# Hosts, parasites, and their interactions respond to different climatic variables

Timothée Poisot Cynthia Guéveneux-Julien Marie-Josée Fortin
Dominique Gravel Pierre Legendre

This document will collect the data online, and perform all of the analyses presented in the main document. Figures in the paper were generated by compiling this document.

## Getting the data and setting up the work environment

### Loading the required packages

```
library(devtools)

install_github("mangal-wg/rmangal")
library(rmangal)

install_github("poisotlab/betalink")
library(betalink)

library(igraph)
library(plyr)
library(vegan)
library(dismo)
library(rgeos)
library(maps)
library(mapdata)
library(maptools)
library(cluster)
```

### Getting the interaction data from `mangal.io`

In this next section, we will query the data from `mangal.io`, and format them as `igraph` objects. This step can be relatively long, depending on your local network speed. The first time we run this script, it will download the required data and save them. The next runs, however, it will load directly from the rdata file. We will use this approach throughout this file – the first run will be the longest.

```
if(file.exists("networks.Rdata")){
  load("networks.Rdata")
} else {
  api <- mangalapi(url="http://mangal.io", v="v1")
  dataset <- getDataset(api, 4)

  # Get lattitude and longitude
  get_lat_lon <- function(x){getNetwork(api, x)[c('name', 'latitude', 'longitude')]}
```

```
  coordinates <- data.frame(aaply(dataset$networks, 1, get_lat_lon))

  # Get networks themselves as igraph objects
  networks <- alply(dataset$networks, 1, function(x) toIgraph(api, x))

  # Save for future use
  coordinates$name <- as.vector(coordinates$name)
  coordinates$latitude <- as.numeric(as.vector(coordinates$latitude))
  coordinates$longitude <- as.numeric(as.vector(coordinates$longitude))
  names(networks) <- coordinates$name
  save(coordinates, networks, file="networks.Rdata")
}
```

We will now aggregate all networks into a regional-level metaweb:

```
continental <- metaweb(networks)
```

There are 326 species and 1945 interactions in the continental network, which is an aggregation of `length(networks)` local observations.

In the next step, we will get some basic informations about the local-scale networks, *i.e.* their number of hosts, parasites, and number of interactions:

```
getInfos <- function(G) {
    c(
        hosts = sum(degree(G, mode="in") > 0),
        paras = sum(degree(G, mode="out") > 0),
        int = length(E(G))
    )
}

net_inf <- ldply(networks, getInfos)

summary(net_inf$hosts)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3.00    7.50   11.00   12.24   16.00   27.00
```

```
summary(net_inf$paras)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    7.00   15.00   19.00   20.47   25.50   40.00
```

```
summary(net_inf$int)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12.00   40.00   63.00   78.12  104.00  226.00
```

We will also have a look at the occurrence data – specifically, how many unique species/location pairs there are, how many species occur exactly once, and how many species occur at more than 10 locations:

```
occ <- sort(table(unlist(llply(networks, function(x) V(x)$name))))

sum(occ == 1)
```

```
## [1] 94
```

```
sum(occ > 10)
```

```
## [1] 43
```

```
length(occ)
```

```
## [1] 326
```

## Download additional files for analysis

Some of this analyses relies on files that are not part of any R package. We will download them directly. All of the functions we download come from the webpage of Pierre Legendre.

```
archive_file <- "http://adn.biol.umontreal.ca/~numericalecology/labo/fonctions_r/beta-diversit

# Download the file
store_in <- "./beta_div_archive.zip"
download.file(archive_file, store_in)
unzip(store_in)

# Load the files
r_files_to_load <- list.files(path="beta-diversity",
  pattern=".R", full.names = TRUE)
for(r_file in r_files_to_load){source(r_file)}
```

## Download climatic data

The last step is to download the climatic variables, that will serve as predictor variables throughout the analysis. This can be done directly from within R.

```
if(file.exists("climate.Rdata")) {
    load("climate.Rdata")
} else {
    xl <- range(coordinates$longitude) + c(-3, 3)
    yl <- range(coordinates$latitude) + c(-3, 3)
    if(file.exists("bioclim.grd")){
        cat("\n")
    } else {
        bioclim <- crop(
          getData("worldclim", var="bio", res=5),
          c(min(xl), max(xl), min(yl), max(yl))
        )
        writeRaster(bioclim, filename="bioclim.grd", overwrite=T)
    }
    bclim <- brick("bioclim.grd")
    climate <- extract(bclim, coordinates[,c('longitude', 'latitude')])
    row.names(climate) = coordinates$name
    maps_climate <- bclim
    save(climate, maps_climate, file="climate.Rdata")
}
```

The climate matrix can be scaled – although this is not necessary, and the unscaled version will therefore be used throughout.

```
scaled_climate <- scale(climate)
```

Once these steps are done, we can start the actual analyses.

# Analyses

## Network beta-diversity

The first step will be to measure the $\beta$-diversity of networks (Poisot et al. 2012) – specifically, we will focus on $\beta'_{OS}$, which is a measure of the difference between the network of *potential* and *realized* interactions within a location.

```
# Beta OS prime
osp <- betalink::beta_os_prime(networks)

hist(osp, xlim=c(0, 1), xlab="Interaction dissimilarity",
  freq=T, main="", xaxs='i', yaxs='i',
  border="lightgrey", col="lightgrey"
)

axis(1)
axis(2)
```

```
# We also generate the figure for the paper
# This will be done in all chunks
dev.copy2pdf(file="../figures/osprime.pdf")
```

```
## png
##   2
```

```
summary(osp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.06422 0.17220 0.26760 0.26700 0.34580 0.56360
```

The values of $\beta'_{OS}$ (where a value of 0 indicates the presence of all potential interactions, and values closer to 1 indicate a decreasing number of potential interactions) range from 0.0642202 to 0.5636364, which indicates that the networks show a range of variation – there is some local filtering of potential interactions.

## LCBD of hosts, parasites, and interactions

The first step of this analysis is to measure the LCBD (location contribution to beta-diversity; Legendre & De Cáceres (2013)) of all 51 locations, for hosts, parasites, and potential as well as realized interactions.

The first step is to create community data matrices for hosts and parasites. This requires a few new functions for automation, specifically to get the names of hosts, and parasites.

```
getHosts <- function(x) {
  el <- get.edgelist(x)[,2]
  return(unique(el))
}

getParasites <- function(x) {
  el <- get.edgelist(x)[,1]
  return(unique(el))
}

y_matrix <- function(webs, f){
  all_items <- llply(webs, f)
  items_u <- unique(unlist(all_items))
  Y_items <- matrix(0, ncol=length(items_u), nrow=length(webs))
```

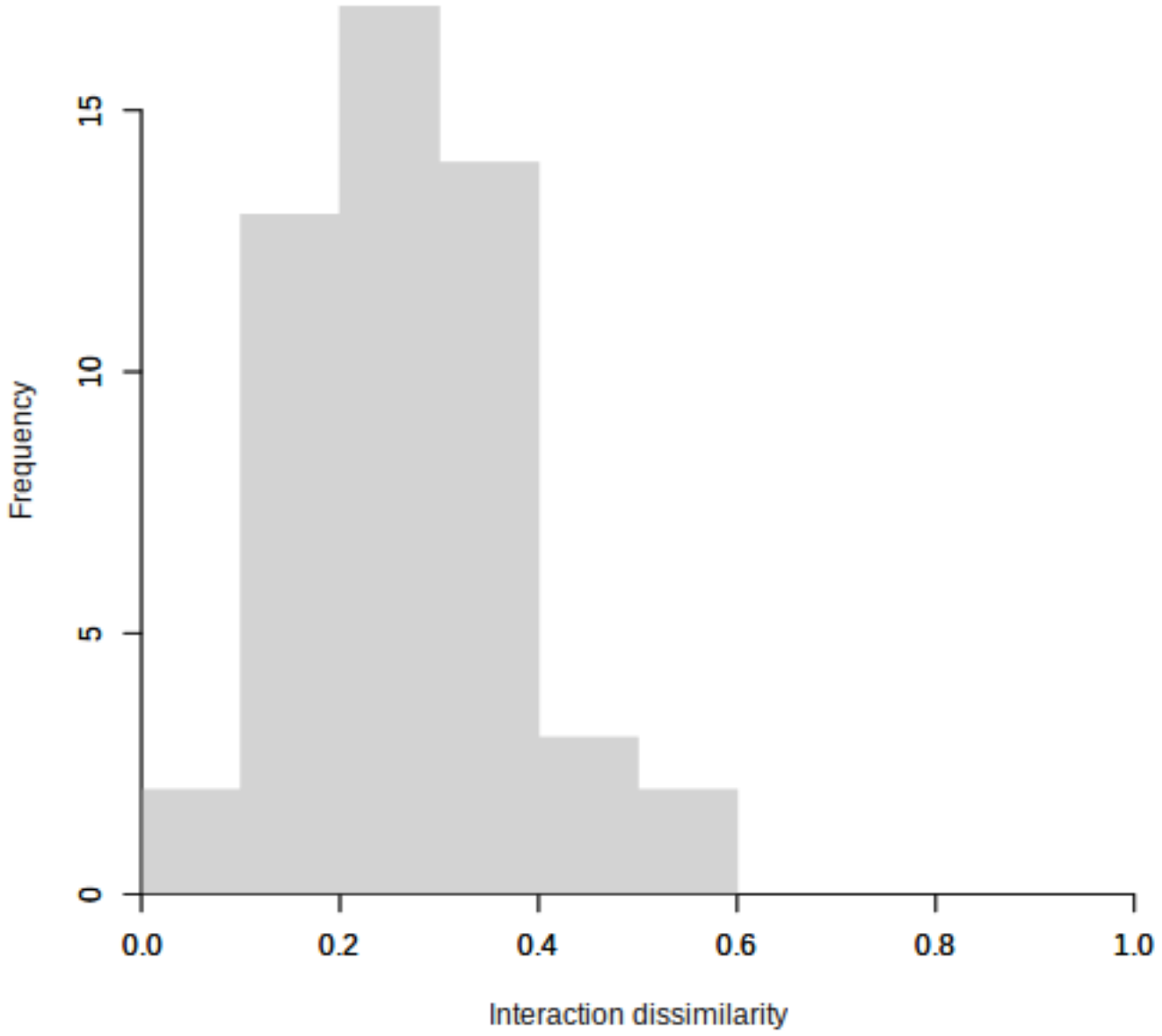


Figure 1: Distribution of Beta OS' values. Values close to 0 indicate that all potential interactions are realized, and values close to 1 indicate that almost all potential interactions are lost.

```
  rownames(Y_items) <- names(all_items)
  colnames(Y_items) <- items_u
  for(i in c(1:length(all_items))) {
    Y_items[i,all_items[[i]]] <- 1
  }
  return(Y_items)
}

Y_host <- y_matrix(networks, getHosts)
Y_para <- y_matrix(networks, getParasites)
```

We now need to do the same thing for the interactions. Potential interactions can be inferred from the metaweb, knowing the list of species:

```
potential_networks <- networks
for(i in c(1:length(potential_networks))) {
  potential_networks[[i]] <- induced.subgraph(
    continental,
    which(V(continental)$name %in% V(networks[[i]])$name)
  )
}

getInteractions <- function(x) {
  el <- get.edgelist(x)
  return(unique(aaply(el, 1, paste, collapse="-")))
}

Y_local <- y_matrix(networks, getInteractions)
Y_poten <- y_matrix(potential_networks, getInteractions)
```

We can now measure the beta diversity of the four different matrices, including the LCBD:

```
if(file.exists("betadiversity.Rdata")){
    load("betadiversity.Rdata")
} else {
    # These two parameters can be changed to chose the
    # number of permutations and the distance measure.
    n_perm <- 9999
    DIST_MEASURE <- "hellinger"
    BD_local <- beta.div(Y_local , DIST_MEASURE , nperm=n_perm , save.D=T)
    BD_poten <- beta.div(Y_poten , DIST_MEASURE , nperm=n_perm , save.D=T)
    BD_host <- beta.div(Y_host   , DIST_MEASURE , nperm=n_perm , save.D=T)
    BD_para <- beta.div(Y_para   , DIST_MEASURE , nperm=n_perm , save.D=T)
    save(BD_local, BD_poten, BD_host, BD_para, file="betadiversity.Rdata")
}
```

We can summarize this information – the first column is the $\beta$ diversity of the community data matrix, and the second column is the number of locations that are identified as having a significant contribution to $\beta$-diversity.

| matrix | $\beta$-diversity | nb. sign. locations |
|---|---|---|
| **H** | 0.808 | 8 |
| **P** | 0.798 | 12 |
| **R** | 0.943 | 23 |
| **Q** | 0.906 | 25 |

## Using beta-diversity to cluster locations

We will now look at the overall structure of the data, using hierarchical clustering. This is done by partitioning around medoids on the $\beta$-diversity distance matrix, and selecting the number of medoids that yield the smallest silhouette width.

```
getClusters <- function(bd) {
    # We only test between 2 and 10 clusters
    rng <- c(2:10)
    wid <- numeric(length(rng))
    # For each number of cluster ...
    for (i in c(1:length(rng))) {
        # We find the best clustering
        temp <- pam(bd$D, k=rng[i])
        # And measure its silhouette width
        wid[i] <- temp$silinfo$avg.width
    }
    # The 'correct' number of clusters is the one with
    # the smallest average silhouette
    k <- rng[which.min(wid)]
    ids <- pam(bd$D, k=k)
    return(ids$clustering)
}

plot_cluster <- function(coordinates, bd)
{
    ICOL <- "cornsilk"
    group <- getClusters(bd)
    signif <- c("white", "black")[as.numeric(bd$p.LCBD<0.05)+1]
    size <- bd$LCBD*80
    XR <- range(coordinates$longitude)+c(-3, 3)
    YR <- range(coordinates$latitude)+c(-3, 3)
    plot(XR, YR, mar=par("mar"),
         xlab=NA, ylab=NA, xaxt="n",
         yaxt="n", type="n",
         mar=c(0.05, 0.05, 0.05, 0.05)
        )
    rect(par("usr")[1],par("usr")[3],par("usr")[2],par("usr")[4], col="lightblue")
    map("world",
        xlim=XR,
        ylim=YR,
        fill=T, col=ICOL,
        border=ICOL,
        add=T
       )
    points(latitude~longitude, coordinates,
           bg=signif, pch=21+group, cex=size)
    box()
}

la <- matrix(c(1, 2, 3, 4), nrow=2, byrow=T)
layout(la, c(9, 9), c(2.6, 2.6))

plot_cluster(coordinates, BD_host)
text(
  min(coordinates$longitude), max(coordinates$latitude),
  "Hosts", cex=1.5, pos=4, font=2, offset=-0.5
)
```

```
plot_cluster(coordinates, BD_para)
text(
  min(coordinates$longitude), max(coordinates$latitude),
  "Parasites", cex=1.5, pos=4, font=2, offset=-0.5
)

plot_cluster(coordinates, BD_local)
text(
  min(coordinates$longitude), max(coordinates$latitude),
  "Local", cex=1.5, pos=4, font=2, offset=-0.5
)

plot_cluster(coordinates, BD_poten)
text(
  min(coordinates$longitude), max(coordinates$latitude),
  "Potential", cex=1.5, pos=4, font=2, offset=-0.5
)
```

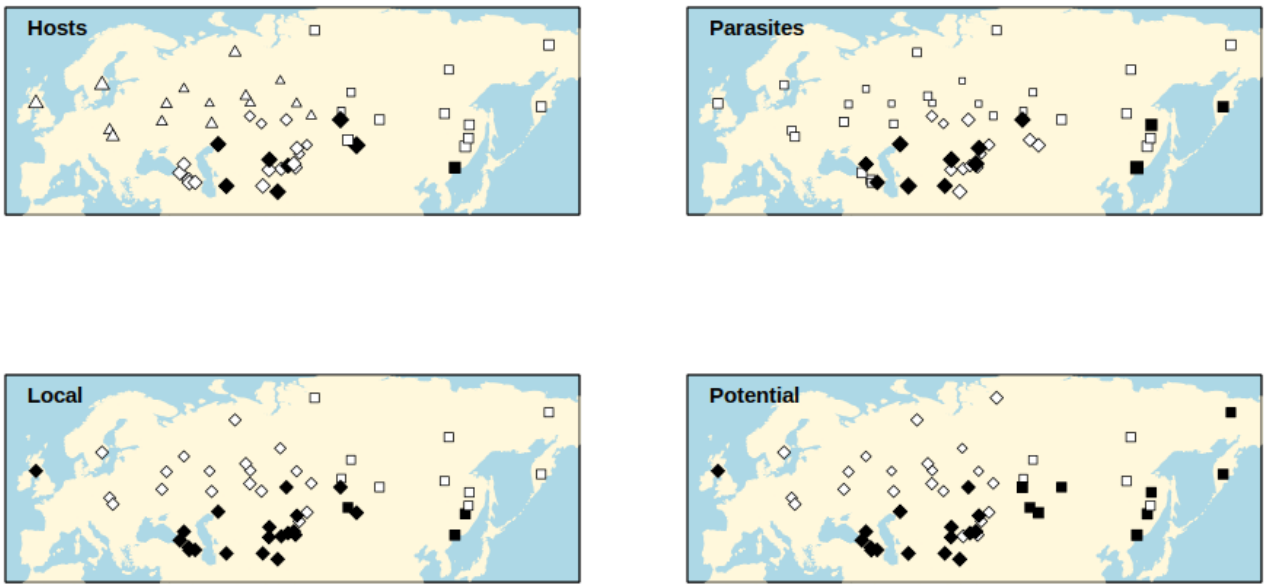


Figure 2: plot of chunk get_clusters_from_beta

```
dev.copy2pdf(file="../figures/clusters.pdf", width=13)
```

```
## png
##   2
```

This approach identifies three clusters for hosts, and two for parasites and interactions. The dataset seems to have a strong north-south structure.

## Impact of climatic variables

We will start by forward-selecting the variables that go into the model. If the file `capscale.Rdata` is already present, then we skip this (typically time-consuming) step.

```
fwsel <- function(bd) {
    int <- capscale(bd$D~1, data=data.frame(climate))
    all <- capscale(bd$D~., data=data.frame(climate))
    hstep <- ordistep(int, scope=formula(all), direction='forward', trace=F)
    return(hstep)
}


if(file.exists("capscale.Rdata")) {
        load("capscale.Rdata")
} else {
    permax <- 9999

    host_ordistep <- fwsel(BD_host)

    para_ordistep <- fwsel(BD_para)

    local_ordistep <- fwsel(BD_local)

    poten_ordistep <- fwsel(BD_poten)

    save(host_ordistep, para_ordistep,
         local_ordistep, poten_ordistep,
        file="capscale.Rdata")
}
```

Then we create plots for each of the four models.

```
plot_ordination <- function(ordi, bd)
{
    bgcol <- rgb(0.4, 0.4, 0.4, alpha=0.1)
    group <- getClusters(bd)
    signif <- c("white", "black")[as.numeric(bd$p.LCBD<0.05)+1]
    size <- bd$LCBD*80
    plot(ordi, type="n", ylim=c(-2.2, 2.2), xlim=c(-2.2, 2.2))
    symbols(x=0, y=0, circles=1, inches=F, add=T, bg=bgcol, fg=NA)
    points(ordi, pch=20+group, bg=signif, cex=bd$LCBD*100)
    points(latitude~longitude, coordinates, bg=signif, pch=21+group, cex=size)
    text(ordi, dis="cn")
    box()
}

par(mfrow=c(2,2))

plot_ordination(host_ordistep, BD_host)
title("Hosts")

plot_ordination(para_ordistep, BD_para)
title("Parasites")

plot_ordination(local_ordistep, BD_local)
title("Local")

plot_ordination(poten_ordistep, BD_poten)
title("Potential")
```

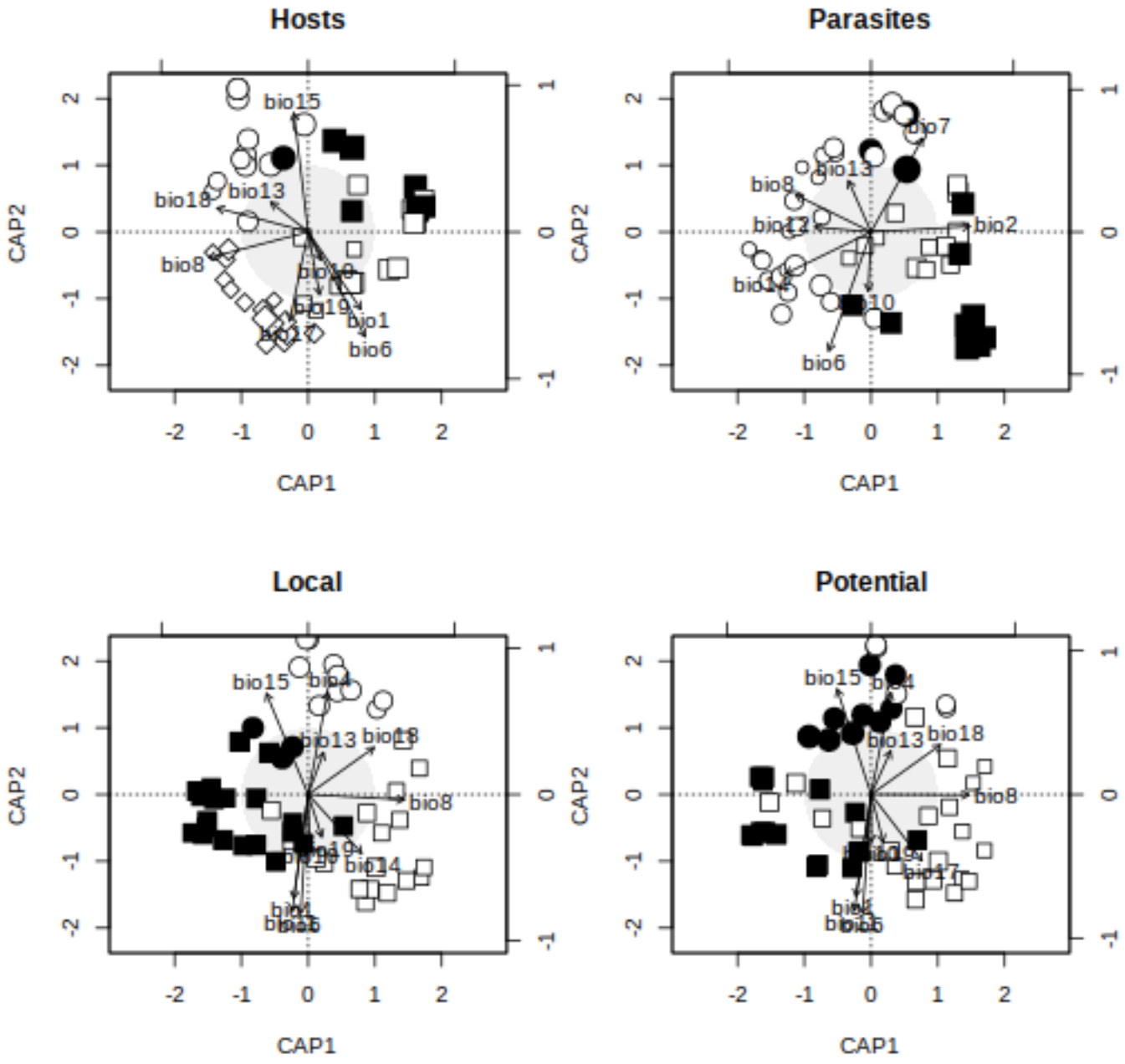


Figure 3: plot of chunk make_plots_ordi

```
dev.copy2pdf(file="../figures/ordinations.pdf", width=10, height=10)
```

```
## png
##   2
```

One this is done, we typically want a bit more informations about each of the four models – notably the number of contrained *vs.* unconstrained axes.

```
host_ordistep
```

```
## Call: capscale(formula = bd$D ~ bio6 + bio8 + bio1 + bio13 + bio10
## + bio15 + bio18 + bio19 + bio17, data = data.frame(climate))
##
##               Inertia Proportion Rank
## Total         40.3858     1.0000
## Constrained   17.0906     0.4232    9
## Unconstrained 23.2952     0.5768   41
## Inertia is squared Euclidean distance
##
## Eigenvalues for constrained axes:
##  CAP1  CAP2  CAP3  CAP4  CAP5  CAP6  CAP7  CAP8  CAP9
## 4.904 4.014 1.990 1.653 1.227 1.131 1.081 0.618 0.471
##
## Eigenvalues for unconstrained axes:
##   MDS1   MDS2   MDS3   MDS4   MDS5   MDS6   MDS7   MDS8
## 2.4125 1.9172 1.5595 1.3014 1.2844 1.1448 1.0938 0.9867
## (Showed only 8 of all 41 unconstrained eigenvalues)
```

```
para_ordistep
```

```
## Call: capscale(formula = bd$D ~ bio6 + bio2 + bio8 + bio10 + bio13
## + bio7 + bio14 + bio12, data = data.frame(climate))
##
##               Inertia Proportion Rank
## Total         39.8829     1.0000
## Constrained   14.4621     0.3626    8
## Unconstrained 25.4208     0.6374   42
## Inertia is squared Euclidean distance
##
## Eigenvalues for constrained axes:
##  CAP1  CAP2  CAP3  CAP4  CAP5  CAP6  CAP7  CAP8
## 4.036 3.589 2.194 1.634 1.170 0.818 0.566 0.455
##
## Eigenvalues for unconstrained axes:
##   MDS1   MDS2   MDS3   MDS4   MDS5   MDS6   MDS7   MDS8
## 2.5106 2.0422 1.6100 1.4495 1.3455 1.2141 1.1037 1.0693
## (Showed only 8 of all 42 unconstrained eigenvalues)
```

```
local_ordistep
```

```
## Call: capscale(formula = bd$D ~ bio6 + bio8 + bio1 + bio15 + bio18
## + bio10 + bio4 + bio11 + bio14 + bio13 + bio19, data =
## data.frame(climate))
##
##               Inertia Proportion Rank
## Total         47.1315     1.0000
## Constrained   15.4096     0.3269   11
## Unconstrained 31.7220     0.6731   39
## Inertia is squared Euclidean distance
##
```

```
## Eigenvalues for constrained axes:
##   CAP1   CAP2   CAP3   CAP4   CAP5   CAP6   CAP7   CAP8   CAP9  CAP10
## 2.7364 2.4985 1.6559 1.5166 1.2872 1.2358 1.1749 0.9337 0.8488 0.8015
##  CAP11
## 0.7203
##
## Eigenvalues for unconstrained axes:
##   MDS1   MDS2   MDS3   MDS4   MDS5   MDS6   MDS7   MDS8
## 1.9581 1.6041 1.4985 1.2646 1.1996 1.1461 1.1071 1.0338
## (Showed only 8 of all 39 unconstrained eigenvalues)
```

```
poten_ordistep
```

```
## Call: capscale(formula = bd$D ~ bio6 + bio8 + bio1 + bio13 + bio10
## + bio15 + bio18 + bio19 + bio17 + bio4 + bio11, data =
## data.frame(climate))
##
##               Inertia Proportion Rank
## Total         45.2924     1.0000
## Constrained   17.5685     0.3879   11
## Unconstrained 27.7239     0.6121   39
## Inertia is squared Euclidean distance
##
## Eigenvalues for constrained axes:
##  CAP1  CAP2  CAP3  CAP4  CAP5  CAP6  CAP7  CAP8  CAP9 CAP10 CAP11
## 3.665 3.062 2.183 1.707 1.504 1.273 1.200 0.952 0.763 0.669 0.590
##
## Eigenvalues for unconstrained axes:
##   MDS1   MDS2   MDS3   MDS4   MDS5   MDS6   MDS7   MDS8
## 1.9442 1.6723 1.6361 1.4175 1.1923 1.1326 1.0549 1.0287
## (Showed only 8 of all 39 unconstrained eigenvalues)
```

The last step is to identify which climatic variables were retained during the forward-selection step. They are presented in the table below according to their rank, *i.e.* bio6 having a value of one for all four matrices means it was the first selected variable in all four models.

```
varnames <- paste("bio", c(1:19), sep='')
variables <- matrix(0, ncol=4, nrow=19)
rownames(variables) <- varnames
colnames(variables) <- c("host", "parasite", "local", "potential")
variables <- as.data.frame(variables)

variables[labels(para_ordistep$terms), "parasite"] <- c(1:length(labels(para_ordistep$terms)))
variables[labels(host_ordistep$terms), "host"] <- c(1:length(labels(host_ordistep$terms)))
variables[labels(local_ordistep$terms), "local"] <- c(1:length(labels(local_ordistep$terms)))
variables[labels(poten_ordistep$terms), "potential"] <- c(1:length(labels(poten_ordistep$terms


# All variables
variables
```

```
##       host parasite local potential
## bio1     3        0     3         3
## bio2     0        2     0         0
## bio3     0        0     0         0
## bio4     0        0     7        10
## bio5     0        0     0         0
## bio6     1        1     1         1
## bio7     0        6     0         0
## bio8     2        3     2         2
```

```
## bio9     0        0     0         0
## bio10    5        4     6         5
## bio11    0        0     8        11
## bio12    0        8     0         0
## bio13    4        5    10         4
## bio14    0        7     9         0
## bio15    6        0     4         6
## bio16    0        0     0         0
## bio17    9        0     0         9
## bio18    7        0     5         7
## bio19    8        0    11         8
```

```
# Variables with no effect
variables[rowSums(variables)==0,]
```

```
##       host parasite local potential
## bio3     0        0     0         0
## bio5     0        0     0         0
## bio9     0        0     0         0
## bio16    0        0     0         0
```

# References


**Legendre & De Cáceres**. (2013). Beta diversity as the variance of community data: dissimilarity coefficients and partitioning. *Ecol Lett.* 16:951–63.

**Poisot et al.** (2012). The dissimilarity of species interaction networks. *Ecology Letters.* 15:1353–61.