 i_matrix <- incidence(db[, c(7, 13)][, colnames(d)])
51
52
53 # filtering incidence matrix with incidence > 2
54 i <- i_matrix[rowSums(i_matrix)>2, ]
55
56 # distance matrix
57
58 #logistic regression coefficients for all the set
59 coef <- log_reg_bootstrap(i, d, 1000)
60
61 #logistic regression coefficients for each case
62 coef_all <- sapply(1:nrow(i),
63                   function(x) log_reg_bootstrap(i[x, ], drop=F), d, 1000)
64
65 #filter matrix for each case taking only source infected host to all target
66 d_all <- lapply(1:nrow(i), function(x){ d[which(i[x, ] %in% 1), ] })
67 names(d_all) <- rownames(i)
68
69 #probability matrix for all set
70 p <- prob_logit(coef[c(1,7)], d)
71
72 #probability matrix for each case
73 p_all <- lapply(1:nrow(coef_all),
74               function(x) prob_logit(coef_all[x, c(1,7)], d_all[[x]]))
75 names(p_all) <- rownames(i)
76
77 #points database retrived from gbif by species data base
78 ml_db <- read.csv("ml_pts.csv")
79
80 #points database retrived from gbif by genera data base
81 m2_db <- read.csv("m2_pts.csv")
82
83 #presense-absence matrix from ml_db at 1 degree resolution
84 M1 <- read.csv("M1.csv")
85 row.names(M1) <- M1[,1]

```

```

86 M1 <- M1[ , -1]
87
88 ##presense-absence matrix from m1_db 1 degree resolution
89 M2 <- read.csv("M2.csv")
90 row.names(M2) <- M2[ ,1]
91 M2 <- M2[ , -1]
92
93 #M1 and M2 can be build with geotax::PAM() with geotax::world shape
94
95 #match data from genera in points databases and probability matrix
96
97 n1 <- colnames(p)[ unique(m1_db[ ,1]) %>% as.matrix %>%
98                   match(. , colnames(p)) %>% na.omit ]
99 n2 <- colnames(p)[ unique(m2_db[ ,1]) %>% as.matrix %>%
100                   match(. , colnames(p)) %>% na.omit ]
101
102 #presence-absence matrix M1 and M2 filter with match
103 M1 <- M1[n1, ] %>% as.matrix
104 M2 <- M2[n2, ] %>% as.matrix
105
106 #probability matrix with filter
107 p_all_m1 <- lapply(p_all , function(x) x[ , n1] )
108 p_all_m2 <- lapply(p_all , function(x) x[ , n2] )
109
110 p_m1 <- p[n1, n1]
111 p_m2 <- p[n2, n2]
112
113 #calculate geographical phylogenetical information
114
115 #full set case
116 G1 <- p_m1 %*% M1
117 G2 <- p_m2 %*% M2
118
119 #all single case G matrix
120 G1_all <- lapply(p_all_m1, function(x) x %*% M1)
121 G2_all <- lapply(p_all_m2, function(x) x %*% M2)
122
123 g1 <- richness_PAM(world, G1, 1 ) #g raster
124 g1_log <- 1 - g1 %>% normalize() %>% cumulative() %>% logistic()
125
126 g2 <- richness_PAM(world, G2, 1 )
127 g2_log <- 1 - g2 %>% normalize() %>% cumulative() %>% logistic()
128
129 g1_all <- lapply(G1_all , function(x) richness_PAM(world, x, 1 ) )
130 g2_all <- lapply(G2_all , function(x) richness_PAM(world, x, 1 ) )

```

```

131
132 #g raster in logistic scale
133 g1_log_all <- lapply(g1_all, function(x){
134   1 - x %>% normalize() %>% cumulative() %>% logistic() } )
135 g2_log_all <- lapply(g2_all, function(x){
136   1 - x %>% normalize() %>% cumulative() %>% logistic() } )
137
138 #richnes from M1 and M2 as raster
139 richness_M1 <- richness_PAM(world, M1, 1 )
140 richness_M2 <- richness_PAM(world, M2, 1 )
141
142
143 #correlation between g and M for both cases
144 cor_r_g1_M1 <- lapply(g1_log_all, function(i){
145   x <- i@data@values
146   y <- richness_M1@data@values
147   cor(x,y)
148 }) %>% do.call(rbind, .)
149
150 cor_r_g2_M2 <- lapply(g2_log_all, function(i){
151   x <- i@data@values
152   y <- richness_M2@data@values
153   cor(x,y) }) %>% do.call(rbind, .)
154
155
156 #data frame with correlation and host range
157 r_vs_h <- data.frame(rowSums(i), exp(-rowSums(i)),
158                     cor_r_g1_M1, cor_r_g2_M2)
159
160 colnames(r_vs_h) <- c("host_range", "h", "r_M1", "r_M2" )
161
162 model_M1 <- lm(r_M1 ~ h , r_vs_h)
163 model_M2 <- lm(r_M2 ~ h , r_vs_h)

```