Supporting information

Table S1: Species studied and number of occurrences available in BANDASCA database at a resolution of 10 km width UTM cells.

Species	Nº of occurrences
Bubas bison	169
Bubas bubalus	213
Caccobius schreberi	212
Copris hispanus	205
Copris lunaris	180
Euoniticellus fulvus	255
Euonthophagus amyntas	185
Onthophagus fracticornis	171
Onthophagus furcatus	319
Onthophagus lemur	147
Onthophagus opacicollis	154
Onthophagus similis	311
Onthophagus taurus	391
Onthophagus vacca	355

Table S2: Eigenvalues and variance explained for each factor from a PCA of the 29 variables analysed. The factors in bold are those with eigenvalue >1 and were selected for the subsequent analyses.

Factors	Eigenvalue	% of variance explained	Cumulative % variance explained
1	**12.97191**	**44.73071**	**44.7307**
2	**8.22967**	**28.37818**	**73.1089**
3	**3.00149**	**10.34998**	**83.4589**
4	**1.16006**	**4.00021**	**87.4591**
5	**1.03982**	**3.58560**	**91.0447**
6	0.85888	2.96165	94.0063
7	0.63293	2.18252	96.1889
8	0.42208	1.45544	97.6443
9	0.28696	0.98953	98.6338
10	0.14081	0.48555	99.1194
11	0.08691	0.29969	99.4191
12	0.05427	0.18714	99.6062
13	0.03680	0.12688	99.7331
14	0.02037	0.07025	99.8034
15	0.01575	0.05431	99.8577
16	0.01008	0.03477	99.8924
17	0.00884	0.03047	99.9229
18	0.00698	0.02407	99.9470
19	0.00424	0.01462	99.9616
20	0.00415	0.01431	99.9759
21	0.00285	0.00984	99.9858
22	0.00167	0.00577	99.9915
23	0.00118	0.00408	99.9956
24	0.00081	0.00280	99.9984
25	0.00029	0.00101	99.9994
26	0.00009	0.00032	99.9997
27	0.00008	0.00027	100.0000
28	0.00000	0.00000	100.0000

Table S3: Variables and factor loadings from PCA analysis. The most important variables for each PCA factor selected are highlighted in bold. * indicates the highest factor loading values for each factor.

Variable	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Mean altitude	-0.601385	0.631511	-0.412350	-0.025136	-0.086015
Minimum altitude	-0.424255	0.738791	-0.362353	-0.161703	0.052871
Maximum altitude	-0.673290	0.488897	-0.389170	0.104526	-0.202866
Altitudinal range	-0.604853	-0.002150	-0.229594	0.327083	-0.367405
Annual Mean Temperature	0.872139	-0.425117	0.143941	0.149267	-0.000430
Mean Diurnal Range	0.496098	0.620147	-0.467709	-0.136929	0.050583
Isothermality	0.120588	-0.457115	-0.115594	-0.776212*	-0.189735
Temperature Seasonality	0.346190	0.801908	-0.303308	0.299506	0.143507
Max Temperature of Warmest Month	0.912777	0.151553	-0.231211	0.181691	0.084102
Min Temperature of Coldest Month	0.580662	-0.762328	0.238195	0.063751	-0.038814
Temperature Annual Range	0.437286	0.741782	-0.411000	0.127435	0.112226
Mean Temperature of Wettest Quarter	0.431512	0.118323	0.737270*	0.135021	-0.086262
Mean Temperature of Driest Quarter	0.675089	-0.223404	-0.366586	-0.092434	0.022077
Mean Temperature of Warmest Quarter	0.950589*	-0.087132	0.004155	0.254472	0.044888
Mean Temperature of Coldest Quarter	0.695517	-0.664871	0.224477	0.046828	-0.061880
Annual Precipitation	-0.676048	-0.674017	-0.216972	0.120745	0.086983
Precipitation of Wettest Month	-0.500732	-0.801441	-0.232599	0.113244	0.071353
Precipitation of Driest Month	-0.927993	0.090569	0.248523	0.031381	0.004707
Precipitation Seasonality	0.657626	-0.597858	-0.347288	0.068762	-0.068313
Precipitation of Wettest Quarter	-0.457883	-0.824900	-0.280031	0.088325	0.070189
Precipitation of Driest Quarter	-0.936290	0.025761	0.236240	0.051567	0.031136
Precipitation of Warmest Quarter	-0.910990	0.014589	0.332038	0.106629	0.023898
Precipitation of Coldest Quarter	-0.338070	-0.836770*	-0.394876	0.047057	0.086614
Solar radiation monthly average	0.166144	0.016862	0.069182	-0.138772	0.817528*
Aridity index	0.880048	0.297289	0.248760	-0.007729	-0.090736
PET	0.845739	-0.316865	-0.326104	0.161137	-0.066836
AET	-0.639882	-0.676323	-0.211893	0.123169	0.127876
Water balance	-0.864800	-0.436498	-0.119452	0.054002	0.120258
Distance to the Pyrenees	0.610325	-0.529178	-0.477031	-0.120686	-0.127314

Table S4: Values of Pearson's correlation among the Maps of Biogeographic Ignorance of all species considered.

	B. bison	*B. bubalus*	*C. schreberi*	*C. hispanus*	*C. lunaris*	*E. fulvus*	*E. amyntas*	*O. fracticornis*	*O. furcatus*	*O. lemur*	*O. opacicollis*	*O. similis*	*O. taurus*
B. bubalus	0.94												
C. schreberi	0.89	0.96											
C. hispanus	0.95	0.97	0.93										
C. lunaris	0.87	0.95	0.96	0.92									
E. fulvus	0.92	0.96	0.98	0.95	0.96								
E. amyntas	0.87	0.96	0.97	0.93	0.96	0.95							
O. fracticornis	0.88	0.91	0.94	0.88	0.94	0.93	0.93						
O. furcatus	0.90	0.96	0.96	0.96	0.95	0.97	0.96	0.93					
O. lemur	0.85	0.90	0.92	0.88	0.92	0.91	0.93	0.96	0.93				
O. opacicollis	0.94	0.91	0.90	0.92	0.88	0.94	0.85	0.89	0.90	0.87			
O. similis	0.90	0.95	0.97	0.94	0.97	0.98	0.95	0.94	0.96	0.94	0.92		
O. taurus	0.93	0.96	0.97	0.96	0.95	0.98	0.95	0.94	0.98	0.93	0.95	0.98	
O. vacca	0.94	0.97	0.97	0.96	0.95	0.98	0.95	0.94	0.96	0.93	0.95	0.98	0.99

Table S5: Values of model performance metrics for each Species Distribution Modelling technique and species.

	Sensitivity					Specificity					Kappa					TSS				
Species	Bioclim	GAM	GLM	MaxEnt	RF	Bioclim	GAM	GLM	MaxEnt	RF	Bioclim	GAM	GLM	MaxEnt	RF	Bioclim	GAM	GLM	MaxEnt	RF
B. bison	0.80	0.88	0.81	0.89	0.88	0.70	0.65	0.72	0.74	0.71	0.50	0.54	0.53	0.63	0.59	0.50	0.53	0.53	0.62	0.59
B. bubalus	0.68	0.64	0.68	0.68	0.67	0.59	0.74	0.72	0.72	0.78	0.27	0.38	0.40	0.39	0.45	0.27	0.38	0.40	0.40	0.45
C. schreberi	0.59	0.68	0.79	0.71	0.56	0.61	0.66	0.53	0.65	0.84	0.20	0.34	0.32	0.36	0.39	0.20	0.34	0.32	0.36	0.40
C. hispanus	0.72	0.86	0.85	0.88	0.76	0.63	0.54	0.52	0.57	0.64	0.35	0.41	0.38	0.45	0.41	0.34	0.40	0.38	0.44	0.41
C. lunaris	0.70	0.71	0.82	0.72	0.78	0.57	0.75	0.64	0.76	0.73	0.26	0.46	0.46	0.48	0.51	0.26	0.46	0.46	0.48	0.51
E. fulvus	0.52	0.68	0.67	0.74	0.62	0.63	0.68	0.65	0.60	0.73	0.15	0.36	0.32	0.34	0.34	0.15	0.36	0.32	0.34	0.34
E. amyntas	0.50	0.61	0.72	0.66	0.63	0.66	0.86	0.70	0.80	0.79	0.16	0.47	0.42	0.46	0.42	0.16	0.47	0.42	0.46	0.42
O. fracticornis	0.62	0.80	0.78	0.83	0.77	0.76	0.75	0.75	0.71	0.84	0.38	0.55	0.53	0.55	0.61	0.38	0.55	0.53	0.55	0.61
O. furcatus	0.66	0.72	0.71	0.69	0.73	0.48	0.52	0.57	0.59	0.61	0.15	0.24	0.28	0.28	0.35	0.15	0.24	0.28	0.28	0.35
O. lemur	0.61	0.65	0.58	0.72	0.79	0.63	0.80	0.81	0.69	0.73	0.24	0.46	0.39	0.42	0.52	0.24	0.46	0.39	0.42	0.52
O. opacicollis	0.55	0.73	0.66	0.75	0.70	0.58	0.62	0.72	0.66	0.73	0.13	0.35	0.37	0.40	0.43	0.13	0.35	0.37	0.40	0.43
O.similis	0.49	0.77	0.70	0.72	0.71	0.77	0.71	0.72	0.74	0.78	0.26	0.48	0.42	0.46	0.49	0.26	0.48	0.42	0.46	0.49
O. taurus	0.40	0.50	0.56	0.56	0.57	0.67	0.69	0.62	0.68	0.70	0.07	0.19	0.18	0.24	0.27	0.08	0.19	0.18	0.24	0.27
O. vacca	0.63	0.63	0.73	0.68	0.64	0.51	0.66	0.60	0.62	0.66	0.14	0.29	0.33	0.30	0.31	0.14	0.29	0.33	0.30	0.31

Table S6: Number and correspondent percentage of cell in the study area with BI values greater than 0.7 and 0.9.

Species	**Nº of cells** B I > 0.6	**% of area** BI > 0.6	**Nº of cells** B I > 0.7	**% of area** BI > 0.7
B. bison	4222	71.33	2560	43.25
B. bubalus	4009	67.73	1929	32.59
C. schreberi	3818	64.50	1947	32.89
C. hispanus	4258	71.94	2061	34.82
C. lunaris	4104	69.34	2491	42.08
E. fulvus	3865	65.30	1901	32.12
E. amyntas	3902	65.92	2254	38.08
O. fracticornis	4224	71.36	2737	46.24
O. furcatus	3882	65.59	1745	29.48
O. lemur	3915	66.14	2058	34.77
O. opacicollis	3741	63.20	1721	29.08
O.similis	3893	65.77	1963	33.16
O. taurus	4016	67.85	1955	33.03
O. vacca	3941	66.58	1854	31.32

Figure S1: Main steps for calculating Biogeographical Ignorance values.

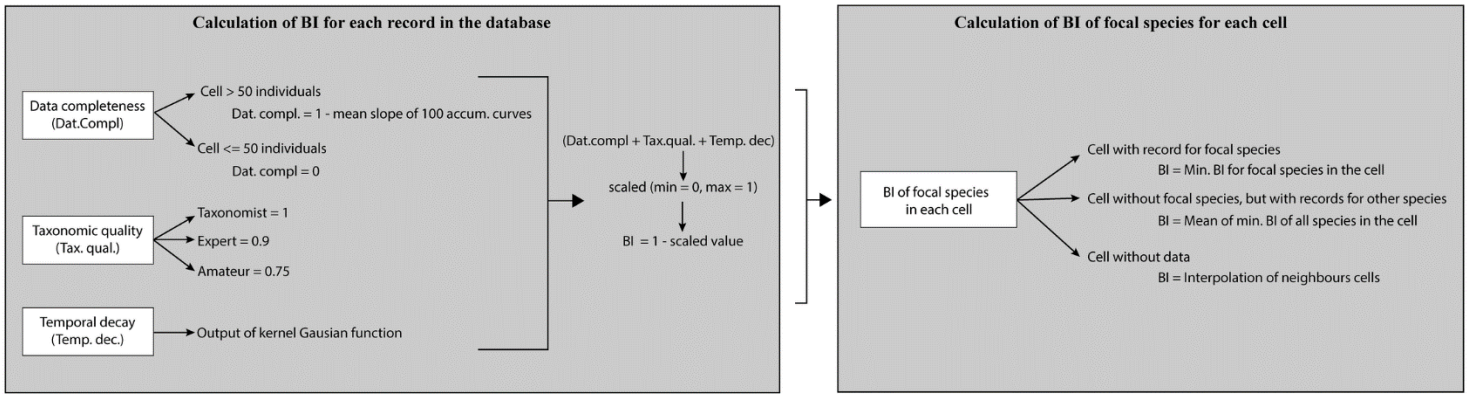
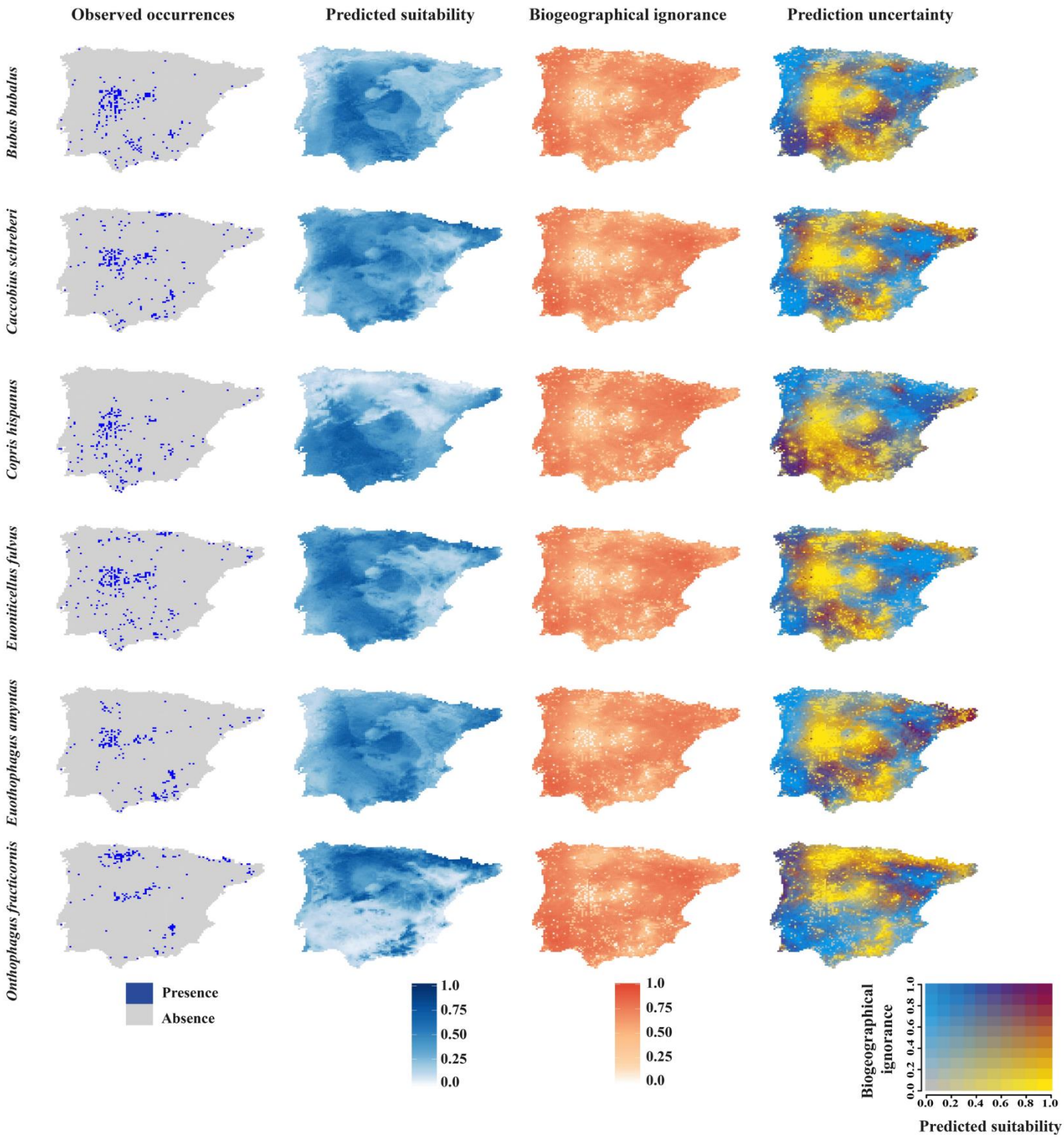


Figure S2: Observed occurrences from BANDASCA database, suitability predicted by the ensemble SDM technique, Maps of Biogeographical Ignorance and prediction uncertainty for each species studied.



Cont. Figure S2.

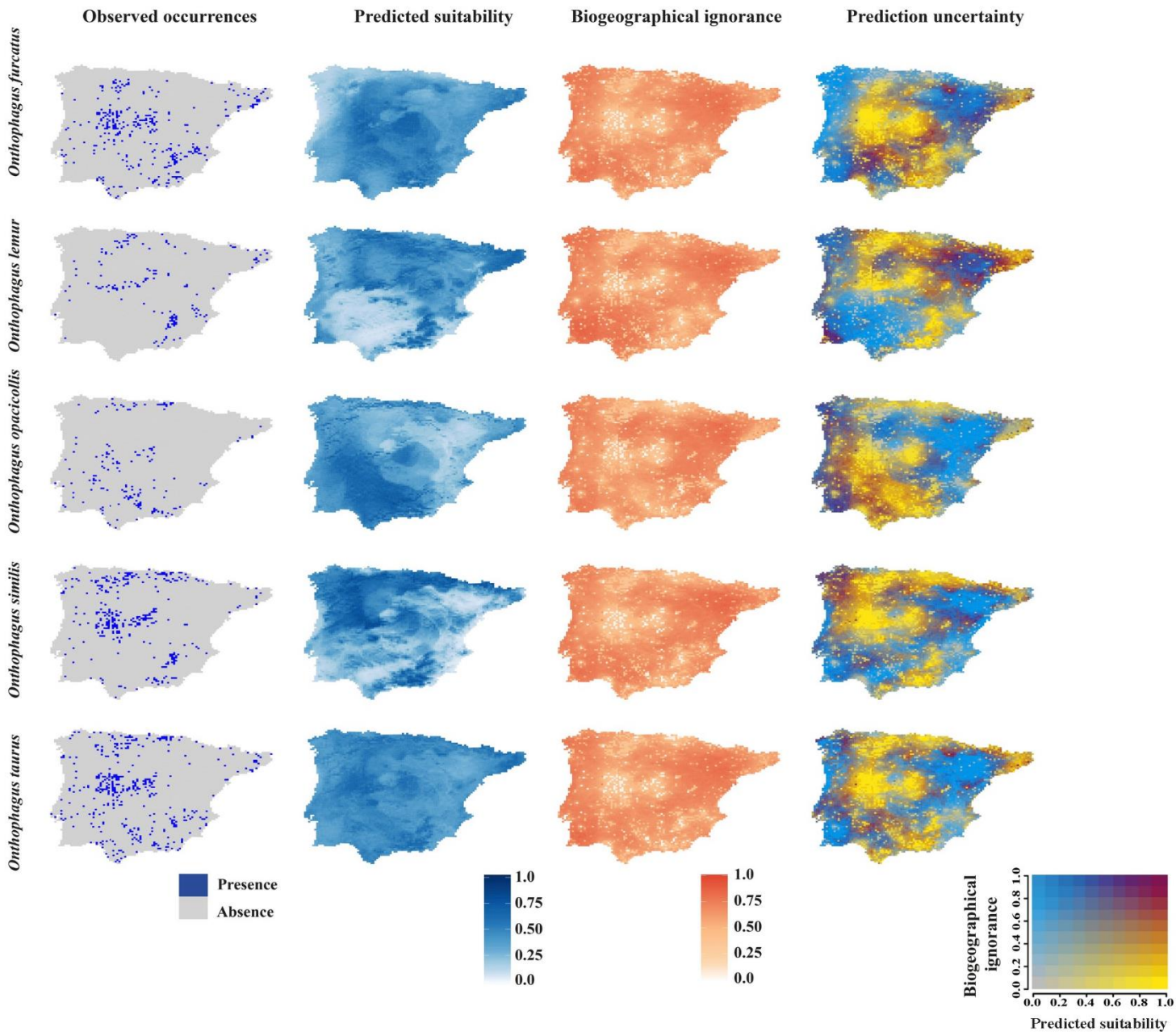
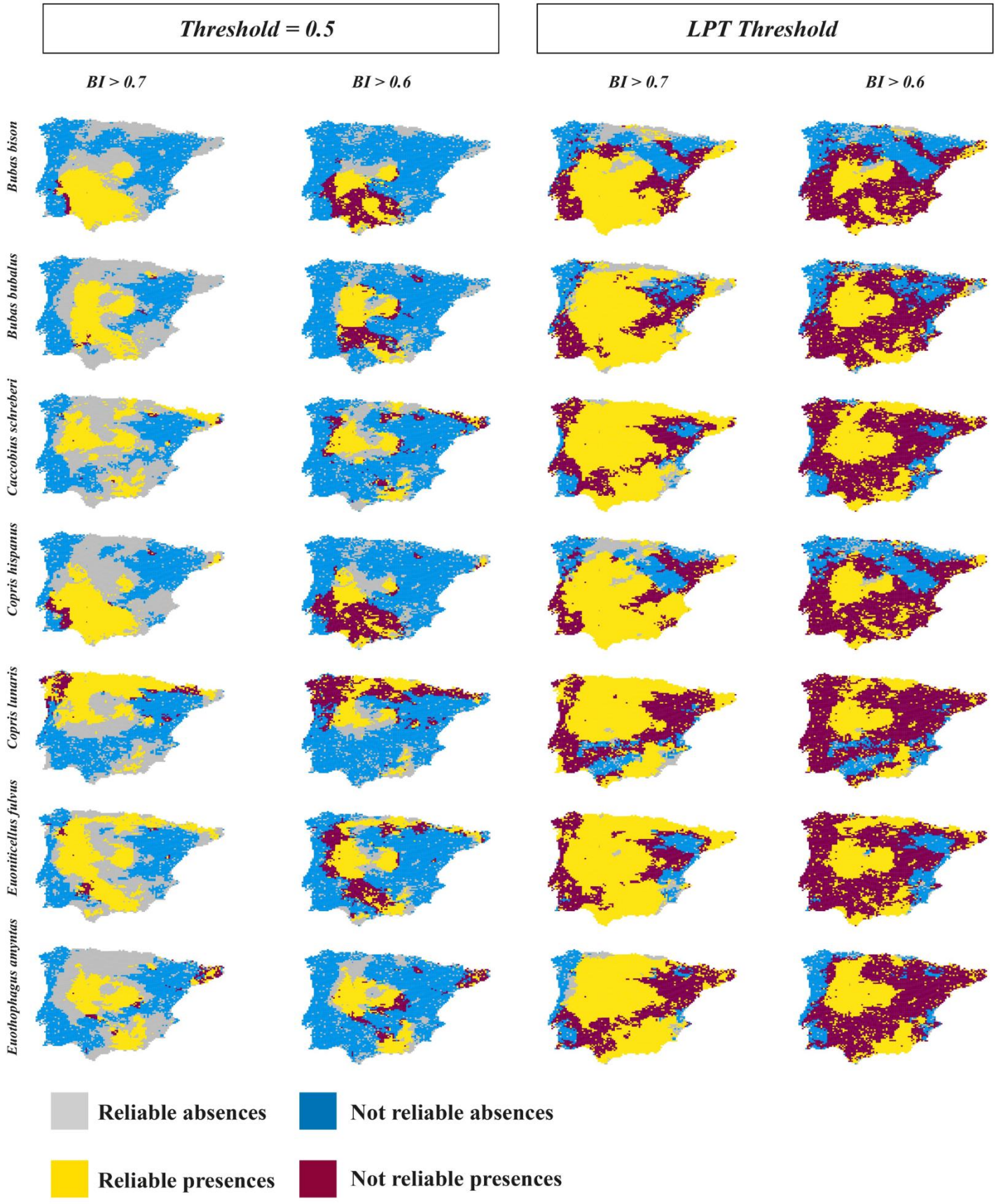


Figure S3: Binary uncertainty maps generated by choosing an ignorance threshold of 0.9 and 0.7 and two ensemble suitability thresholds (0.5 and LPT) for each species.



Cont. Figure S3.

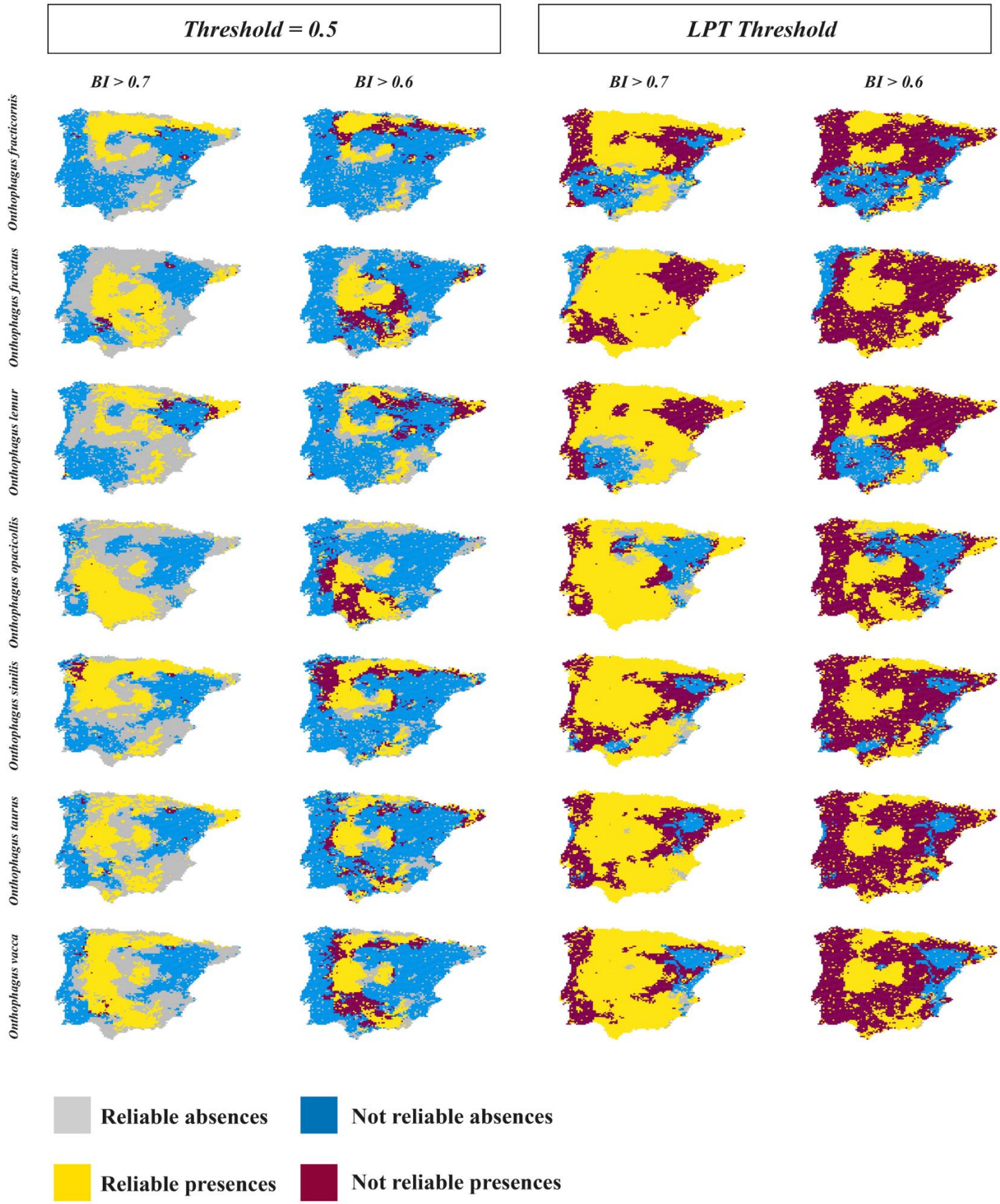


Figure S4: Density curves of BI values generated by different configurations of completeness calculation.

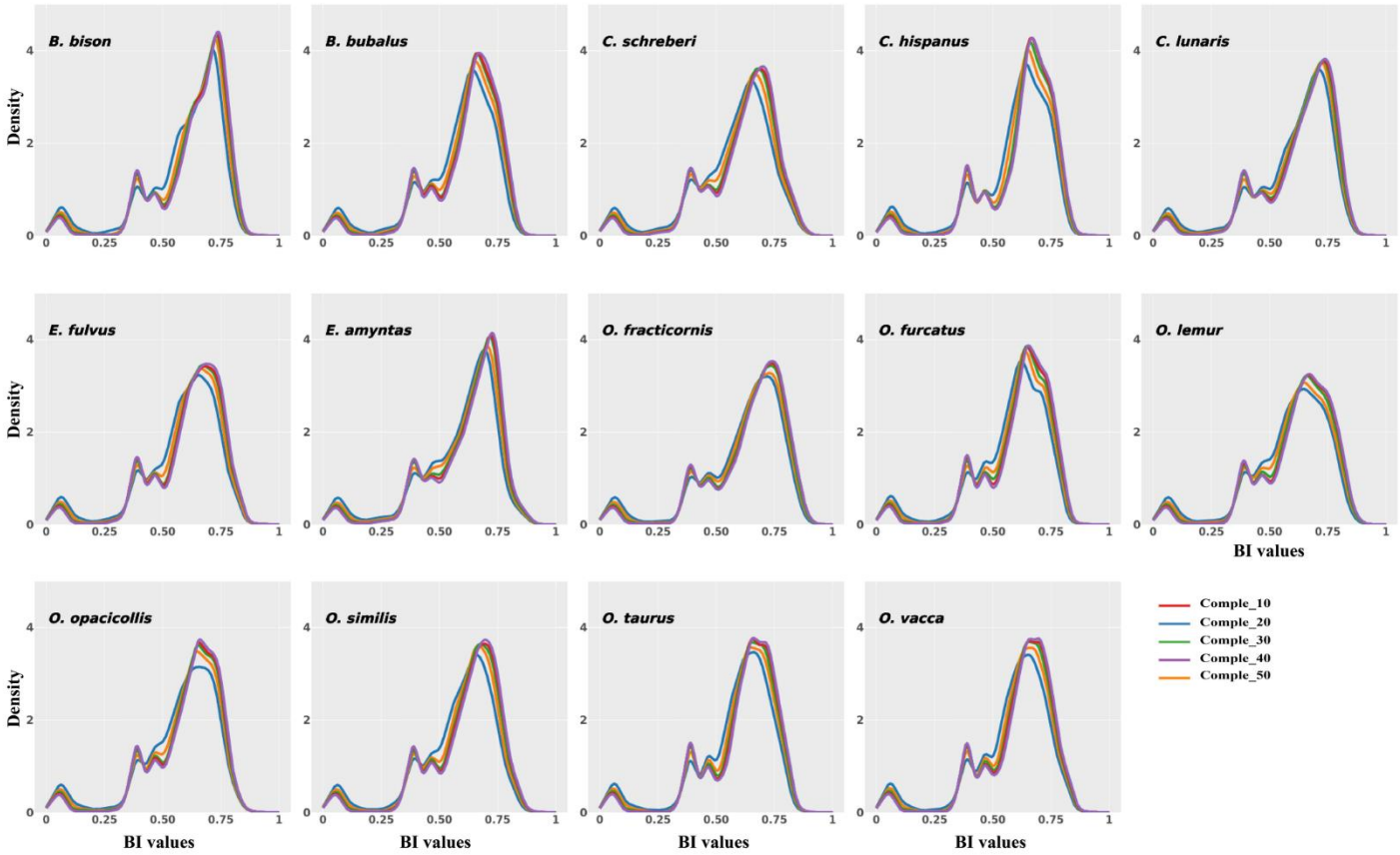


Figure S5: Coefficient of variation of BI values generated by different configurations of completeness calculation.

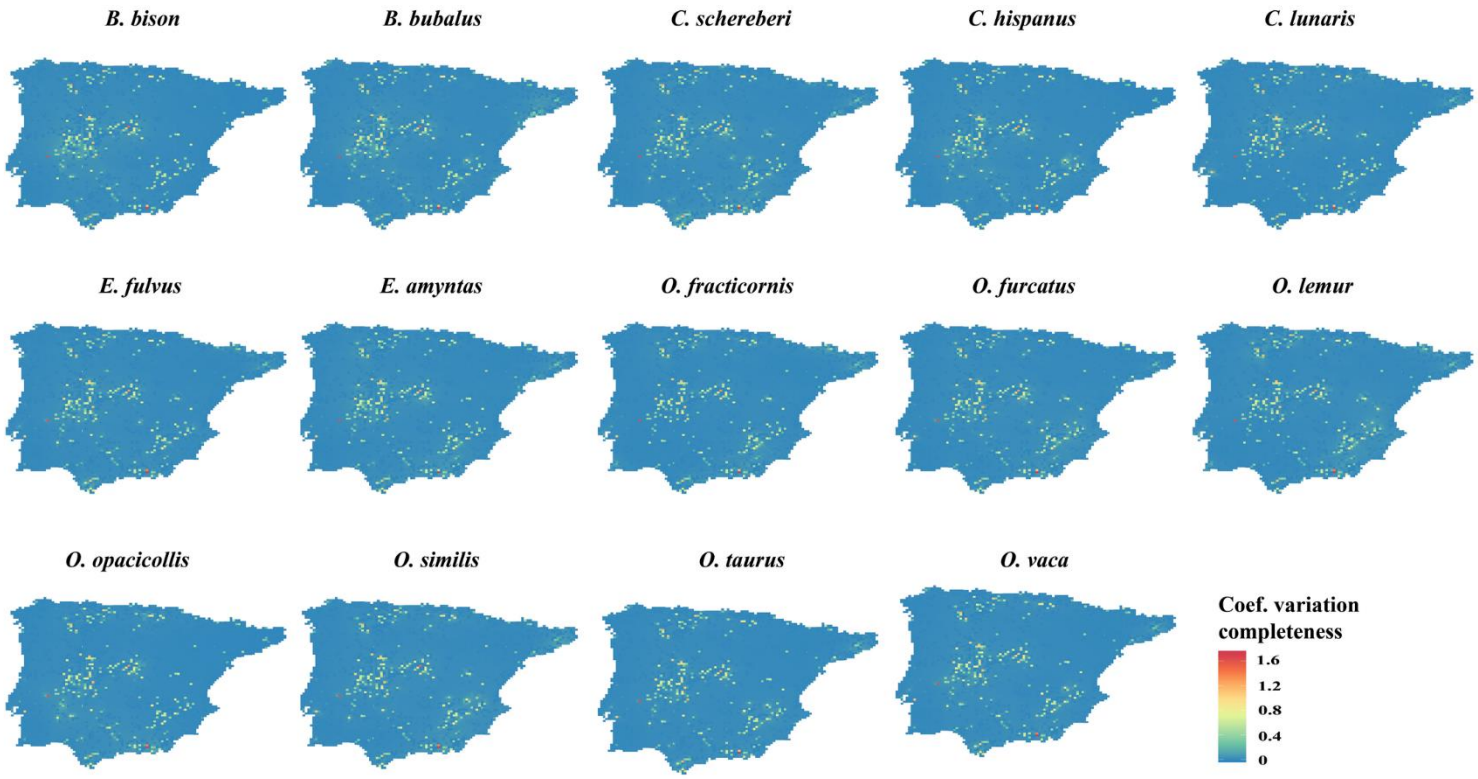


Figure S6: Density curves of BI values generated by different configurations of taxonomic decay curves.

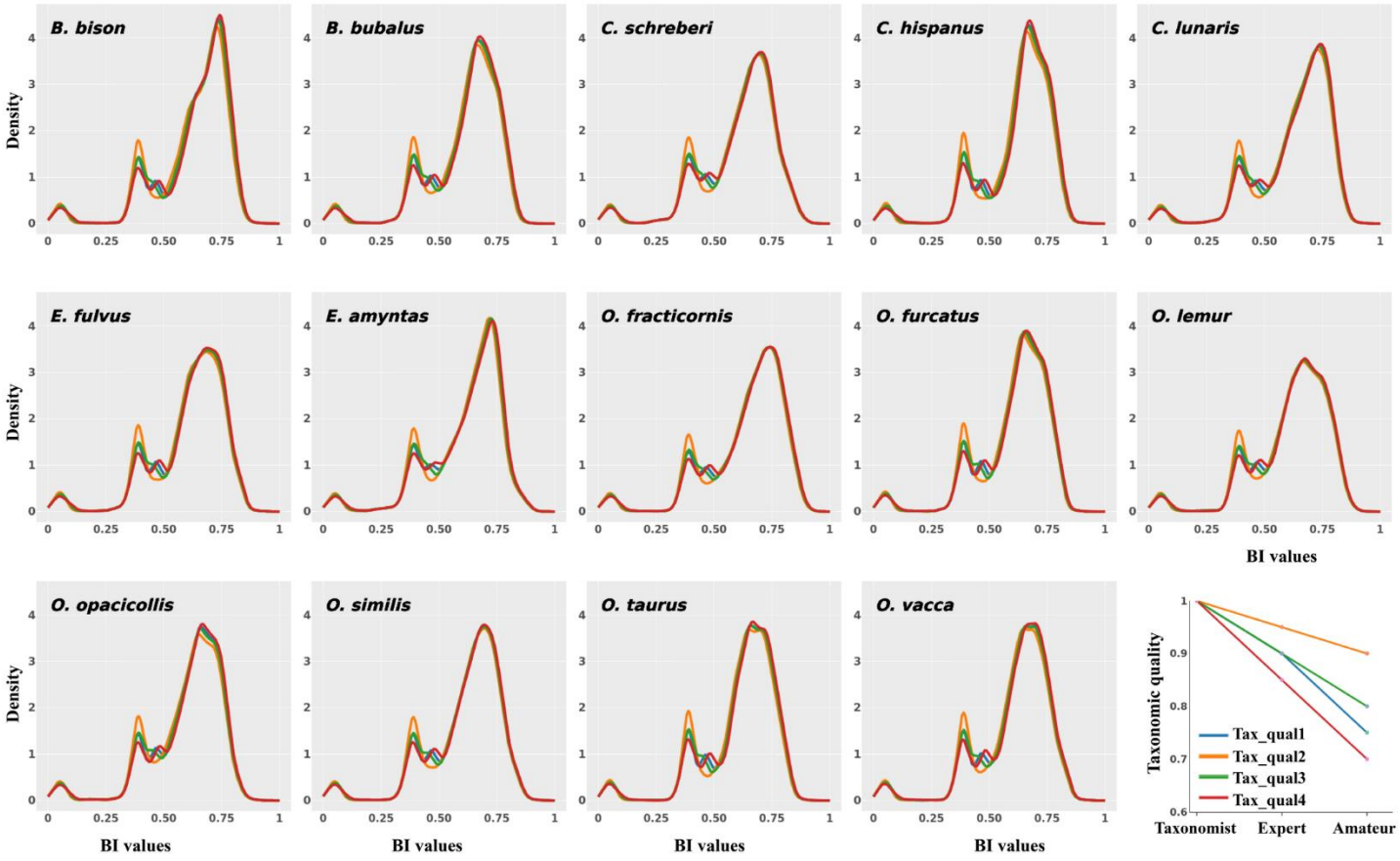


Figure S7: Coefficient of variation of BI values generated by different configurations of taxonomic decay curves.

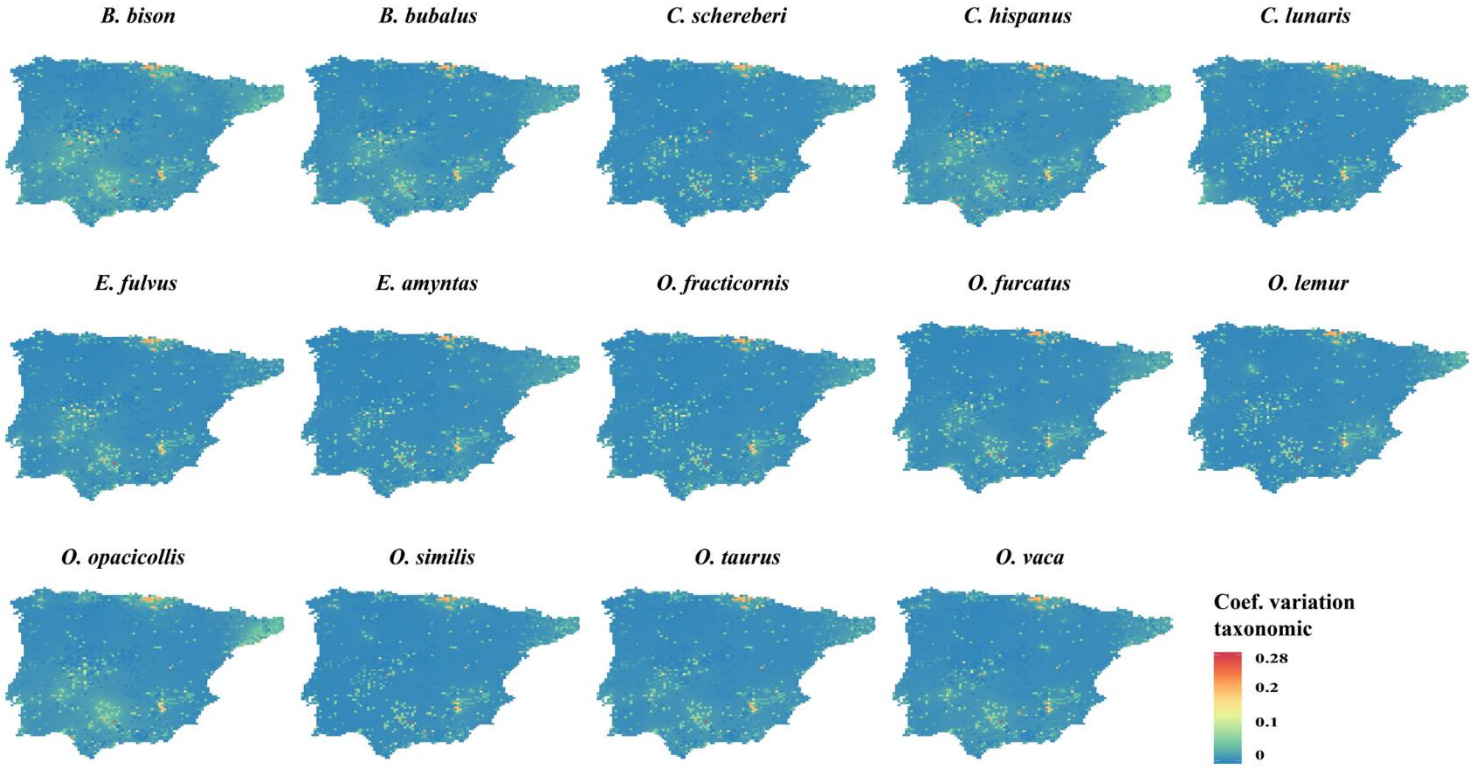


Figure S8: Density curves of BI values generated by different configurations of temporal decay curves.

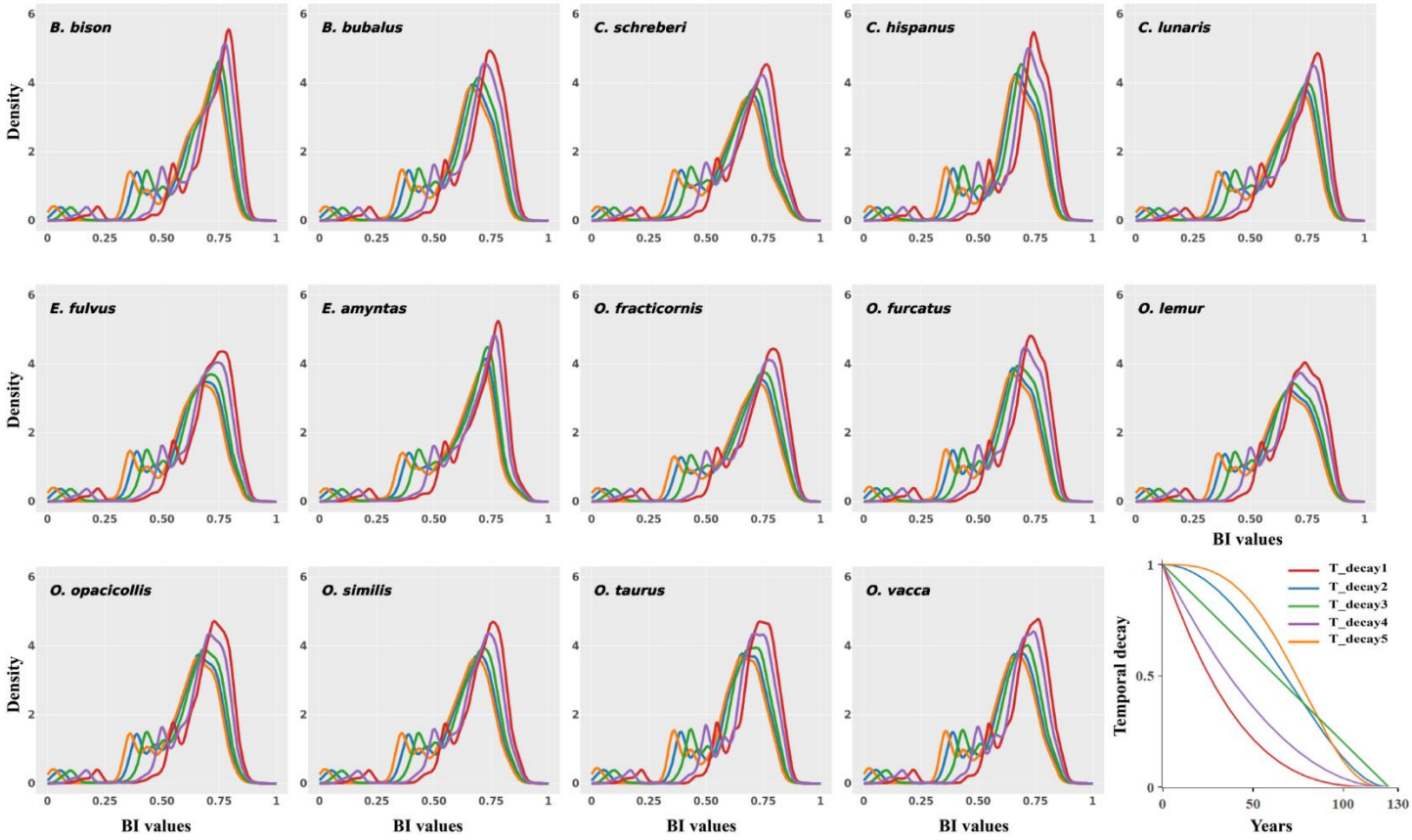


Figure S9: Coefficient of variation of BI values generated by different configurations of temporal decay curves.

.

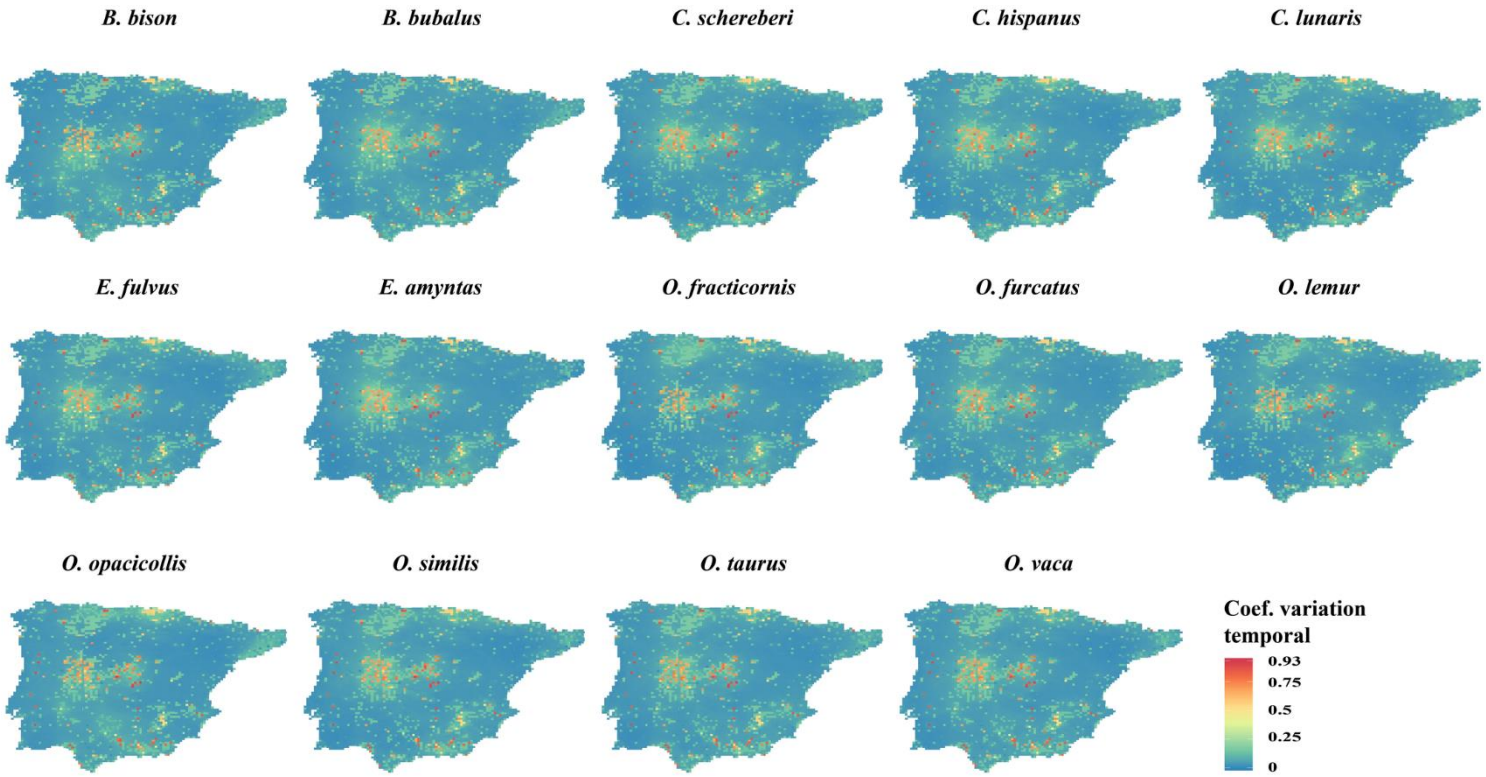


Figure S10: Density curves of BI values generated by different configurations of spatio-environmental matrix weight.

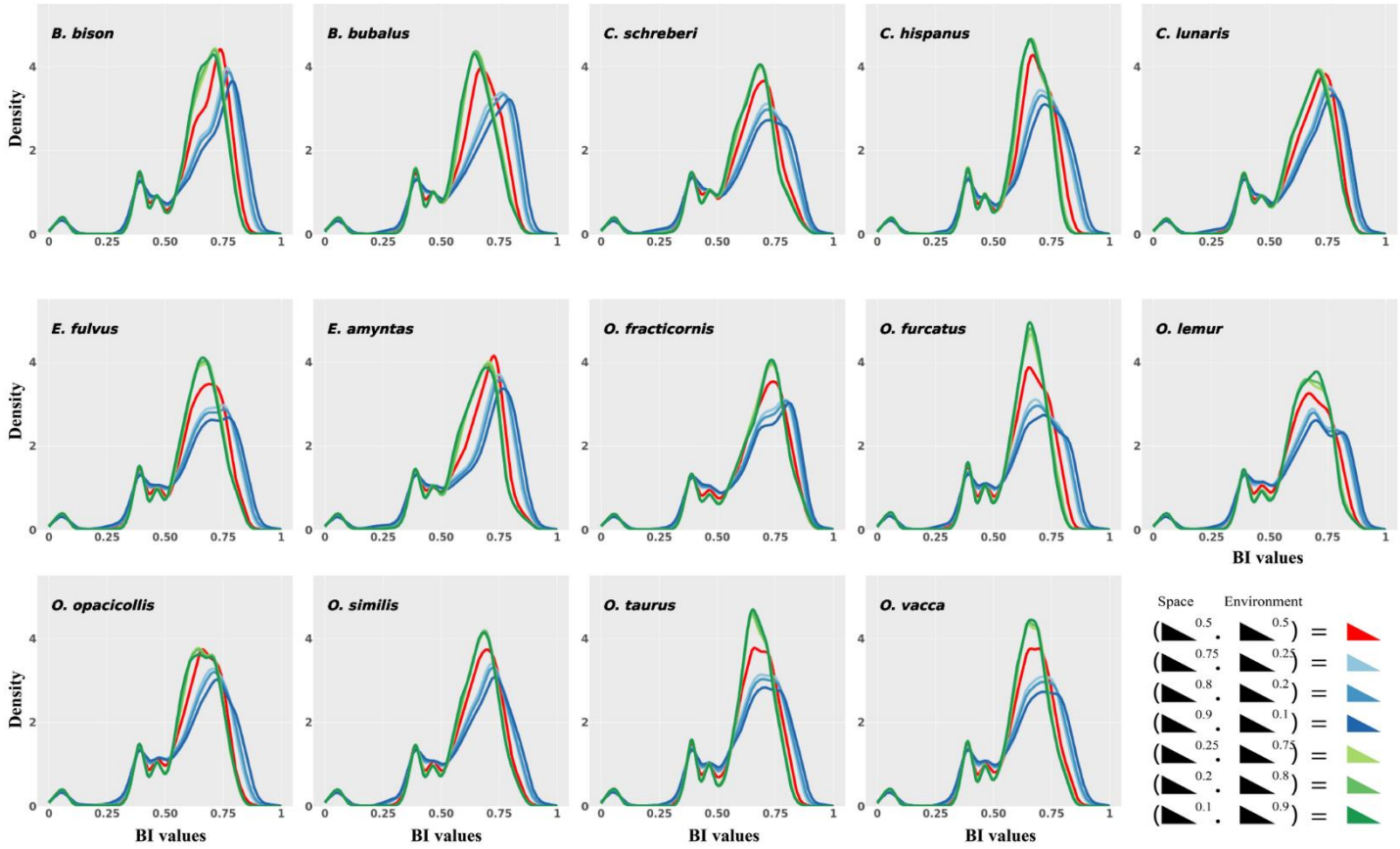
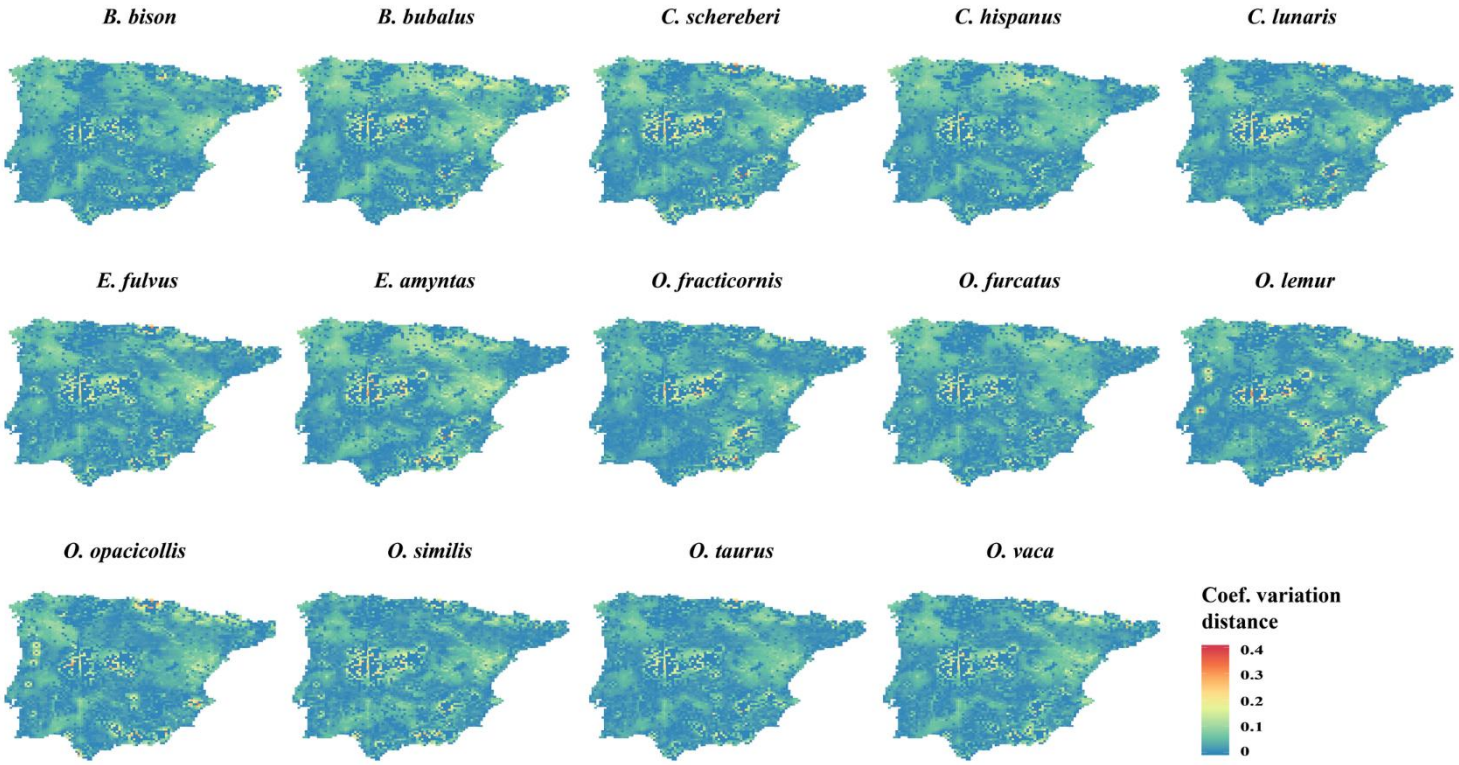


Figure S11: Coefficient of variation of BI values generated by different configurations of spatio-environmental matrix weight.



Appendix S1 - R-script for calculate values of Biogeographical ignorance.

Required packages

```
library(sqldf)
library(spaa)
library(vegan)
library(dplyr)
library(cluster)
library(fields)
library(geosphere)
library(scales)
library(data.table)
```

Necessary data/information

- **data** = database with (in this order): cell id, species name, number of individuals, year of collection,identifier quality (1 - 3).
- **nr** = minimum value of individuals in a cell to the cell be selected as a well sampled cell and calculation of completeness curves.
- **year** = year of analysis, if not specified the year is the one in R.
- **Var** = table with (in this order): cell id, longitude,latitude and environmental variables that will be used to calculate the spatio-temporal distance.

PART 1: Prepare data and load functions of each component

Import data and set up some parameters

```
data <- read.table("bandasca_to_BI.txt", h = T, as.is = T)
var <- read.table("Env_data_to_BI.txt", h = T, as.is = T)
nr <- 50
options(scipen = 999)
```

Load function to calculate spatio-environmental distance and maximum distance of correlation (Both will be used in the interpolation)

```
spt_env_dist <- function(var, wspt = 0.5, wenv = 0.5) {
  # Var  = table with (in this order): cell id, longitude,latitude and environmental variables that will be used to calculate the spatio-temporal distance
  # wspt = exponential weighting to be used in the space distance matrix
  # wenv = exponential weighting to be used in the environmental distance matrix

  # Environmental distance

  env_dist <- daisy(var[, 4:ncol(var)], metric = "euclidean", stand = T)

  # Spatial distance

  spt_dist <- distm(var[, 2:3], var[, 2:3], fun = distHaversine) # distance in meters

  # Set columns and row names

  spt_dist <- as.matrix(spt_dist)
  env_dist <- as.matrix(env_dist)
  rownames(env_dist) <- var[, 1]
  colnames(env_dist) <- var[, 1]
```

```
  rownames(spt_dist) <- var[, 1]
  colnames(spt_dist) <- var[, 1]

  # Rescale distance matrices to (0,1)

  env_stand <- rescale(env_dist, to = c(0, 1))
  spt_stand <- rescale(spt_dist, to = c(0, 1))

  # Create spatio-temporal matrix
  # Here the spatio-temporal matrix is created using equally exponential weighting (0.5)
  # for space and environmental distance matrix.
  # Note that weight can be set up according need to give more weight for one of the matrix.

  spt.envir.dist <- (spt_stand^wspt) * (env_stand^wenv)

  # Calculate mantel correlogram to find max distance of correlation to be used in the inter
polation

  spt_dist2 <- as.dist(spt_dist, diag = T, upper = FALSE)
  env_dist2 <- as.dist(env_dist, diag = T, upper = FALSE)

  correl <- mantel.correlog(env_dist2, spt_dist2) # this can take a long time...ZzzzZZzZz

  max.dist <- correl$mantel.res[(which(correl$mantel.res[, 3] < 0)[1] - 1), 1] # find maximu
m distance of correlation

  res <- list(spt_dist, max.dist, spt.envir.dist)
  names(res) <- c("spt_dist", "max.dist", "spt.envir.dist")
  return(res)
}
```

Load function to calculate species accumulation curves (SAC)

```
# Accumulation curves are generated using the data for the entire taxonomic group
# Slope values are diminished from 1 so that values near 1 indicate high completeness (low i
gnorance) while values near zero indicate hight ignorance.

Accum_curve <- function(data, nr) {
  # data = database with (in this order): cell id, species name, number of individuals, year
 of collection,identifier quality (1 - 3).
  # nr   = minimum value of individuals in a cell for calculation of completeness curves.

  df <- data[complete.cases(data[, 3]), ] # remove NA
  df2 <- df[, c(1:3)] # Selec the needed columns (id_cell, species and abundance)
  colnames(df2) <- c("id_cell", "species", "abundance") # set column names

  # Identify the cell of each indivudal

  b <- df2[, 3]
  df2 <- df2[rep(1:nrow(df2[, 1:2]), b), ]
  df[, 3] <- 1
  df2$abundance <- as.integer(df2$abundance)
  df2$id <- seq(1:nrow(df2))

  # Count the number of individuals in each cell and select only cell with > nr (50) individ
uals

  wsc <- fn$sqldf("select id_cell, count(species) as N_rec from df2 group by id_cell having
```

```
count(species) > `nr`")
  df3 <- merge(x = df2, y = wsc, by = "id_cell", all.y = TRUE) # Merge tables
  colnames(df3) <- c("cutm", "Species", "abundance", "id", "N_rec") # Set column names
  cell <- unique(df3$cutm)

  # create objects to receive results

  res1 <- list()
  specum_res <- list()
  slope_10 <- numeric(0)
  sp_remove <- numeric(0)

  # SAC and slope for each cell

  for (i in 1:length(cell)) { # for each cell

    dat2 <- df3[df3$cutm == cell[i], c("id", "Species", "abundance")] # subset data from uni
que cell
    if (length(unique(dat2[, 2])) <= 1) { # find cells with only one species
      sp_remove <- c(sp_remove, i) # these cell will be disregarded
      next
    }
    res1[[i]] <- data2mat(dat2)

    # Accumulation curves using specaccum

    specum_res[[i]] <- specaccum(res1[[i]], method = "random", permutations = 100)

    # 1- Slope 10% (calculate and invert slope values,so that higher values indicate higher
completeness)

    slope_10[[i]] <- 1 - (specum_res[[i]][[4]][length(specum_res[[i]][[4]])] - specum_res
[[i]][[4]][ceiling(length(specum_res[[i]][[4]]) * 0.9)]) / (length(specum_res[[i]][[4]]) - c
eiling(length(specum_res[[i]][[4]]) * 0.9))
  }

  slope_cells <- cbind(as.data.frame(cell, stringsAsFactors = F), as.data.frame(slope_10))

  # Remove cells with only one species and for which accumulation curve were not calculated
  slope_cells <- slope_cells[-sp_remove, ]

  # Save results

  write.table(slope_cells, "Completeness_SAC_10.txt", sep = "\t", row.names = F)
  return(slope_cells)
}
```

Load function to calculate taxonomic quality

```
# Values between 0 e 1 of identification quality. Values near 1 mean high quality (low ignor
ance) and 0 high ignorance.
# In our table the column "identifier quality", have values 1, 2 and 3, indicating respectiv
ely
# amateur, expert and taxonomist. Here lower quality is assigned to amateur identifiers (0.7
5),
#  followed by experts (0.9) and taxonomist (1).
#  Note that quality values and number of categories for identifiers can be set up different
ly.
```

```r
Tax_qual <- function(data) {
  tax.qual <- numeric(0) # vector to receive data quality

  # find cases for each identifier
  taxono <- which(data[, 5] == 3)
  expert <- which(data[, 5] == 2)
  amateur <- which(data[, 5] == 1)

  # Calculate values for each case

  tax.qual[taxono] <- 1
  tax.qual[expert] <- 0.9
  tax.qual[amateur] <- 0.75
  tax.qual <- data.frame(data$cutm, tax.qual) # join with cell id
  colnames(tax.qual) <- c("Id", "tax.qual")

  # Save results

  write.table(tax.qual, "Taxonomic_quality.txt", sep = "\t", row.names = F)

  return(tax.qual)
}
```

Load function to calculate temporal decay

```r
# year - Year of study or year of data utilization. If it is not specified the year of R sys
tem is used (year= auto). year=="auto" calculate the time passed from the year of data colle
ction
# until de year in which the analysis is done. Values near 1 indicate low ignorance, while v
alues near zero, high ignorance.

Temp.dec <- function(data, year = "auto") {
  if (year == "auto") { # use the year of R system.
    year <- as.numeric(format(Sys.Date(), "%Y"))
  }
  dif <- sapply(data[, 4], function(x, year) dif <- (year - x), year = year) # Calculate the
 number of years
  # since each record was collected.

  # Gausian

  h <- max(dif, na.rm = T) # find the maximum value of difference
  res1 <- sapply(dif, function(x, h) (1 - ((x / h)^2))^2, h = h) # Fits a kernel Gausian fun
ction with the data
  Temp_qual <- cbind.data.frame(data$cutm, res1)
  colnames(Temp_qual) <- c("id", "temporal_quality")
  write.table(Temp_qual, "Temporal_decay.txt", sep = "\t", row.names = F)

  return(Temp_qual)
}
```

PART 2: Calculating BI values

Calculate each component

```
spt.env.dist <- spt_env_dist(var, wspt = 0.5, wenv = 0.5)
completeness <- Accum_curve(data, nr = 50)
taxononomic_quality <- Tax_qual(data)
time_decay <- Temp.dec(data, year = 2020) # Here we are using year = 2020, the same used in the MoBIs' paper
```

Join id, species, taxonomic, temporal quality and completeness

```
df4 <- cbind.data.frame(data[, 1:2], taxononomic_quality[, 2], time_decay[, 2])

for (i in 1:nrow(completeness)) {
  indx <- which(df4[, 1] == completeness[i, 1])
  df4[indx, "completeness"] <- completeness[i, 2] # add completeness values
}

colnames(df4) <- c("ID", "species", "Taxonomic_quality", "Temporal_quality", "completeness_uncertainty") # set columns names
df4[is.na(df4)] <- 0 # For cells with no values for completeness, values are set to 0, to indicate higher ignorance on this component.
df4$biogeographic_certainty <- rowSums(df4[, c(3, 4, 5)]) # sum of the columns - BIOGEOGRAPHIC CERTAINTY (0 indicate low certainty and 1 high certainty, they will be inverted below)
```

Reescale BC values to vary from 0 to 1

```
df4$biogeographic_ignorance <- rescale(df4$biogeographic_certainty, to = c(0, 1), from = c(0, 3))
```

Invert values so that they became Biogeographic ignorance

```
df4$biogeographic_ignorance <- 1 - df4$biogeographic_ignorance
```

Subset 14 species with highest number of records

```
# Calculate number of cells the species are present
# Here we selected the 14 species with more records, for which MoBIs will be generated.
# However, you can generate MoBIs for all species or for other specified subset.

# Calculate number of cells a species is present

all_sp <- unique(data[, 2])
sp_presence <- data.frame(matrix(nrow = length(all_sp), ncol = 2)) # data.frame to receive results
colnames(sp_presence) <- c("species", "num_occurrences")

for (i in 1:length(all_sp)) {
  index2 <- which(data[, 2] == all_sp[i])
  sp_presence[i, 1] <- all_sp[i]
  sp_presence[i, 2] <- length(unique(data[index2, 1]))
}

sp_presence <- sp_presence[order(sp_presence$num_occurrences, decreasing = T), ]
sp_14 <- sp_presence[1:14, 1]
```

Generate and export BI values for each species

```
for (e in 1:length(sp_14)) { # for each one of the 14 selected species
  df5 <- data.frame(matrix(NA, length(var[, 1]), (ncol(df4)))) # Matrix to receive BI values
 for each cell
  df5[, 1] <- var[, 1]
  df5[, 2] <- "nsp"
  colnames(df5) <- colnames(df4)
  selec <- df4[which(df4[, 2] == sp_14[e]), ] # Select all records of the focal species
  cell_pre <- unique(selec[, 1]) # cells with the presence of the focal species
  rem <- which(df5[, 1] %in% cell_pre) #  Find and remove cell with records for the focal sp
ecies
  BI_cell_abse <- df5[-rem, ] # Select cell in which the focal species is ABSENT

  # Calculate BI values for cells in which the focal species is NOT PRESENT, but that have i
nformation
  # about other species of the taxonomic group. For these cells BI values is calculated as t
he mean of the
  # lowest BI values of each species in the cell.

  for (t in 1:nrow(BI_cell_abse)) { # calculate BI value for each cell in which the focal sp
ecies is ABSENT
    cur <- df4[which(df4[, 1] == BI_cell_abse[t, 1]), ] # Select all records in a cell in wh
ich the focal species is absent

    if (nrow(cur) != 0) { # Some cells have no records for any species, for these, BI values
 will be interpolated latter (see below)
      unique_sp <- unique(cur[, 2])
      min_BI <- numeric(0)
      for (y in 1:length(unique_sp)) { # Calculate the minimum BI value for each specie in t
he cell.
        sp_cur <- which(cur[, 2] == unique_sp[y]) # Find records for the species y in the ce
ll.
        minsp <- min(cur[sp_cur, 7]) # Select the minim BI value for the specie y.
        min_BI[y] <- minsp # Save minimum BI for species y
      }
      BI_cell_abse[t, 7] <- mean(min_BI) # Calculate the median of the minimum BI value of a
ll species in the cell t.
    }
  }

  BI_cell_abse <- BI_cell_abse[, c(1, 7)]
  BI_cell_abse <- BI_cell_abse[complete.cases(BI_cell_abse), ]
  colnames(BI_cell_abse) <- c("ID", "biogeographic_Ignorance")

  # Calculate the BI value for the cells in which the species is PRESENT. For these cells
  # the BI value is the minimum BI value of all records of the focal species in the cell.

  BI_cell_prese <- data.frame(matrix(NA, length(cell_pre), 2))
  colnames(BI_cell_prese) <- c("ID", "biogeographic_Ignorance")

  for (s in 1:length(cell_pre)) {
    focal_cell <- selec[which(selec[, 1] == cell_pre[s]), ]
    min_bi_fc <- min(focal_cell[, 7])
    BI_cell_prese[s, 1] <- cell_pre[s]
    BI_cell_prese[s, 2] <- min_bi_fc
  }

  # Join BI values for cells with the presence and absence of the species but with presence
of other species of the group
```

```r
  BI_cells_with_data <- as.data.frame(rbind(BI_cell_prese, BI_cell_abse))

  #### Interpolation

  # calculate BI values for cells without any occurrence, i.e., with no data.
  # For these cells BI values are calculated by IDW interpolation using as distance the spat
io-temporal matrix,
  # i.e., taking into account the geographic and environmental distance.
  # Interpolation uses BI values of the influential cells. The influential cells are those i
nside a radius of max.dist
  # defined by mantel correlogram. Interpolation uses the values of all cells in this radius,
 including those without data, for which
  # BI values as set to 1 i.e., maximum ignorance.

  BI_partial <- as.data.frame(matrix(NA, length(var[, 1]), 3)) # Matrix to receive the parti
al results
  colnames(BI_partial) <- c("id", "IgB", "complete") # set column names
  BI_partial[, 1] <- var[, 1] # id cells

  for (i in 1:length(BI_cells_with_data[, 1])) { # add BI values calculated for cells with d
ata (with and without the species presence)
    indr <- which(BI_partial[, 1] == BI_cells_with_data[i, 1])
    BI_partial[indr, 2] <- BI_cells_with_data[i, 2]
  }

  bi.interpol <- as.data.frame(matrix(NA, nrow(BI_partial), 2)) # Table to receive calculate
d and interpolated BI values.This is the final table.
  colnames(bi.interpol) <- c("id", "BI_IDW")
  bi.interpol[, 1] <- var[, 1]

  BI_partial$presence <- 0 # column to identify cells with the species presence, to give app
ropriated weight in interpolation.
  indt <- which(BI_partial[, 1] %in% cell_pre) # cells with species presence
  BI_partial[indt, 4] <- 1 # SEt 1 for cells with presence for the focal species


  p <- 2 # set parameter of weight of interpolation

  for (c in 1:nrow(BI_partial)) {
    if (is.na(BI_partial[c, 2]) == FALSE) { # if the cell already have a BI value calculated,
 add it to table

      bi.interpol[c, 2] <- BI_partial[c, 2] # add BI values already calculated (cells with a
ny data)
    } else { # If the cell has no BI value, then interpolate it

      indx2 <- which(spt.env.dist$spt_dist[c, ] <= spt.env.dist$max.dist & spt.env.dist$spt_
dist[c, ] != 0) # Using the spatial distance matrix, find the cells that are inside max.dist
ance
      cur <- data.frame(BI_partial[indx2, ], spt.env.dist$spt.envir.dist[indx2, c])

      # Set maximum BI values, i.e, high ignorance for influential cells with no data.
      cur[is.na(cur[, 2]), 2] <- 1

      if (length(cur[, 1]) > 0) {
        divt <- length(cur[, 1]) / 2 # Find number of cells in the radius that correspond to
 50% of the influential cells
        influ_prese <- which(cur[, 4] == 1) # Find influential cells in which the focal spec
```

```
ies is present
        numet <- 0
        denot <- 0

        if (length(influ_prese) < divt & length(influ_prese) > 0) { # if the influential cel
ls with the species presence
          # correspond to less than 50% of cells, a weight is set to make them to contribute
 to at least 50% fo the BI interpolated value
          wprese <- 0.5 / length(influ_prese) # set weight that will be given to each presen
ce cell.
          wabse <- 0.5 / (length(cur[, 1]) - length(influ_prese)) # set weight that will be
given to each absence cell.

          for (t in 1:length(cur[, 1])) {
            if (cur[t, 4] == 1) { # if the cell have a presence for the species set weight f
or presence in the interpolation

              nct <- (cur[t, 2] * wprese) / (cur[t, 5]^p)
              dct <- (wprese / (cur[t, 5]^p))
            } else { # if the cell have no presence for the species set weight for absence i
n the interpolation

              nct <- (cur[t, 2] * wabse) / (cur[t, 5]^p)
              dct <- (wabse / (cur[t, 5]^p))
            }
            numet <- numet + nct
            denot <- denot + dct
          }
        }
        if (length(influ_prese) > divt || length(influ_prese) == 0) { # if the presence cell
s already correspond for 50% of more of the influential cell
          # weight is not modified.The same occurs if there are only absence cells.
          for (t in 1:length(cur[, 1])) {
            nct <- (as.numeric(cur[t, 2]) / (cur[t, 5]^p))
            dct <- 1 / (cur[t, 5]^p)
            numet <- numet + nct
            denot <- denot + dct
          }
        }
        cur.interpol <- numet / denot
        bi.interpol[c, 2] <- cur.interpol
      } else { # if there is no cell in the radius of influence
        bi.interpol[c, 2] <- 1 # BI is set to 1, i.e. maximum ignorance
      }
    }
  }

  bi.interpol$BI_IDW <- round(bi.interpol$BI_IDW, 3) # round values
  BI_final <- data.frame(var[, 1:3], bi.interpol$BI_IDW)
  nome <- paste("BI_", sp_14[e], ".txt", sep = "")
  write.table(BI_final, nome, sep = "\t", row.names = F, quote = F)
}
```

Appendix S2: R-script for generate prediction uncertainty maps based on predicted suitability (SDM results) and Biogeographical ignorance values.

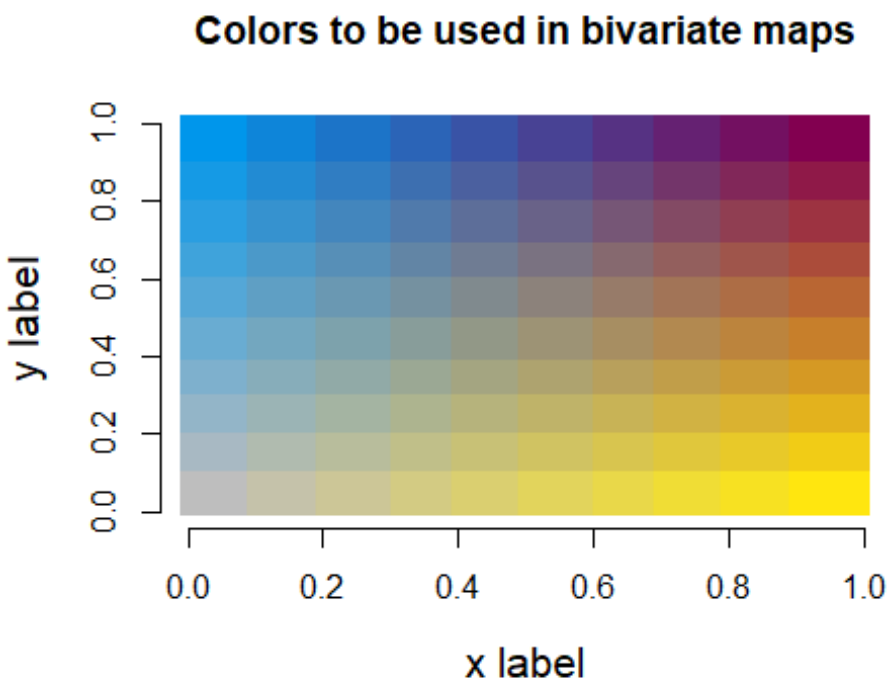
Required packages

```
library(classInt)
library(ggplot2)
library(ggpubr)
library(plyr)
library(rgdal)
library(scales)
library(dismo)
library(grid)
library(raster)
library(rgeos)
library(sp)
```

Load function to create bivariate colors

```
nquantiles=10 # colors are based on 10 quantiles

colmat<-function(nquantiles=10, upperleft=rgb(0,150,235, maxColorValue=255), upperright=rgb
(130,0,80, maxColorValue=255), bottomleft="grey", bottomright=rgb(255,230,15, maxColorValue=
255), xlab="x label", ylab="y label"){
  my.data<-seq(0,1,.01)
  my.class<-classIntervals(my.data,n=nquantiles,style="quantile")
  my.pal.1<-findColours(my.class,c(upperleft,bottomleft))
  my.pal.2<-findColours(my.class,c(upperright, bottomright))
  col.matrix<-matrix(nrow = 101, ncol = 101, NA)
  for(i in 1:101){
    my.col<-c(paste(my.pal.1[i]),paste(my.pal.2[i]))
    col.matrix[102-i,]<-findColours(my.class,my.col)}
  plot(c(1,1),pch=19,col=my.pal.1, cex=0.5,xlim=c(0,1),ylim=c(0,1),frame.plot=F, xlab=xlab,
ylab=ylab,cex.lab=1.3, main="Colors to be used in bivariate maps")
  for(i in 1:101){
    col.temp<-col.matrix[i-1,]
    points(my.data,rep((i-1)/100,101),pch=15,col=col.temp, cex=1)}
  seqs<-seq(0,100,(100/nquantiles))
  seqs[1]<-1
  col.matrix<-col.matrix[c(seqs), c(seqs)]}
col.matrix<-colmat(nquantiles=10)
```




```
colormatrix=col.matrix
```

Load species predicted suitability (SDM results)

```
pred_suit<-read.table ("./ensemble_Bubas_bison.txt", h=T)
colnames(pred_suit)<- c("id","Long","Lat","ense")
```

Load biogeographical ignorance values

```
bi<-read.table("./BI_Bubas_bison.txt", h=T)
colnames(bi)<- c("id","Long","Lat","BI_final")
```

Quantiles for predicted suitability

```
quanmean<-rescale(pred_suit$ense,to=c(0,1),from=c(0,1))
temp<-data.frame(quanmean, quantile=rep(NA, length(quanmean)))
brks<-with(temp, quantile(temp,na.rm=TRUE, probs = c(seq(0,1,1/nquantiles))))
r1<-within(temp, quantile <- cut(quanmean, breaks = brks, labels = 2:length(brks),include.lo
west = TRUE))
quantr<-data.frame(r1[,2])
```

Quantiles for Biogeographical ignorance

```
quanvar<-bi$BI_final
temp<-data.frame(quanvar, quantile=rep(NA, length(quanvar)))
brks<-with(temp, quantile(temp,na.rm=TRUE, probs = c(seq(0,1,1/nquantiles))))
r2<-within(temp, quantile <- cut(quanvar, breaks = brks, labels = 2:length(brks),include.low
est = TRUE))
quantr2<-data.frame(r2[,2])
```

Find bivariate color for each cell

```
as.numeric.factor<-function(x) {as.numeric(levels(x))[x]}
col.matrix2<-colormatrix
cols<-numeric(length(quantr[,1]))

for(j in 1:length(quantr[,1])){
  a<-as.numeric.factor(quantr[j,1])
  b<-as.numeric.factor(quantr2[j,1])
  cols[j]<-col.matrix2[b,a]
}

bivar<-as.data.frame(cbind(bi[,c(1:3)],cols))
bivar<-bivar[,c(1,4)]
colnames(bivar)<-c("id","color")
```

Join information about colors in polygon

```
# In this version polygon data are being used.However a version for
# create bivariate maps based on raster data can be found at:
# https://rfunctions.blogspot.com/2015/03/bivariate-maps-bivariatemap-function.html


shape<-readOGR(dsn=".",layer="malha_int")

## OGR data source with driver: ESRI Shapefile
## Source: "C:\Users\Geizi\Desktop\teste_script_BI_maps", layer: "malha_int"
## with 6056 features
## It has 17 fields
## Integer64 fields read as strings:  fid_1

shape.df<-fortify(shape,region="CODIGO")

shape.df = join(shape.df, bivar, by="id")
shape.df = shape.df[complete.cases(shape.df),]
```

Generate and plot prediction uncertainty maps

```
p_bivar <- ggplot(shape.df) +
  aes(long,lat,group=group,fill= color) +
  geom_polygon() +
  coord_equal()+
  scale_fill_identity()+
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.key = element_blank(),
        panel.border = element_blank(),
        legend.title = element_blank(),
        axis.ticks = element_blank(),
        legend.position="none")+
  ggpubr::theme_transparent()+
  theme(plot.margin = unit(c(0,0,0,0), "cm"))

p_bivar
```