Achatina analysis

Prof. Dr. Bruno Vilela - UFBA | IBio

31-05-2021

Before starting

Prior to the analyzes, we recommend all users to check the latest version of R at http://www.r-project.org/ and to make sure that they are using the updated versions of their installed R packages. Users can automatically update their installed packages with the following code:

```
# Update installed packages
# update.packages(checkBuilt = TRUE, ask = FALSE)
```

The analysis presented here makes use of the following R packages available at CRAN. Use the following code to install them.

```
# Install packages
# install.packages("knitr")
# install.packages("spThin")
# install.packages("rgeos")
# install.packages("sp")
# install.packages("maptools")
# install.packages("raster")
# install.packages("ecospat")
# install.packages("here")
# install.packages("tidyverse")
# install.packages("rnaturalearth")
# install.packages("stars")
# install.packages("sf")
```

Once installed, load them.

```
# Load packages
library(knitr)
library(spThin)
library(rgeos)
library(sp)
library(maptools)
library(raster)
library(ecospat)
library(here)
library(tidyverse)
library(sf)
library(rnaturalearth)
library(stars)
library(readxl)
library(colorspace)
```

Data

Load the occurrence records

Load the occurrence records into the R environment.

```
occ.points1 <- read_excel(here("Data",
                               "achatinapontos.xls"))
occ.points2 <- read_excel(here("Data",
                               "eppsnativaspontos.xls"))
for (i in 2:10) {
occ.points2 <- bind_rows(occ.points2, read_excel(here("Data",
                               "eppsnativaspontos.xls"),
                         sheet = i))
}
colnames(occ.points2)[1] <- "Species"
occ.points <- distinct(bind_rows(occ.points1, occ.points2))
```

The loaded table includes 2604 occurrence records.

Thining occurrence records

The occurrence records gathered (see the methods section of the manuscript, for the description of how we obtained the data) are not free from geographical sample bias. To minimize this problem, we applied a thinning procedure using the **spThin** package to make sure that all the points have at least a minimum distance of 10km from each other (see Aiello-Lammens et al. 2014 for the algorithm description).

```
species <- unique(occ.points$Species)
occ.points.thin <- list()
for (i in 1:length(species)) {
  occ.points.thin[[i]] <- occ.points %>%
    filter(Species == species[i]) %>%
    thin(
      verbose = FALSE,
      lat.col = "Latitude",
      long.col = "Longitude",
      spec.col = "Species",
      thin.par = 10,
      reps = 1,
      write.files = FALSE,
      write.log.file = FALSE,
      locs.thinned.list.return = TRUE
    )
}
occ.points.thin <- lapply(occ.points.thin, "[[", 1)
names(occ.points.thin) <- species
```

After the thinning procedure the number of occurrence points is reduced for each species to n = Achatina fulica 586, elongatus 25, granulosus 19, haemastomus 17, musculus 22, paranaguensis 25, sanctipauli 29, yporanganus 17, dryades 31, intertextus 12, lorentzianus 24.

Define the regions to be tested

The first step is to define the number of groups (regions) to be tested. In the follow case we choose 3 groups.

```
n.groups <- 11
```

Now define the name of the groups, in the same geographical order of the groups, starting from the west to east. You can also define the codes to be used in the tables.

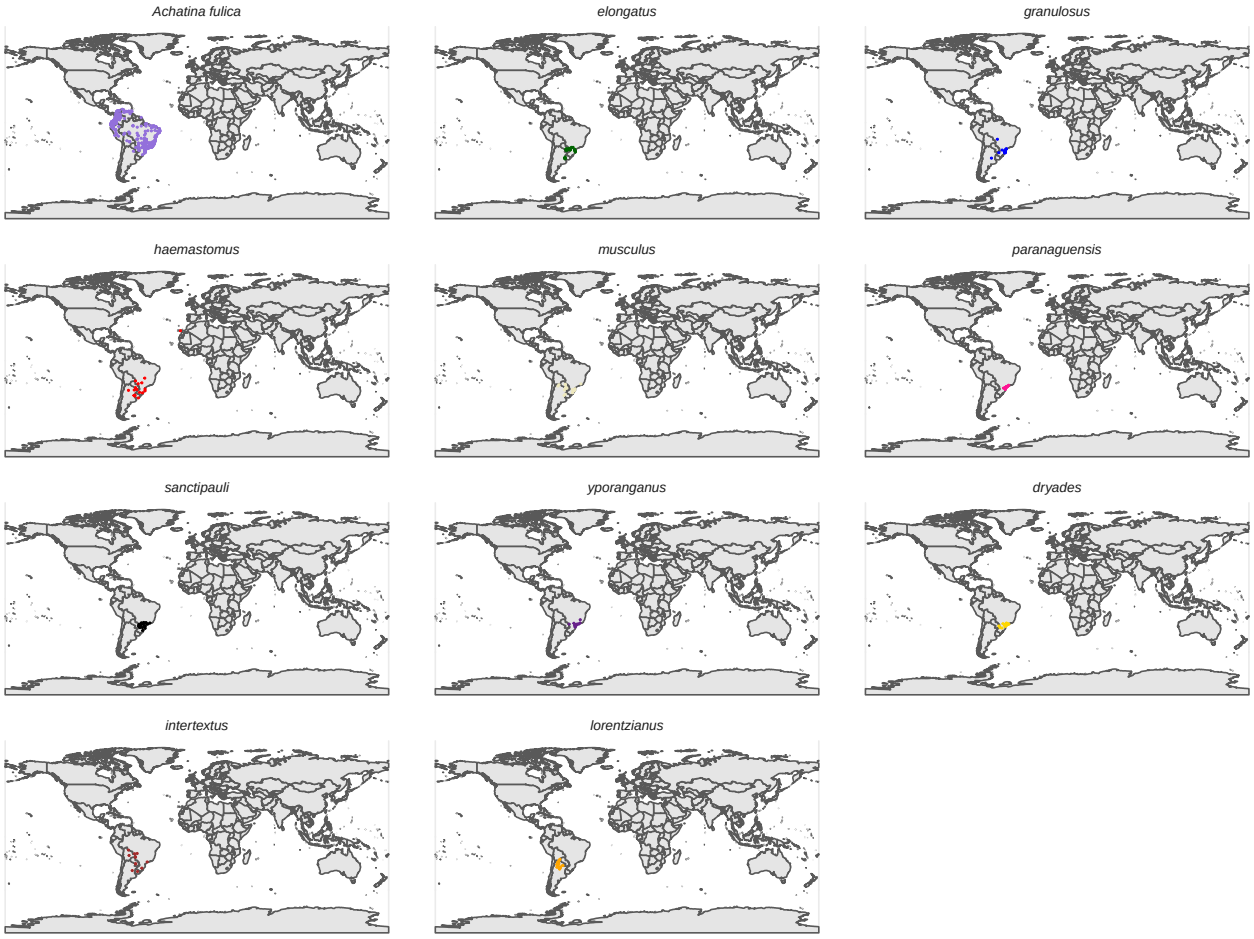
```
g.names <- species

g.codenames <- g.names
```

It is also necessary to set what colors will be used in the next plots for each group (using the same order as the names). Change the colors according to your preferences.

```
g.colors <- c("mediumpurple", 'darkgreen', 'blue','red',
              "lemonchiffon2", "deeppink", "black",
              "darkorchid4", "gold", "brown", "orange")
```

To check the distribution of the occurrence records we map them in a world context.

```
occ.points.thin2 <- do.call(rbind, occ.points.thin) %>%
  tibble::rownames_to_column(var = "species") %>%
  mutate(species = gsub("\\..*", "", species),
         species = factor(species, g.names))
world <- ne_countries(scale = "medium", returnclass = "sf")

g <- ggplot() +
  geom_sf(data = world) +
   theme_minimal() +
  geom_point(occ.points.thin2, mapping = aes(Longitude, Latitude,
                                             col = species),
             size = .3) +
  scale_color_manual(values = g.colors) +
  xlab("") +
  ylab("") +
  theme(
    strip.text = element_text(face = "italic"),
    legend.text = element_text(face = "italic"),
    legend.position = "none"
  )
g1 <- g + facet_wrap(species ~ ., ncol = 3, nrow = 4)
g1
```

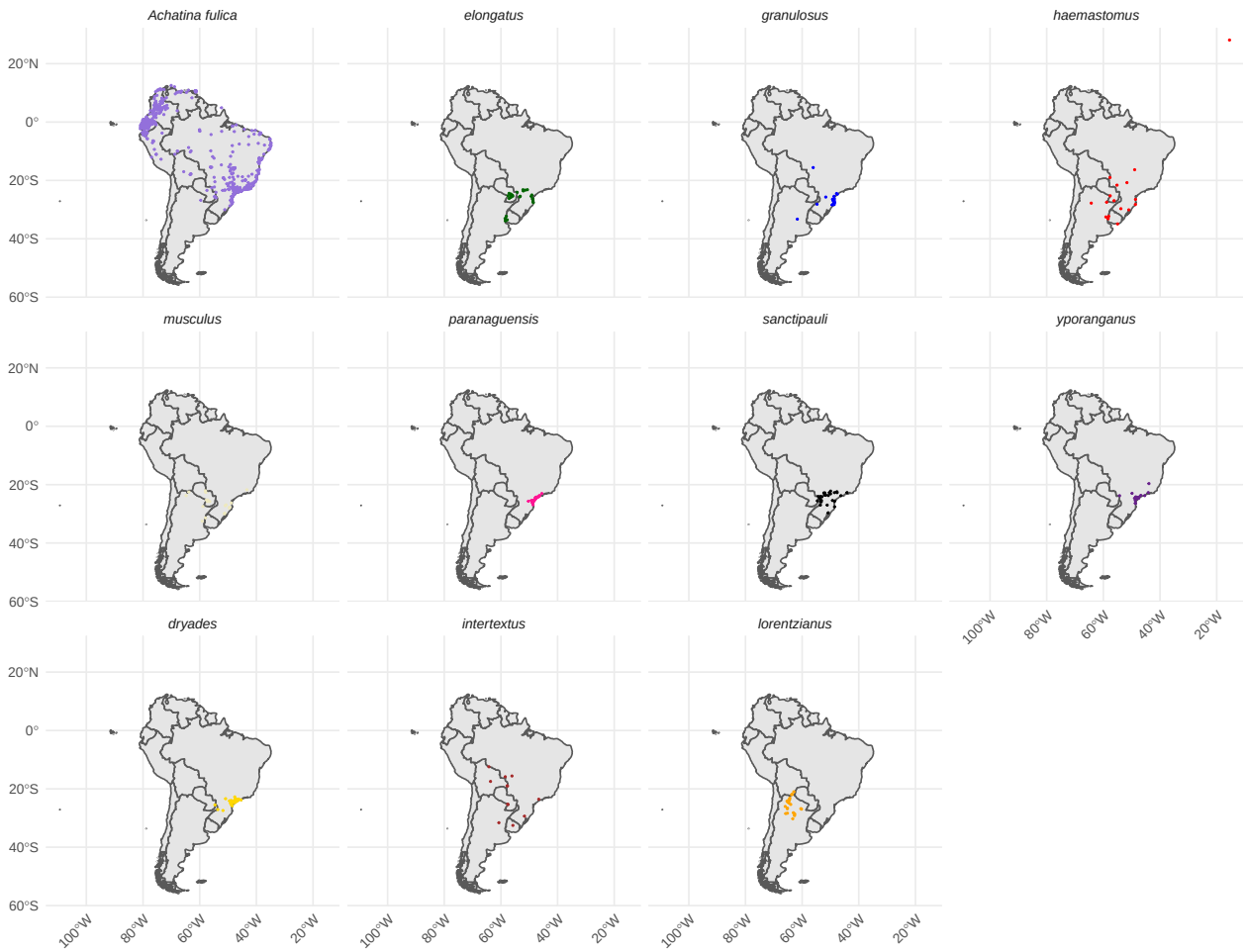



```
loc <- here("Figures", "Figure1.tiff")
# ggsave(loc, g1)
```

Filter points to the countries defined.

```
# map <- world[world$name %in% c("Brazil"), ]

pnts_sf <- st_as_sf(occ.points.thin2,
                    coords = c('Longitude', 'Latitude'),
                    crs = st_crs(map))

# pnts <- pnts_sf %>%
#   mutate(intersection = as.integer(st_intersects(geometry, map)))
#
# remove <- is.na(pnts$intersection)
occ.points.thin3 <- occ.points.thin2

g <- ggplot() +
  geom_sf(data = filter(world, continent == "South America")) +
   theme_minimal() +
  geom_point(occ.points.thin3, mapping = aes(Longitude, Latitude,
                                             col = species),
             size = .3) +
  scale_color_manual(values = g.colors) +
  xlab("") +
```

```
  ylab("") +
  theme(
    strip.text = element_text(face = "italic"),
    legend.text = element_text(face = "italic"),
    legend.position = "none",
     axis.text.x = element_text(angle = 45, hjust = 1))

g1 <- g + facet_wrap(species ~ ., ncol = 4, nrow = 3)
g1
```




```
loc <- here("Figures", "Figure2.tiff")
# ggsave(loc, g1)
```

Background definition

An important step in the niche analysis is the definition of the background. Here we applied a background based on a minimum convex polygon (MCP) made from the occurrence records of each group. Additionally to the MCP we add a buffer around it. The polygon buffer size for the background (in degrees) can be changed below. We chose 2 degrees based on the species dispersion.

```
buffer.size <- 2
```

We define a minimum convex polygon (MCP) function below (this function was obtained from https:

//github.com/ndimhypervol/wallace).

```
mcp <- function (xy) {
  xy <- as.data.frame(coordinates(xy))
  coords.t <- chull(xy[, 1], xy[, 2])
  xy.bord <- xy[coords.t, ]
  xy.bord <- rbind(xy.bord[nrow(xy.bord), ], xy.bord)
  return(SpatialPolygons(list(Polygons(list(Polygon(as.matrix(xy.bord))), 1))))
}
```

Enviromnental variables

The environmental variables used are available at the WorldClim website (http://www.worldclim.org). Download all the 19 bioclimatic ('Biolclim') variables for the current conditions (we used the resolution of 10 arc-min) with the code below. Note you need to have the internet on. The download files are opened directed in the R environment, but they are also saved in your work directory (to see where it is, use `getwd()`).

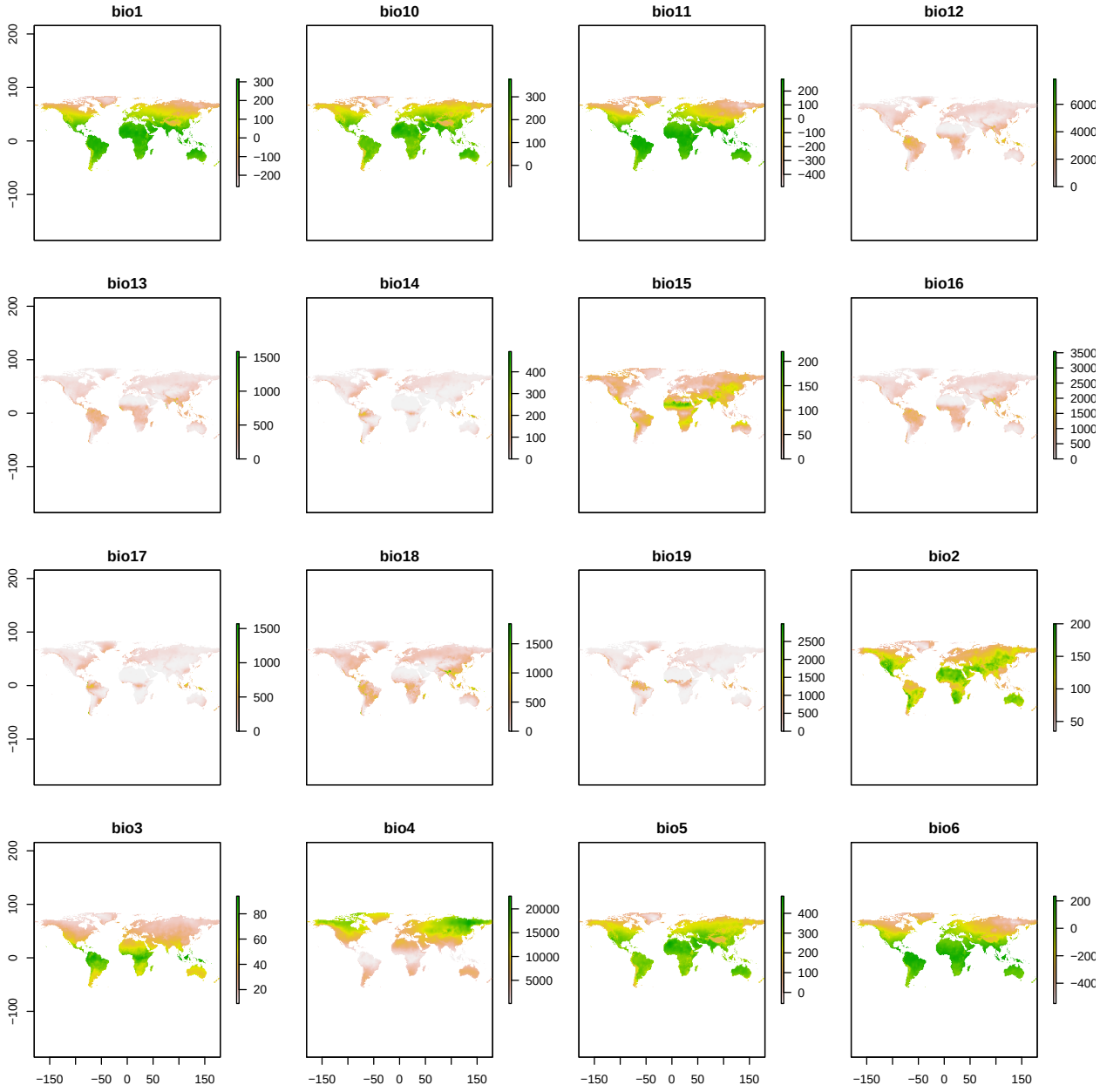
```
variables <- raster::stack(list.files(
  here("pres"),
  full.names = TRUE))
```

In the manuscript we used all the 19 bioclimatic variables as before the analysis we will reduce them to a two-dimensional space with a PCA. However, the readers can choose the number of variables to keep by changing the sequence `1:19` in the code below for the variable number you want to keep (to see the name sequence of the variables apply `names(variables)`).

```
variables <- subset(variables, 1:nlayers(variables))
```

You can also check the variables, by mapping them.

```
plot(variables)
```



Group assigning

Once, we have the occurrence data, the environmental data, the defined groups and their background parameters chosen, we can prepare the data for the analysis. Below we use the occurrence points to generate the MCP plus a buffer defined by the user for the background (see above). Next, the variable values per group are extracted from the species occurrence points and from the background (defined above). Finally we plot the resulting groups with their respective backgrounds.

```
# Empty objects
g.assign <- occ.points.thin3$species
xy.mcp <- list()
back.env <- list()
spec.env <- list()
row.sp <- list()
```

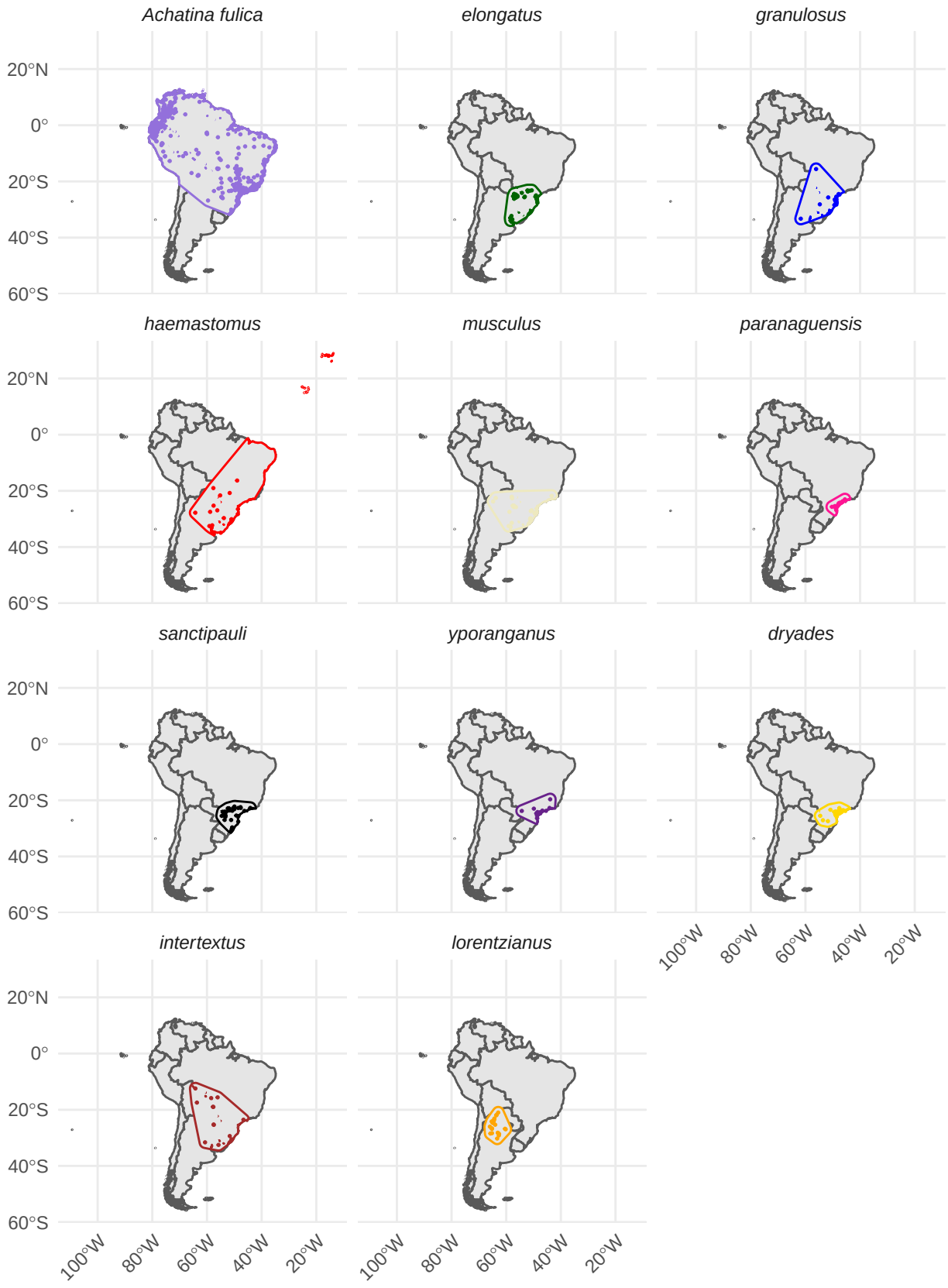
```
united <- st_union(st_make_valid(world))

# Loop
for (i in 1:n.groups) {
  print(i)
  # Save row numbers per species
  g.limit <- g.assign == species[i]
  row.sp[[i]] <- which(g.limit)

  # Background polygon
  mcp.occ <- mcp(as.matrix(occ.points.thin3[g.limit, -1]))
  xy.mcp.i <- gBuffer(mcp.occ, width = buffer.size) %>%
    st_as_sf() %>%
    st_set_crs(value = st_crs(world))

  xy.mcp[[i]] <- st_intersection(xy.mcp.i, united)
  xy.mcp[[i]]$species <- species[i]
  # Background environment
  extract_temp <- raster::extract(variables, as_Spatial(xy.mcp[[i]]))
  back.env[[i]] <- na.exclude(do.call(rbind.data.frame,
                                      extract_temp))
  # Species environment
  spec.env[[i]] <- na.exclude(raster::extract(variables,
                                      occ.points.thin3[g.limit, -1]))
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
```

Map buffers:

```
xy.mcp.is <- do.call(rbind, xy.mcp)
xy.mcp.is$species <- factor(xy.mcp.is$species, xy.mcp.is$species)

g <- ggplot() +
  geom_sf(data = filter(world, continent == "South America")) +
   theme_minimal() +
  geom_sf(data = xy.mcp.is, aes(col = species)) +
  geom_point(occ.points.thin3, mapping = aes(Longitude, Latitude,
                                             col = species),
             size = .3) +
  scale_color_manual(values = g.colors) +
  xlab("") +
  ylab("") +
```

```
  theme(
    strip.text = element_text(face = "italic"),
    legend.text = element_text(face = "italic"),
    legend.position = "none",
     axis.text.x = element_text(angle = 45, hjust = 1))
g1 <- g + facet_wrap(species ~ ., ncol = 3, nrow = 4)
g1
```


Achatina fulica
elongatus
granulosus
haemastomus
musculus
paranaguensis
sanctipauli
yporanganus
dryades
intertextus
lorentzianus
20°N
0°
20°S
40°S
60°S
100°W
80°W
60°W
40°W
20°W

```
loc <- here("Figures", "Figure.tiff")
#ggsave(loc, g1)
```

Save the occurrence points table.

```
write.csv(occ.points.thin3,
          file = here("Data", "occurrences.csv"),
          row.names = FALSE)
```

Now we organize the final tables to be used.

```
# Occurrence points per group
g.occ.points <- occ.points.thin3
colnames(g.occ.points)[1] <- "Groups"
# Environmental values for the background
all.back.env <- do.call(rbind.data.frame, back.env)
# Environmental values for the species occurrence points
all.spec.env <- do.call(rbind.data.frame, spec.env)
# Environmental values all together
data.env <- rbind(all.spec.env, all.back.env)
```

Check the number of occurrence records per region.

```
table(g.occ.points[, 1])
```

Achatina fulica elongatus granulosus haemastomus musculus 586 25 19 17 22 paranaguensis sanctipauli yporanganus dryades intertextus 25 29 17 31 12 lorentzianus 24

Niche comparissons

The niche analyzes and comparisons follow the framework developed by Broennimann et al. (2012) and its derivations (see methods section in the manuscript).

PCA

We chose to apply a PCA (Principal Component Analysis) considering all the environments together, as it presented the best performance when comparing the niches (Broennimann et al., 2012).

```
# Weight matrix
w <- c(rep(0, nrow(all.spec.env)), rep(1, nrow(all.back.env)))
# PCA of all environment
pca.cal <- dudi.pca(data.env, row.w = w, center = TRUE,
                    scale = TRUE, scannf = FALSE, nf = 2)
```

Once we have the pca results, we need the first and second eigenvector values for the background and for the occurrence records per group.

```
# Rows in data corresponding to sp1
adtion <- cumsum(c(0, sapply(back.env, nrow)))
begnd <- nrow(all.spec.env)
# Empty list to save the results
scores.back <- list()
scores.spec <- list()

# Assigning the values
```

```
for (i in 1:n.groups) {
  scores.spec[[i]] <- pca.cal$li[row.sp[[i]], ]
  pos <- (begnd[1] + adtion[i] + 1) : (begnd[1] + adtion[i + 1])
  scores.back[[i]] <- pca.cal$li[pos, ]
}

total.scores.back <- do.call(rbind.data.frame, scores.back)
```

Environmental space

An environmental space is generated based on the pca values calculated for the background and the occurrence records. We defined the resolution of this two-dimensional space grid below.

```
R <- 100
```

Next, we modeled the species density in the environmental grid, considering the observed occurrence density and the availability of the conditions in the background.

```
z <- list()

for (i in 1:n.groups) {
  z[[i]] <- ecospat.grid.clim.dyn(total.scores.back,
                                  scores.back[[i]],
                                  scores.spec[[i]],
                                  R = R)
}
```

Niche overlap

For the niche overlap, we calculate the D metric and its significance, using a similarity test. We define the number of interactions for the similarity test below (see the methods section in the manuscript for details).

```
rep <- 100
```

Once the number of interactions is defined, we can generate the values. Additionally, we calculate the partition of the non-overlapped niche, among niche unfilling, expansion and stability (see methods in the manuscript).

```
# Empty matrices
D <- matrix(nrow = n.groups, ncol = n.groups)
rownames(D) <- colnames(D) <- substr(g.codenames, 1, 3)
unfilling <- stability <- expansion <- sim <- D

for (i in 1) {

  for (j in 2:n.groups) {

    x1 <- z[[i]]
    x2 <- z[[j]]

    # Niche overlap
    D[i, j] <- ecospat.niche.overlap (x1, x2, cor = TRUE)$D

    # Niche similarity
    sim[i, j] <- ecospat.niche.similarity.test (x1, x2, rep,
```

```
                                                alternative = "greater")$p.D
    sim[j, i] <- ecospat.niche.similarity.test (x2, x1, rep,
                                                alternative = "greater")$p.D

    # Niche Expansion, Stability, and Unfilling
    index1 <- ecospat.niche.dyn.index (x1, x2,
                                       intersection = NA)$dynamic.index.w
    index2 <- ecospat.niche.dyn.index (x2, x1,
                                       intersection = NA)$dynamic.index.w
    expansion[i, j] <- index1[1]
    stability[i, j] <- index1[2]
    unfilling[i, j] <- index1[3]
    expansion[j, i] <- index2[1]
    stability[j, i] <- index2[2]
    unfilling[j, i] <- index2[3]
  }
}
```

Numeric results

Below we present the results for each metric, among all the groups.

D value:

```
kable(D, digits = 3, format = "markdown")
```

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
Ach	NA	0.021	0.033	0.033	0.035	0.046	0.024	0.024	0.039	0.014	0.002
elo	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
gra	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
hae	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
mus	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
par	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
san	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
ypo	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
dry	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
int	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
lor	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Niche similarity null model (p-values):

```
kable(sim, digits = 3, format = "markdown")
```

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
Ach	NA	0.03	0.01	0.04	0.01	0.01	0.02	0.02	0.01	0.059	0.218
elo	0.020	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
gra	0.010	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
hae	0.040	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
mus	0.020	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
par	0.010	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
san	0.040	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
ypo	0.010	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
dry	0.010	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
int	0.040	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
lor	0.188	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Niche Unfilling:

```
kable(unfilling, digits = 3,  format = "markdown")
```

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
Ach	NA	0.801	0.783	0.601	0.646	0.736	0.79	0.772	0.728	0.564	0.983
elo	0.000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
gra	0.019	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
hae	0.042	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
mus	0.046	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
par	0.000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
san	0.013	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
ypo	0.000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
dry	0.000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
int	0.033	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
lor	0.558	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Niche Expansion:

```
kable(expansion, digits = 3,  format = "markdown")
```

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
Ach	NA	0	0.019	0.042	0.046	0	0.013	0	0	0.033	0.558
elo	0.801	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
gra	0.783	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
hae	0.601	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
mus	0.646	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
par	0.736	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
san	0.790	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
ypo	0.772	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
dry	0.728	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
int	0.564	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
lor	0.983	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Niche Stability:

```
kable(stability, digits = 3,  format = "markdown")
```

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
Ach	NA	1	0.981	0.958	0.954	1	0.987	1	1	0.967	0.442
elo	0.199	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
gra	0.217	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
hae	0.399	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
mus	0.354	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
par	0.264	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

	Ach	elo	gra	hae	mus	par	san	ypo	dry	int	lor
san	0.210	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
ypo	0.228	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
dry	0.272	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
int	0.436	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
lor	0.017	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Figure results

Individual niche plots

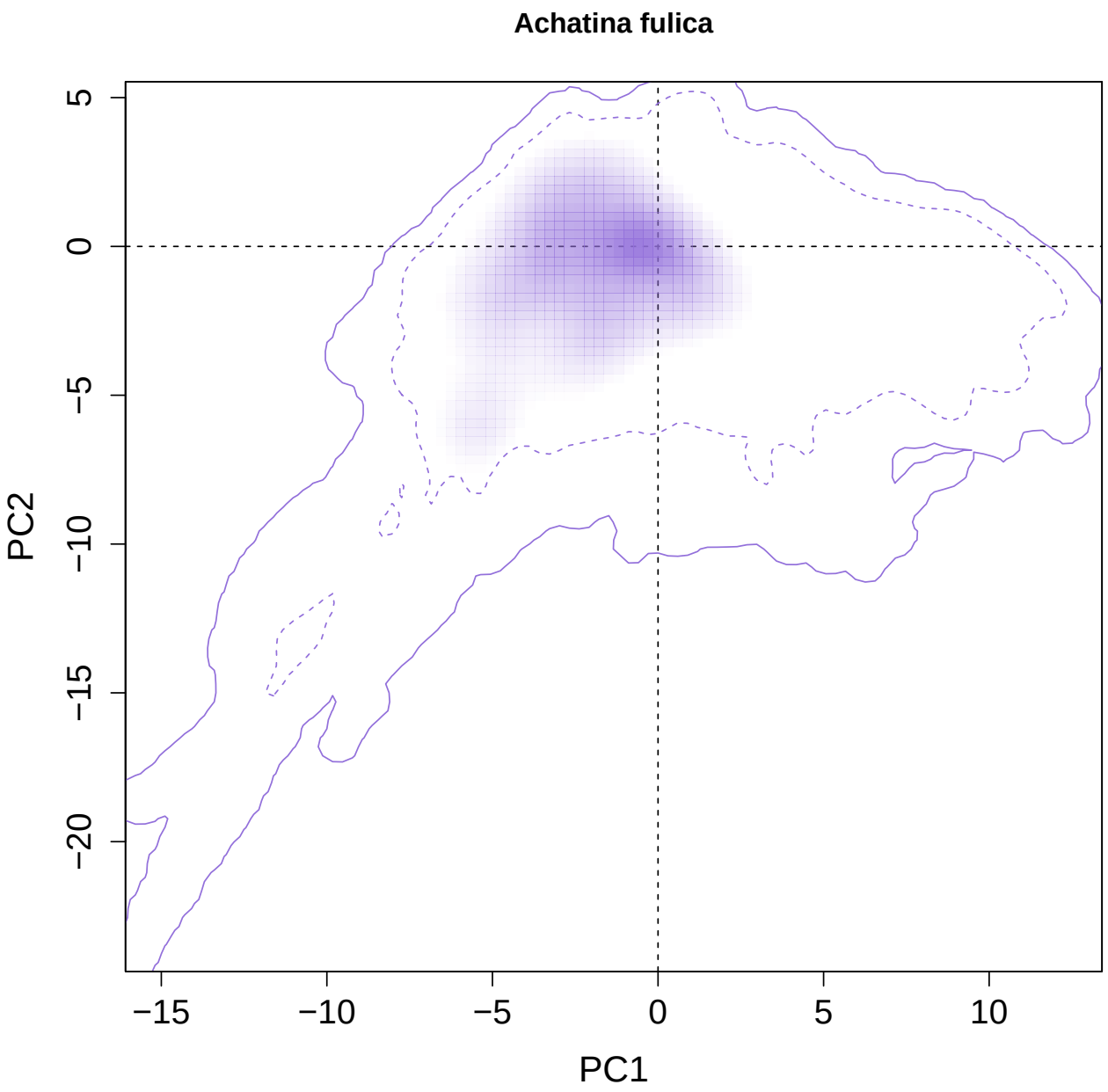
We developed some modifications in the plot.niche function available at Broennimann et al 2012. The modifications include more options and flexibility to the plot.

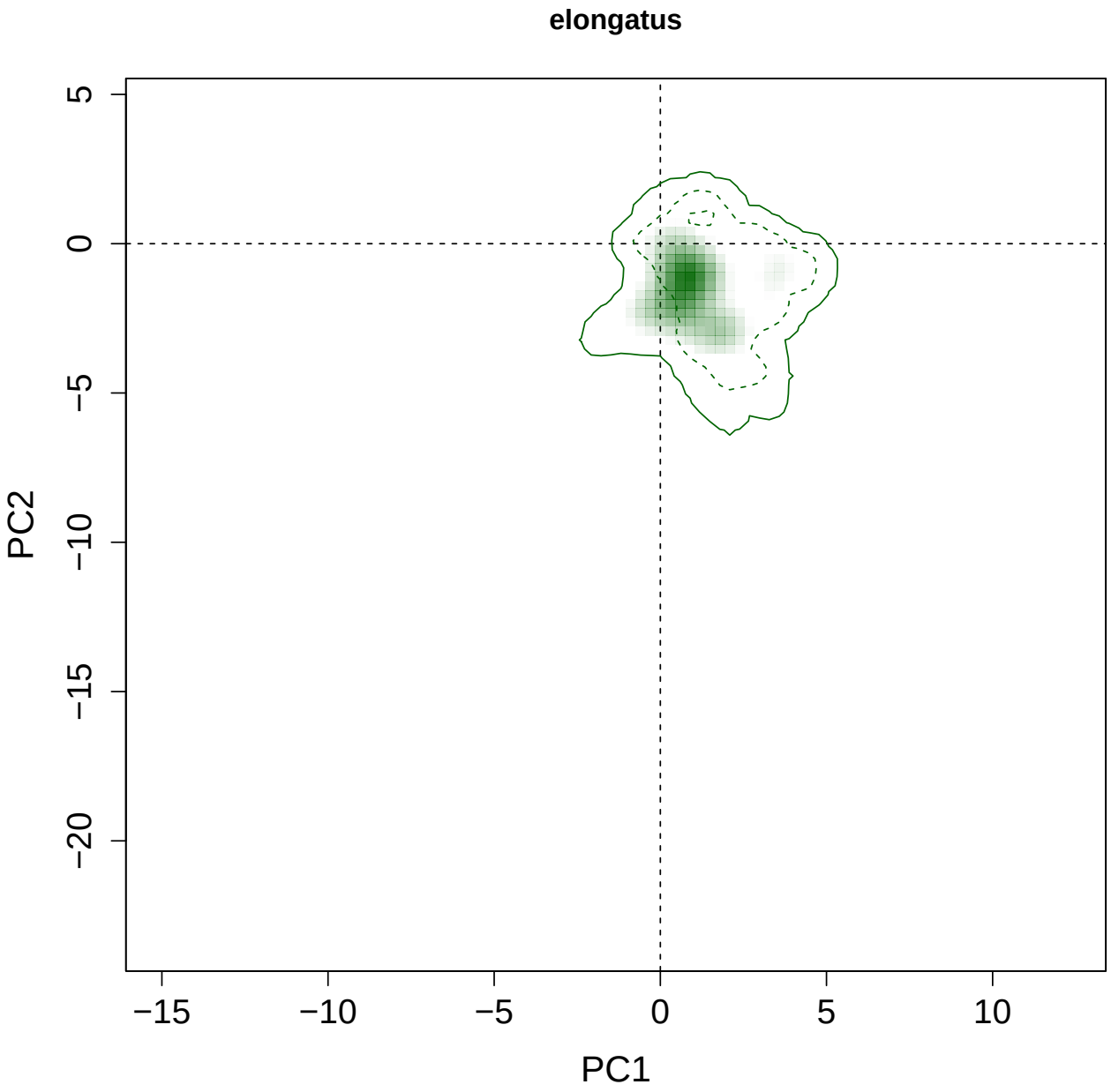
```
plot.niche.mod <- function(z, name.axis1 = "PC1", name.axis2 = "PC2",
                           cor = F, corte,  contornar = TRUE,
                           densidade = TRUE, quantis = 10,
                           back = TRUE, x = "red", title = "",
                           i) {


  cor1 <- function(cores.i, n) {
    al <- seq(0,1,(1/n))
    cores <- numeric(length(n))
    for(i in 1:n) {
      corespar <- col2rgb(cores.i)/255
      cores[i] <- rgb(corespar[1, ], corespar[2, ],
                      corespar[3, ], alpha = al[i])
    }
    return(cores)
  }


  a1 <- colorRampPalette(c("transparent",cor1(x, quantis)), alpha = TRUE)

  xlim <- c(min(sapply(z, function(x){min(x$x)})),
            max(sapply(z, function(x){max(x$x)})))

  ylim <- c(min(sapply(z, function(x){min(x$y)})),
            max(sapply(z, function(x){max(x$y)})))

  graphics::image(z[[1]]$x, z[[1]]$y,
                  t(as.matrix(z[[i]]$z.uncor))[, nrow(as.matrix(z[[i]]$z.uncor)):1],
                  col = "white",
                  ylim = ylim, xlim = xlim,
                  zlim = c(0.000001, max(as.matrix(z[[1]]$z.uncor), na.rm = T)),
                  xlab = "PC1", ylab = "PC2", cex.lab = 1.5,
                  cex.axis = 1.4)

  abline(h = 0, v = 0, lty = 2)

  if (back) {
```

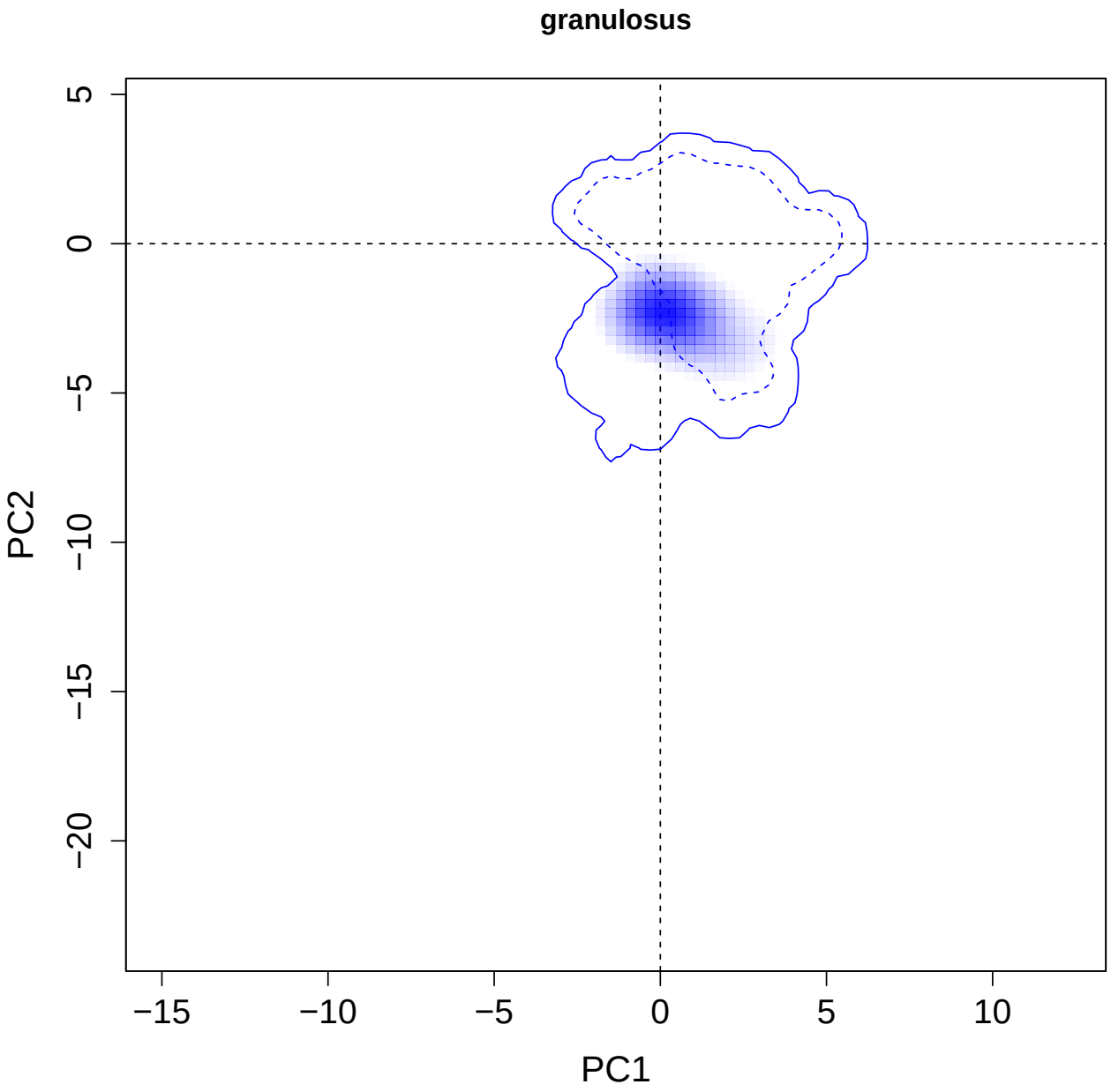
```
    contour(z[[i]]$x, z[[i]]$y,
            t(as.matrix(z[[i]]$Z))[, nrow(as.matrix(z[[i]]$Z)):1],
            add = TRUE, levels = quantile(z[[i]]$Z[z[[i]]$Z > 0],
                                          c(0, 0.5)), drawlabels = FALSE,
            lty = c(1, 2), col = x, lwd = 1)
  }

  if (densidade) {
    image(z[[i]]$x, z[[i]]$y, t(as.matrix(z[[i]]$z.uncor))[, nrow(as.matrix(z[[i]]$z.uncor)):1], col = a
  }


  if(contornar){
    contour(z[[i]]$x, z[[i]]$y, t(as.matrix(z[[i]]$z.uncor))[, nrow(as.matrix(z[[i]]$z.uncor)):1],
            add = TRUE, levels = quantile(z[[i]]$z.uncor[z[[i]]$z.uncor > 0],
                                          seq(0, 1, (1 / quantis))),
            drawlabels = FALSE, lty = c(rep(2,(quantis - 1)), 1),
            col = cor1(x, quantis), lwd = c(rep(1, (quantis - 1)), 2))
  }

  title(title)
  box()
}
```

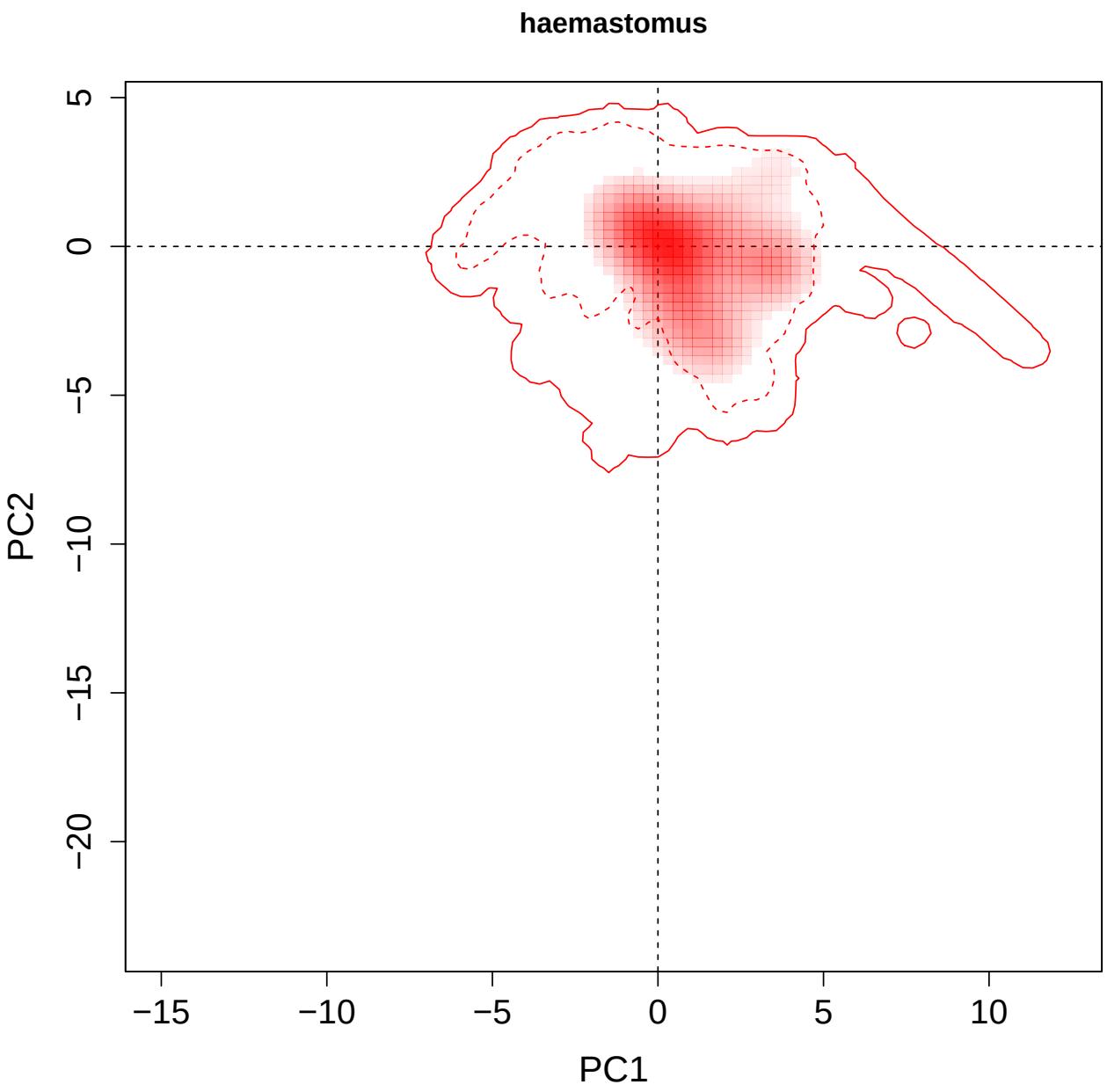
We applied this function here to plot all individual results per group. The continuous line represent the 100% of the available environmental background and the dashed line represents the 50% most common conditions.

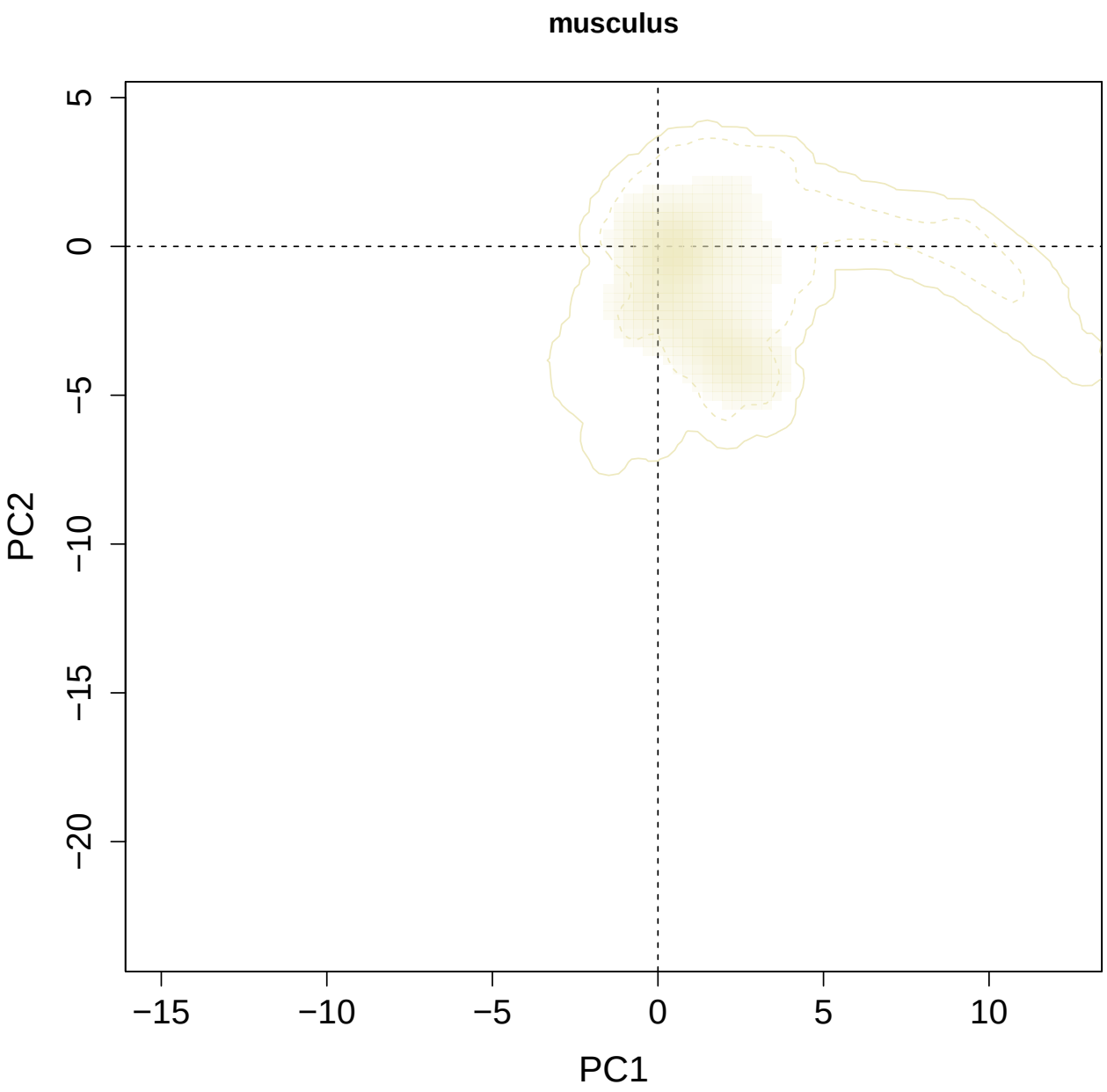
```
for(i in 1:n.groups) {
  plot.niche.mod(z, name.axis1 = "PC1", name.axis2 = "PC2",
                 cor = F, corte,  contornar = FALSE,
                 densidade = TRUE, quantis = 10,
                 back = TRUE, x = g.colors[i], title = g.names[i], i)
}
```

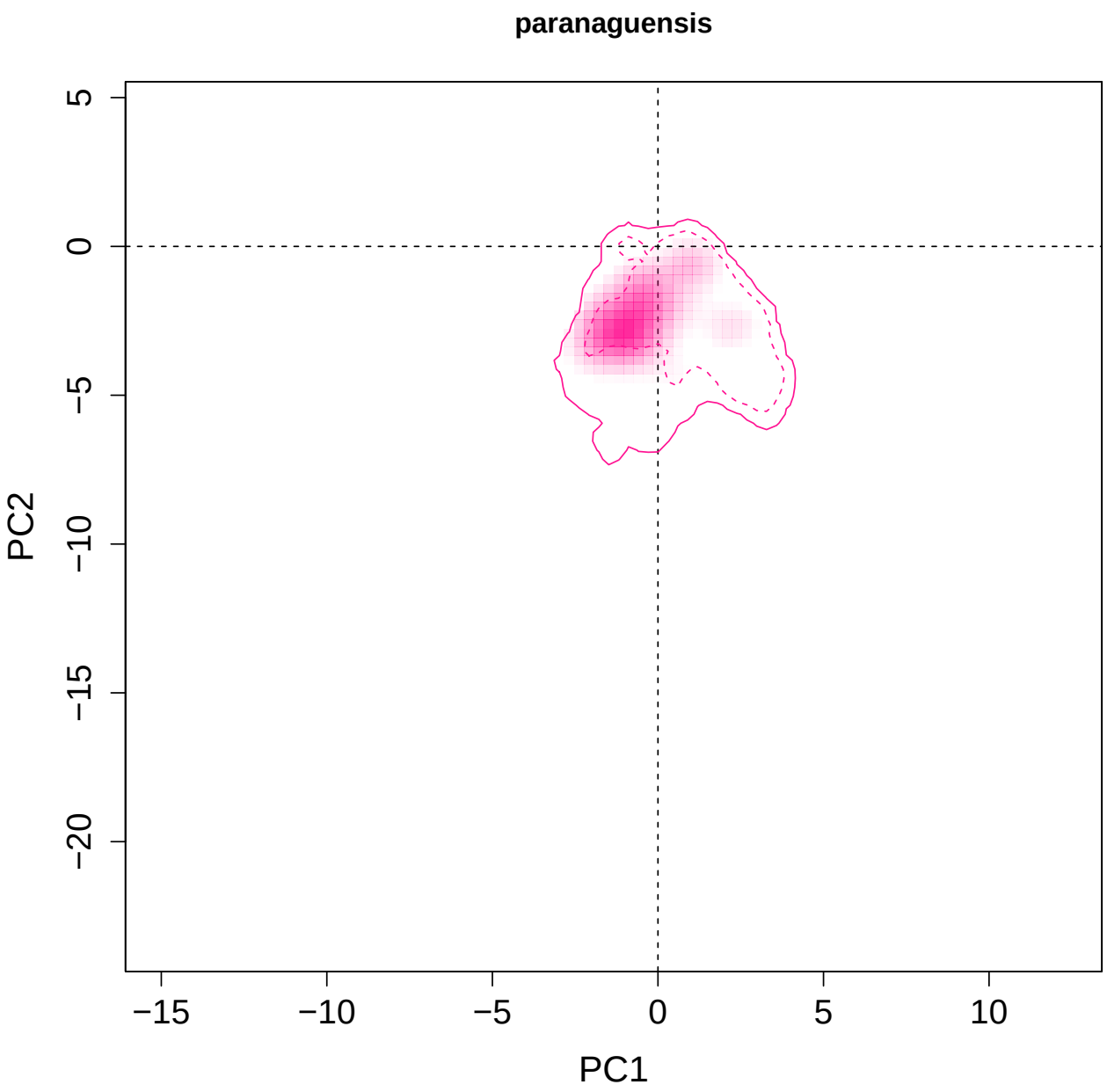

Achatina fulica
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

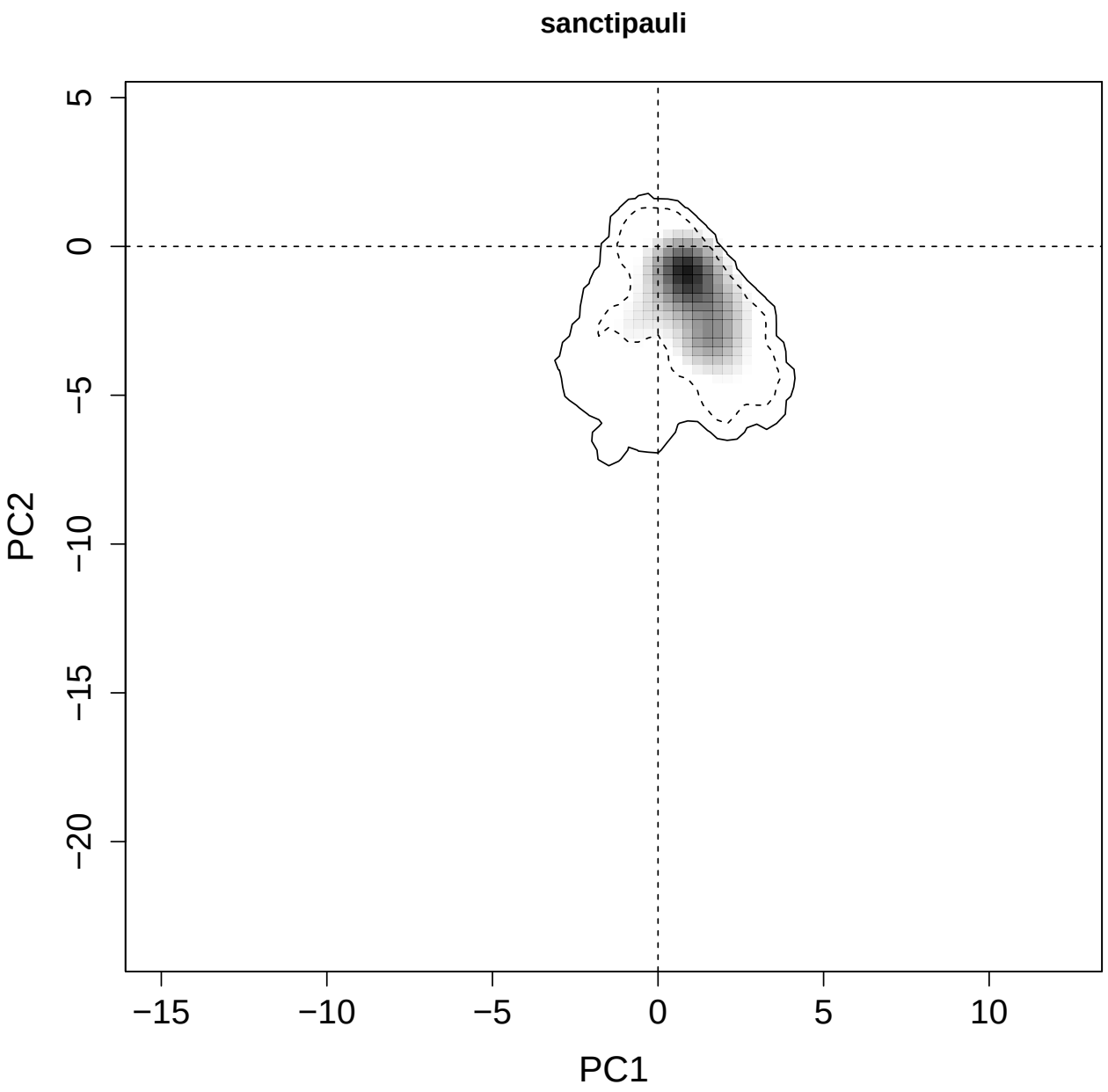

elongatus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

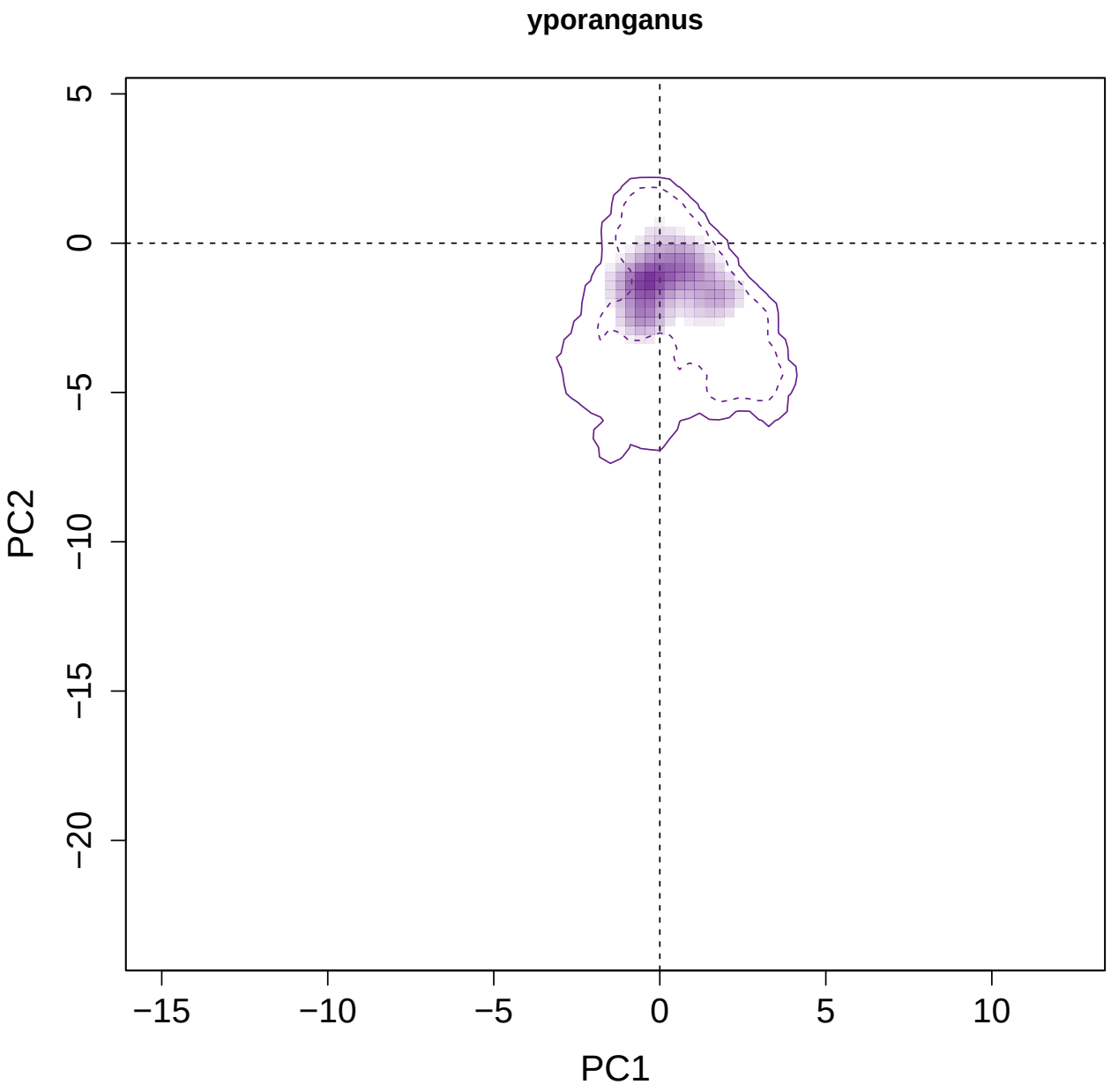

granulosus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

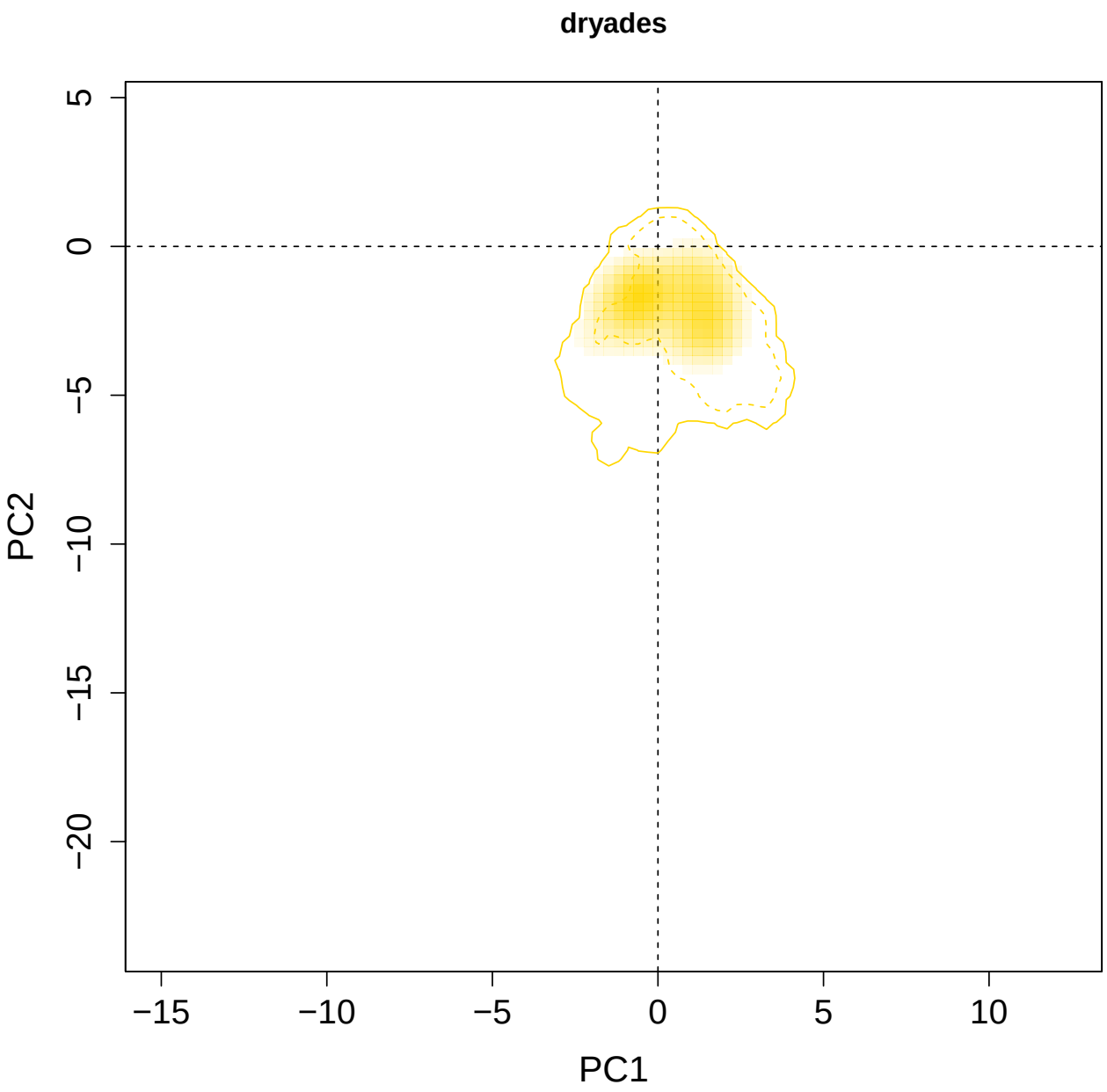

haemastomus
PC2
PC1
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10

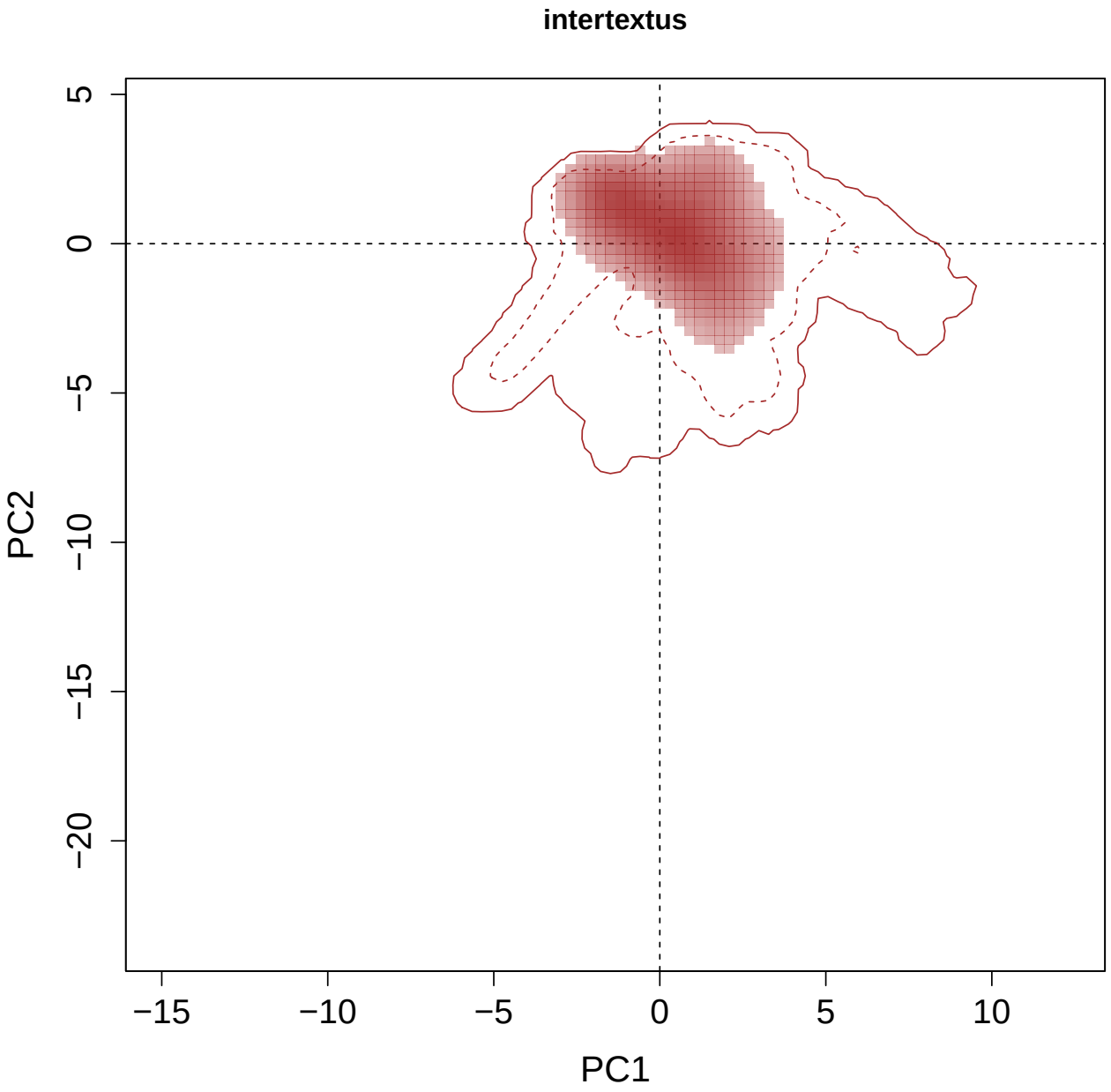

musculus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

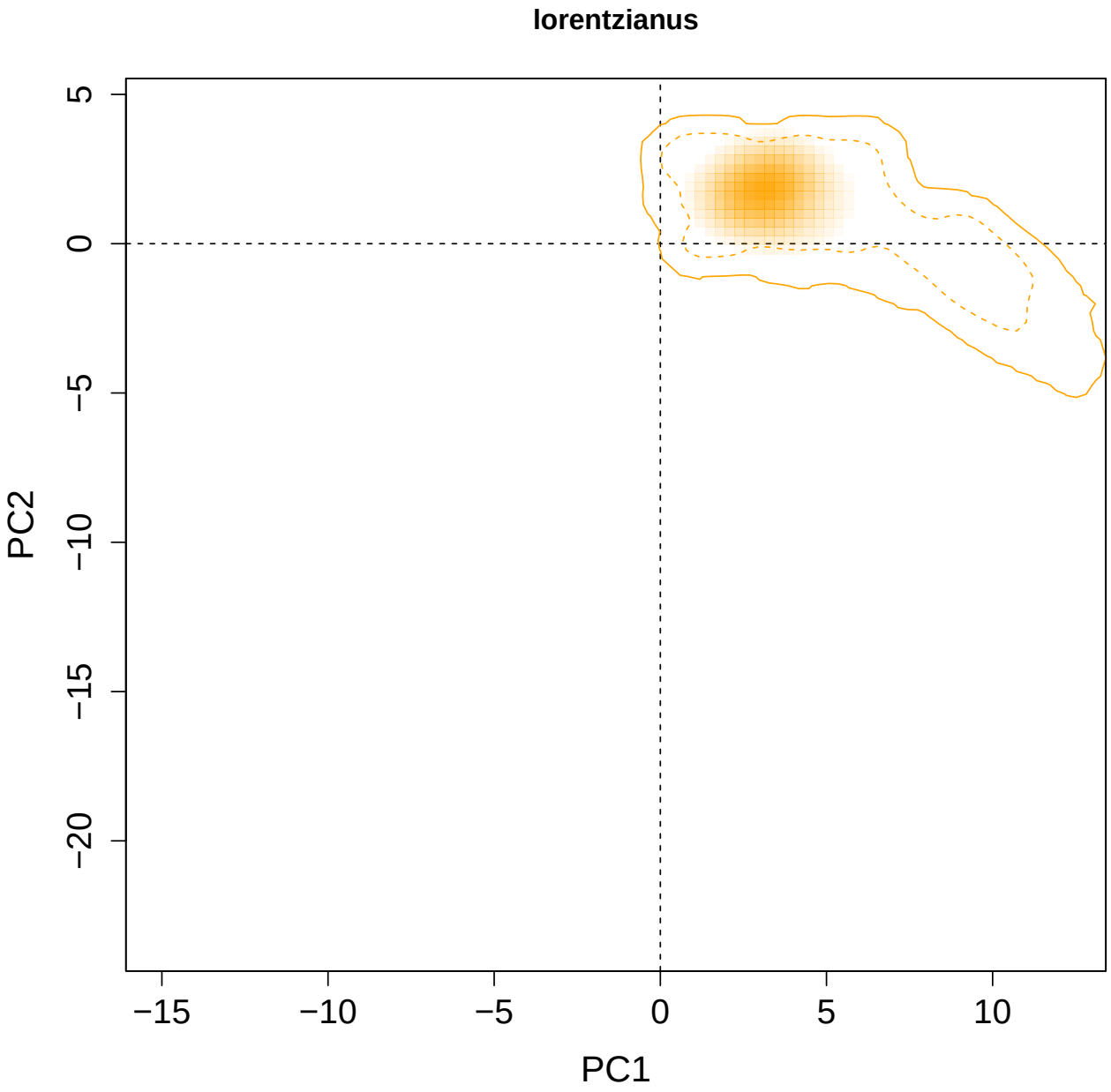

paranaguensis
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1


sanctipauli
PC2
PC1
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10


yporanganus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1


dryades
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1


intertextus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1



Multiple niche plots

We also modified the same function to allow multiple regions/species plots.

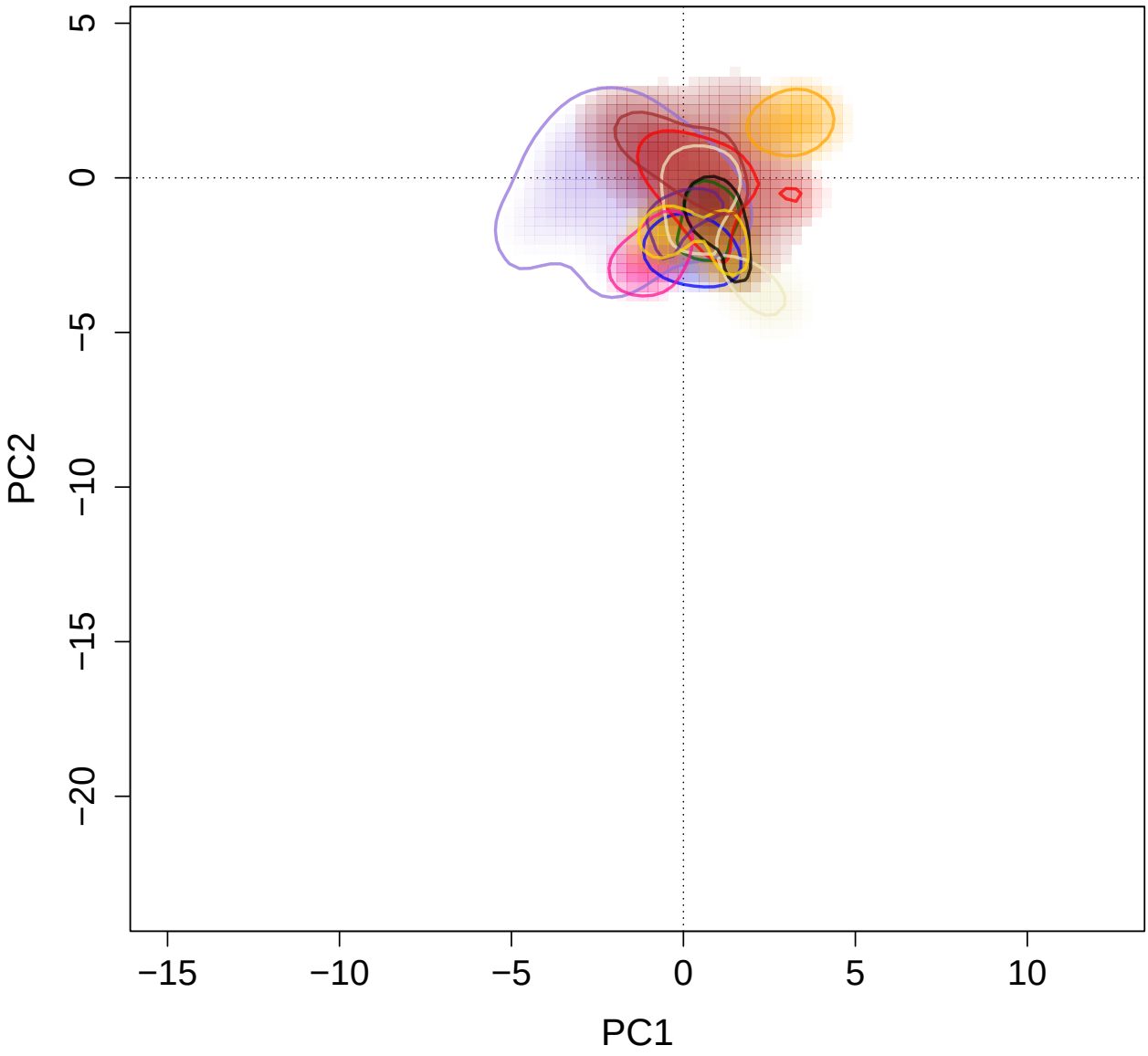
```
plot.niche.all <- function(z, n.groups, g.names,
                           contornar = TRUE,
                           densidade = TRUE,
                           quantis = 10,
                           back = TRUE, title = "",
                           g.colors, n = 5,
                           cor1) {

  # Color func
  cor1 <- function(cores.i, n) {
```

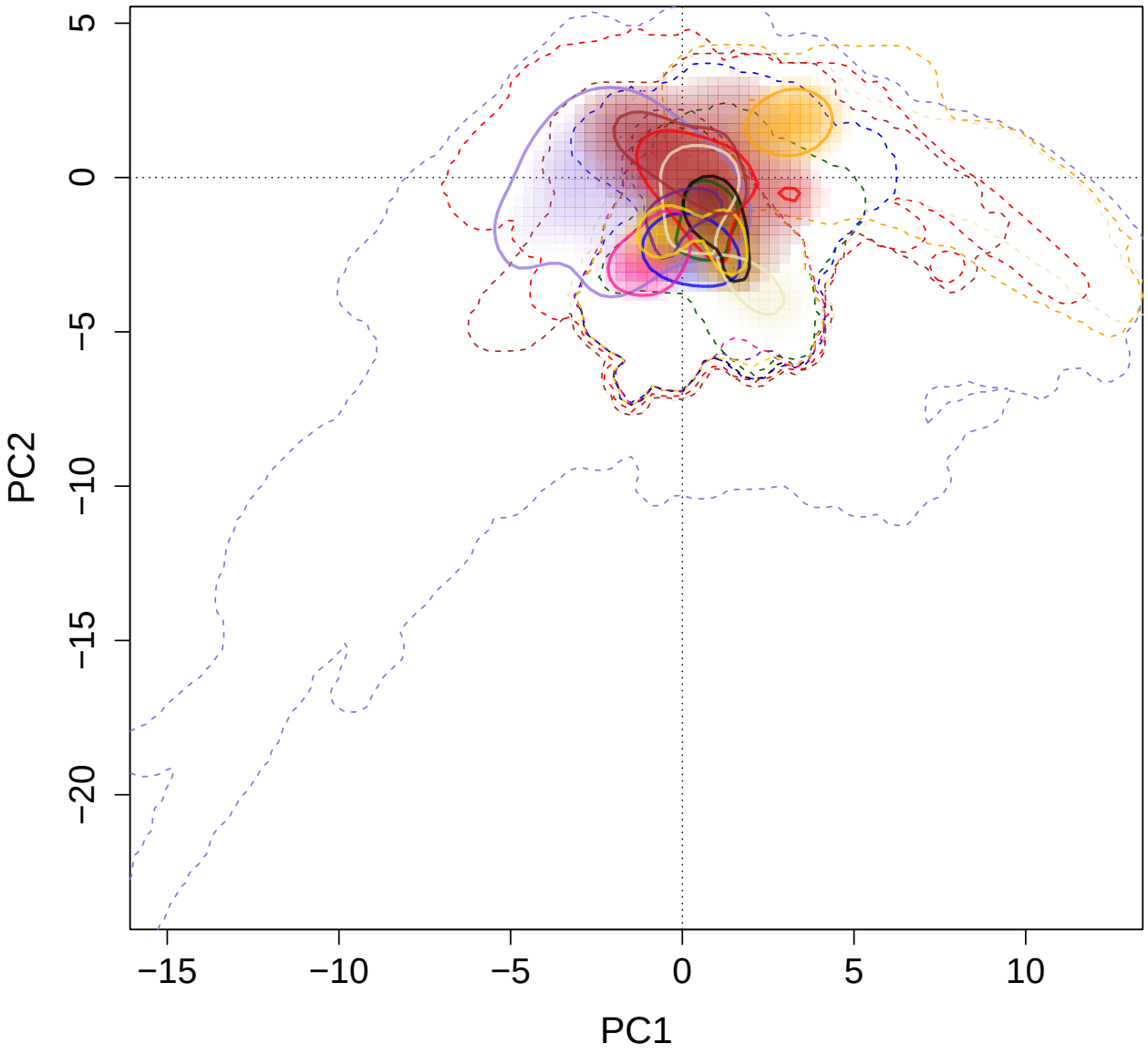
```
  al <- seq(0,1,(1/n))
  cores <- numeric(length(n))
  for(i in 1:n) {
    corespar <- col2rgb(cores.i)/255
    cores[i] <- rgb(corespar[1, ], corespar[2, ],
                    corespar[3, ], alpha = al[i])
  }
  return(cores)
}


a <- list()
for(i in 1:n.groups) {
  a[[i]] <- colorRampPalette(c("transparent", cor1(g.colors[i], n)),
                             alpha = TRUE)
}

xlim <- c(min(sapply(z, function(x){min(x$x)})),
          max(sapply(z, function(x){max(x$x)})))

ylim <- c(min(sapply(z, function(x){min(x$y)})),
          max(sapply(z, function(x){max(x$y)})))

image(z[[1]]$x, z[[1]]$y, t(as.matrix(z[[1]]$z.uncor))[, nrow(as.matrix(z[[1]]$z.uncor)):1], col = "wh
      ylim = ylim, xlim = xlim,
      zlim = c(0.000001, max(as.matrix(z[[1]]$Z), na.rm = T)),
      xlab = "PC1", ylab = "PC2", cex.lab = 1.5,
      cex.axis = 1.4)
abline(h = 0, v = 0, lty = 3)
box()

if (back) {
  for(i in 1:n.groups) {
    contour(z[[i]]$x, z[[i]]$y, t(as.matrix(z[[i]]$Z))[, nrow(as.matrix(z[[i]]$Z)):1], add = TRUE,
            levels = quantile(z[[i]]$Z[z[[i]]$Z > 0], c(0, 1)),
            drawlabels = FALSE,lty = c(2),
            col = g.colors[i], lwd = 1)
  }
}

if (densidade) {
  for(i in 1:n.groups) {
    image(z[[i]]$x, z[[i]]$y, t(as.matrix(z[[i]]$z.uncor))[, nrow(as.matrix(z[[i]]$z.uncor)):1],
          col = a[[i]](100), add = TRUE)
  }
}


if(contornar){
  for(i in 1:n.groups) {
    contour(z[[i]]$x, z[[i]]$y, t(as.matrix(z[[i]]$z.uncor))[, nrow(as.matrix(z[[i]]$z.uncor)):1], add
            levels = quantile(z[[i]]$z.uncor[z[[i]]$z.uncor > 0],
                              seq(0, 1, (1/quantis)))[quantis],
```

```
              drawlabels = FALSE, lty = rev(c(rep(2,(quantis - 1)), 1)),
              col = rev(cor1(g.colors[i], quantis)),
              lwd = rev(c(rep(1, (quantis - 1)), 2)))
    }
  }

}
```

The results can be seen here. The strong contours represent the 20% highest values of density, and the dashed thin lines represent 100% of the background available in each region.

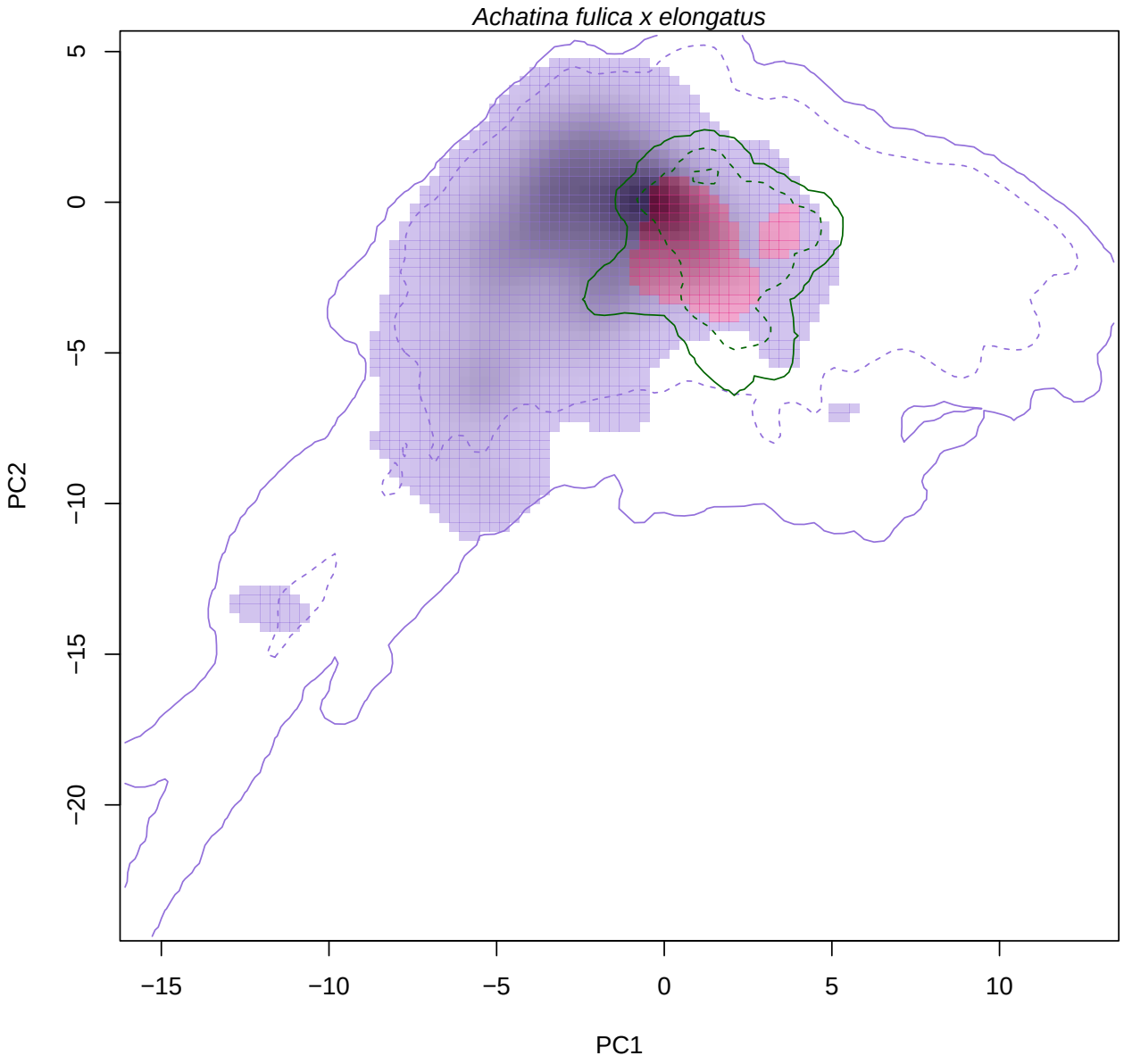
```
plot.niche.all(z, n.groups, g.names,
               contornar = TRUE,
               densidade = TRUE,
               quantis = 4,
               back = FALSE, title = "",
               g.colors, n = 3,
               cor1)

plot.niche.all(z, n.groups, g.names,
               contornar = TRUE,
               densidade = TRUE,
               quantis = 4,
               back = TRUE, title = "",
               g.colors, n = 3,
               cor1)
```

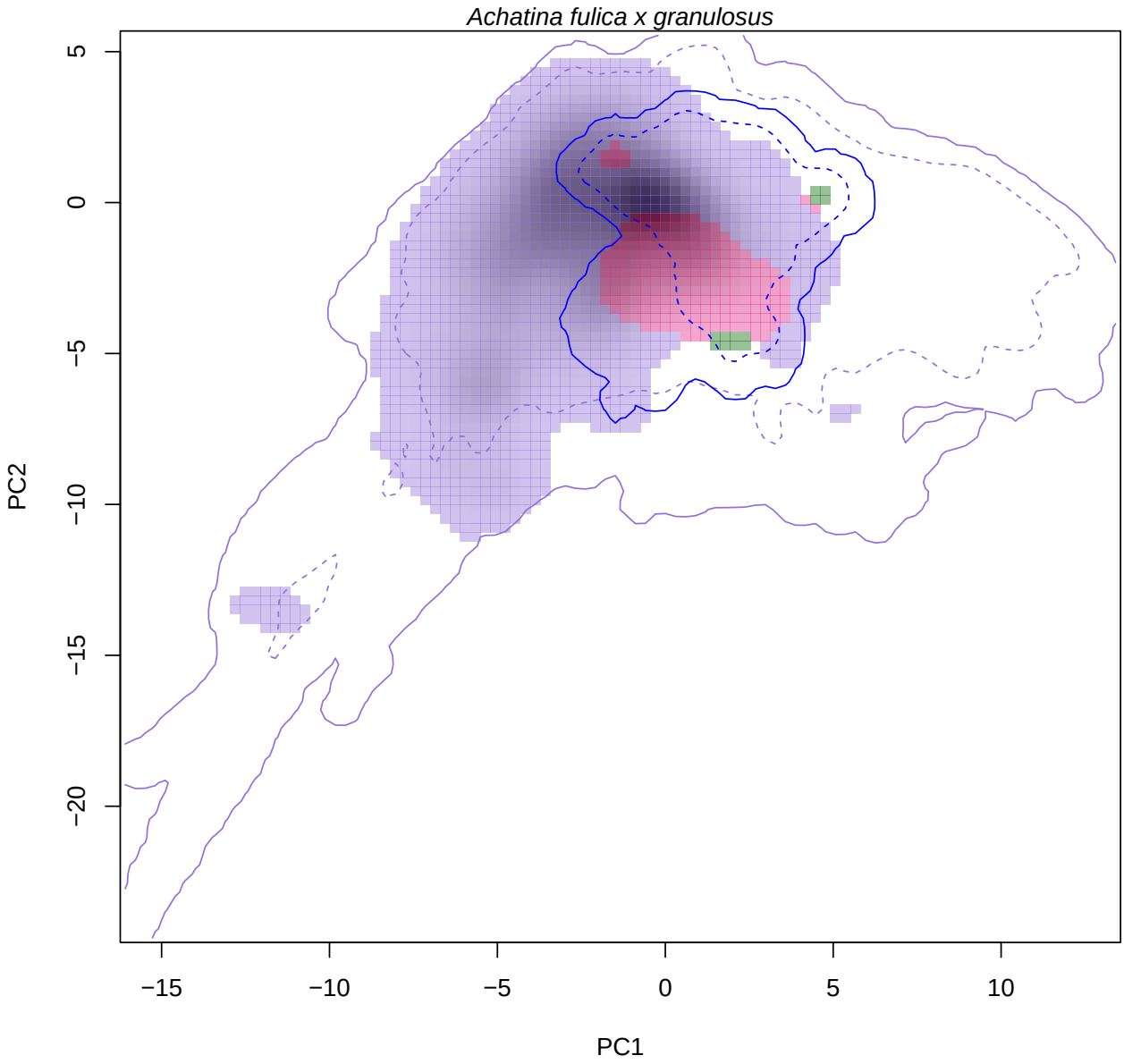

PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

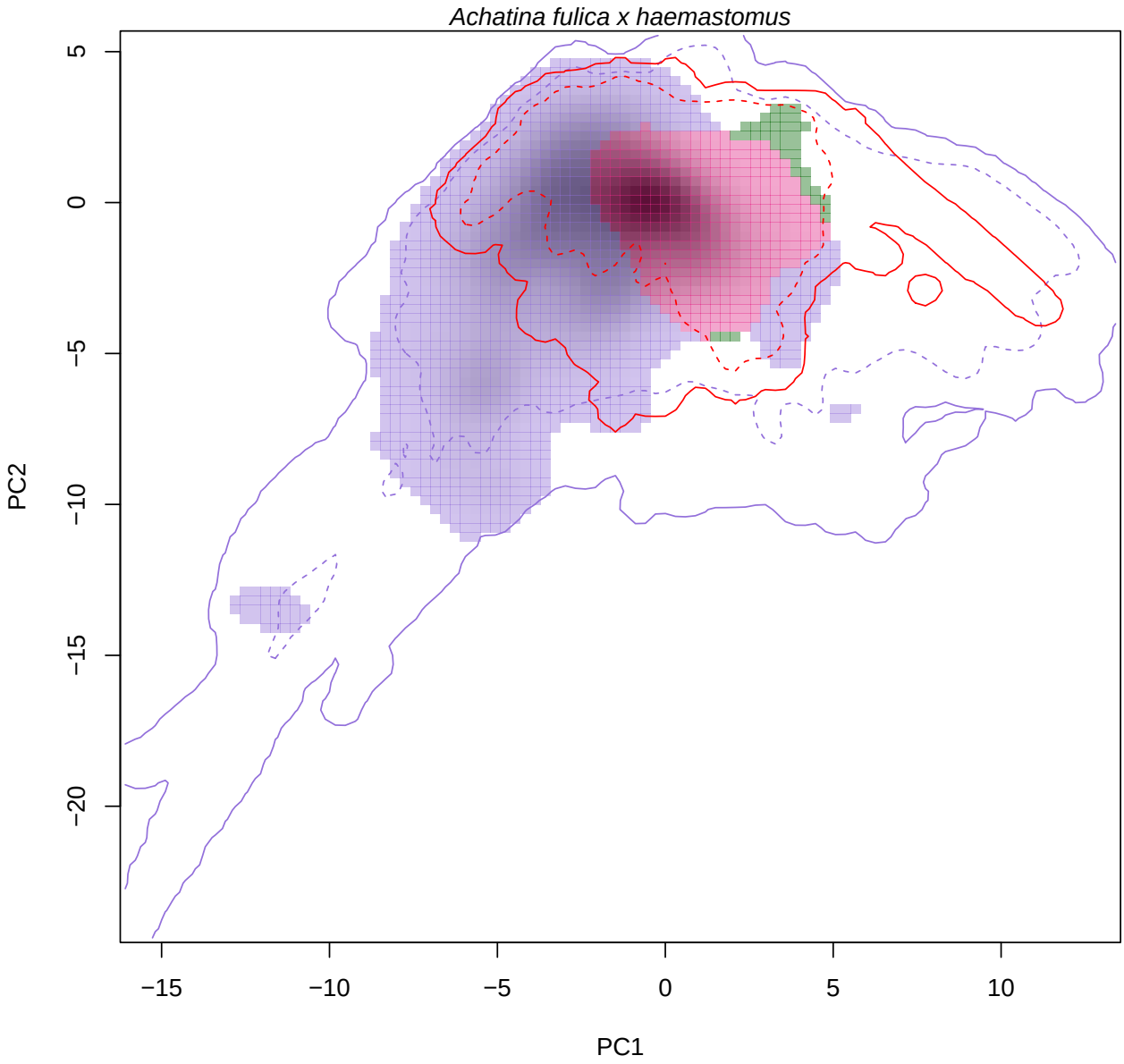


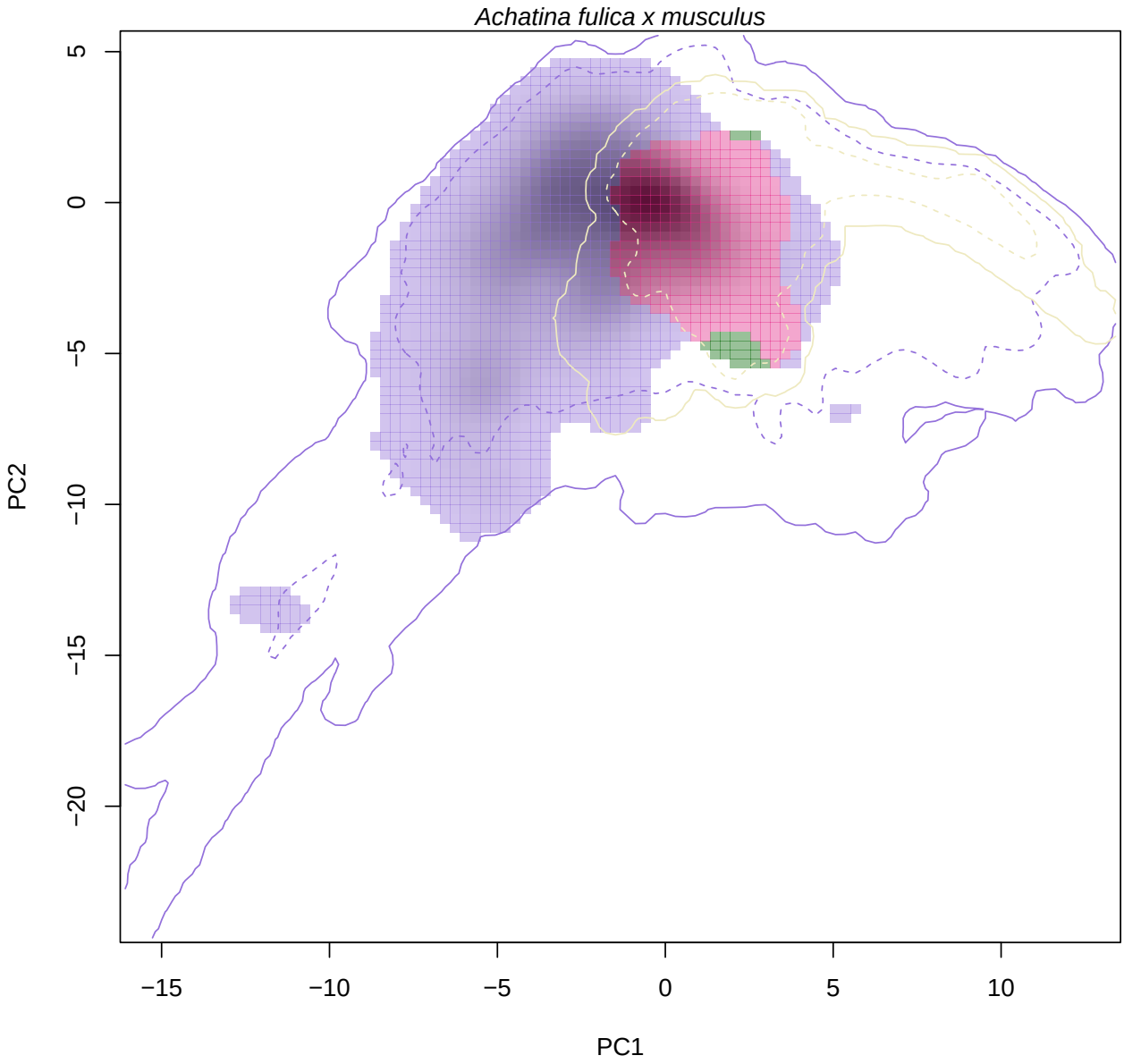
Regular plot:

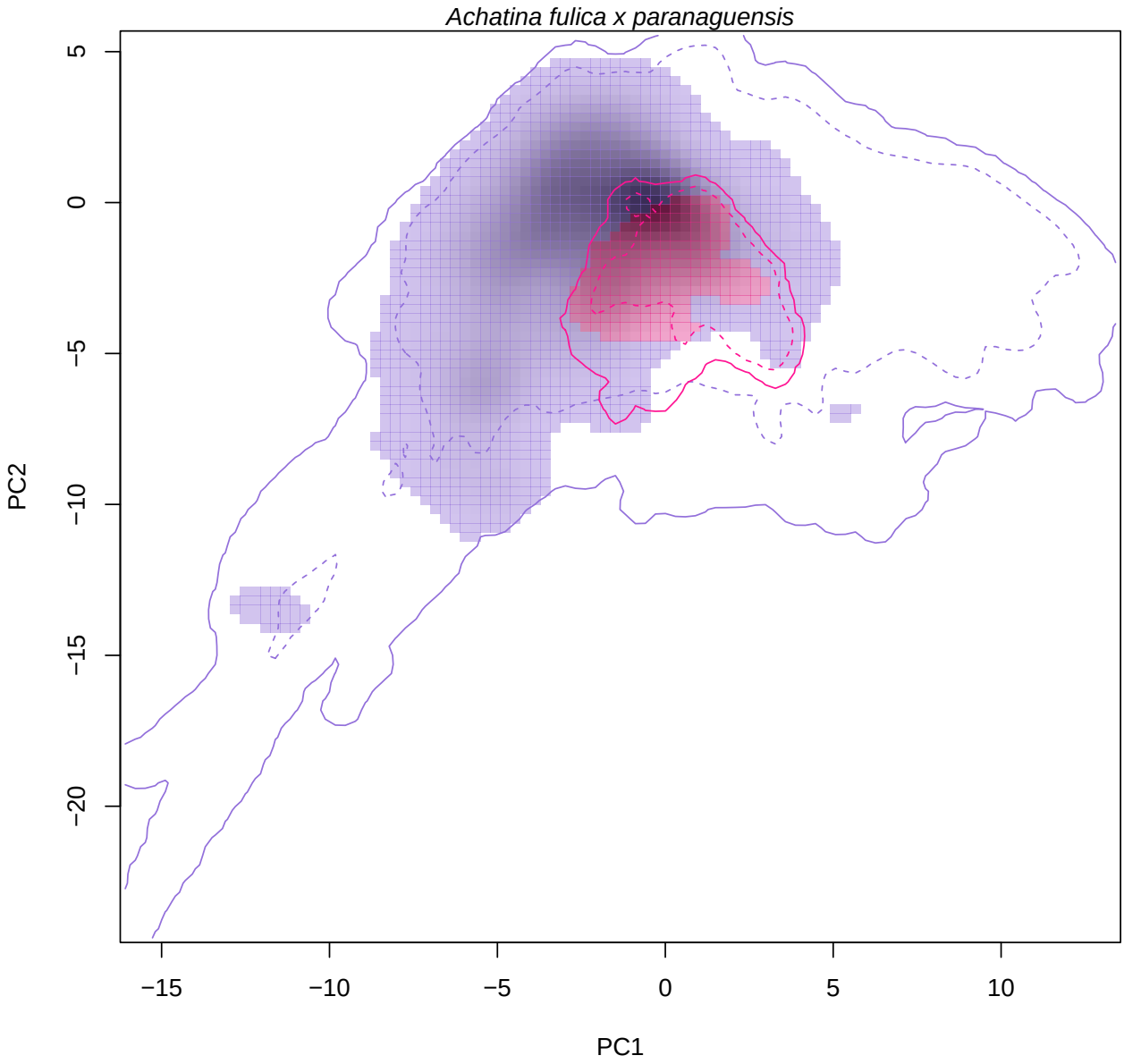
```
col1 <- colorRampPalette(c(desaturate(g.colors[1]), g.colors[1]))(5)
col2 <- colorRampPalette(c(desaturate(g.colors[2]), g.colors[2]))(5)
col3 <- colorRampPalette(c(desaturate(g.colors[3]), g.colors[3]))(5)
col_int <- colorRampPalette(c(desaturate('#e7298a'), '#e7298a'))(5)

for (i in 2:n.groups) {
  ecospat.plot.niche.dyn(z[[1]], z[[i]], .5,
                         colZ1 = g.colors[1],
                         colZ2 = g.colors[i],
                         colinter = adjustcolor(col_int, .4),
                         colz1 = adjustcolor(col1, .4),
                         colz2 = adjustcolor(col2, .4),
                         name.axis1 = "PC1",
                         name.axis2 = "PC2")
  mtext(paste(species[1],"x", species[i]), font = 3)
  }
```

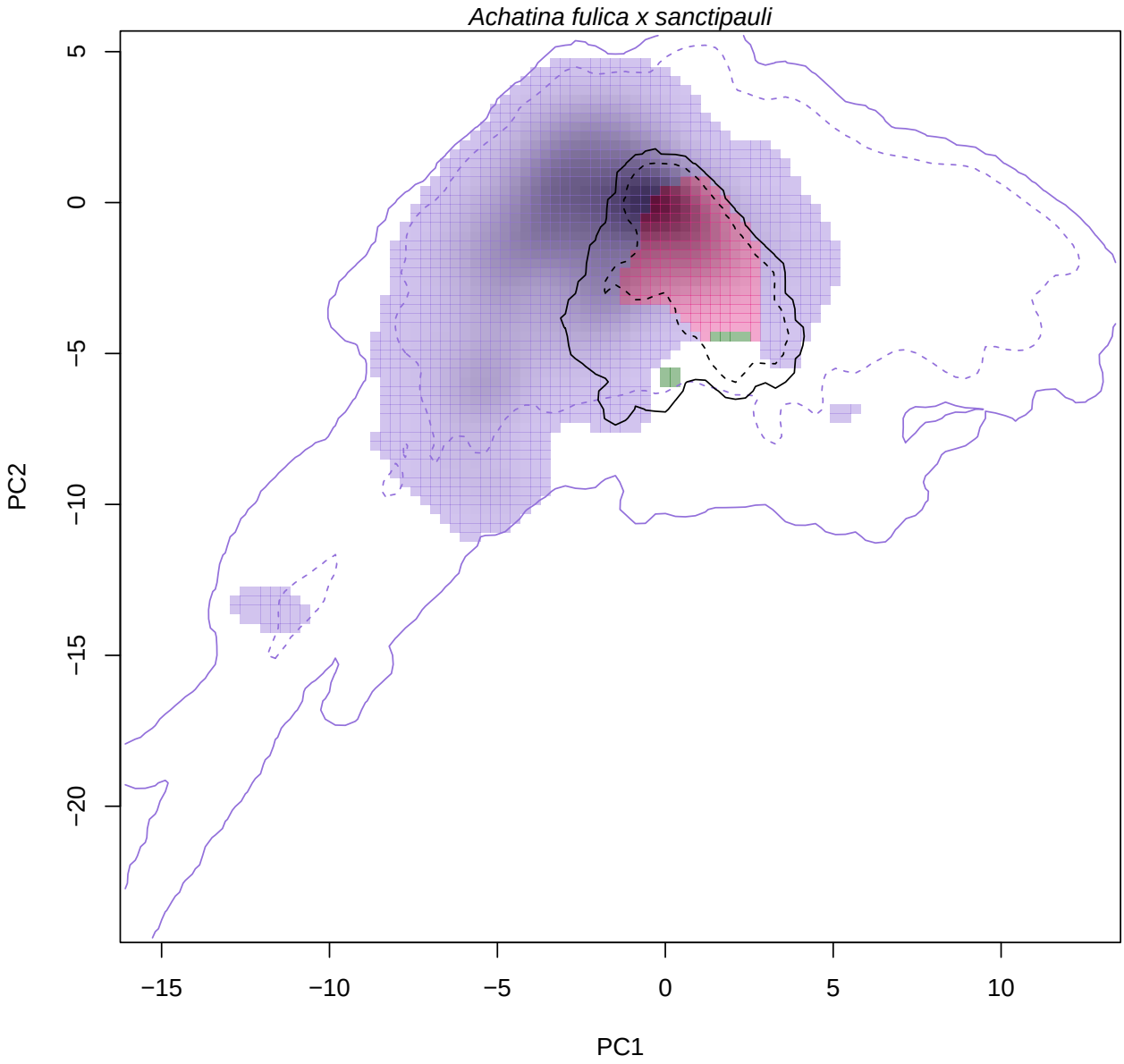

Achatina fulica x elongatus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

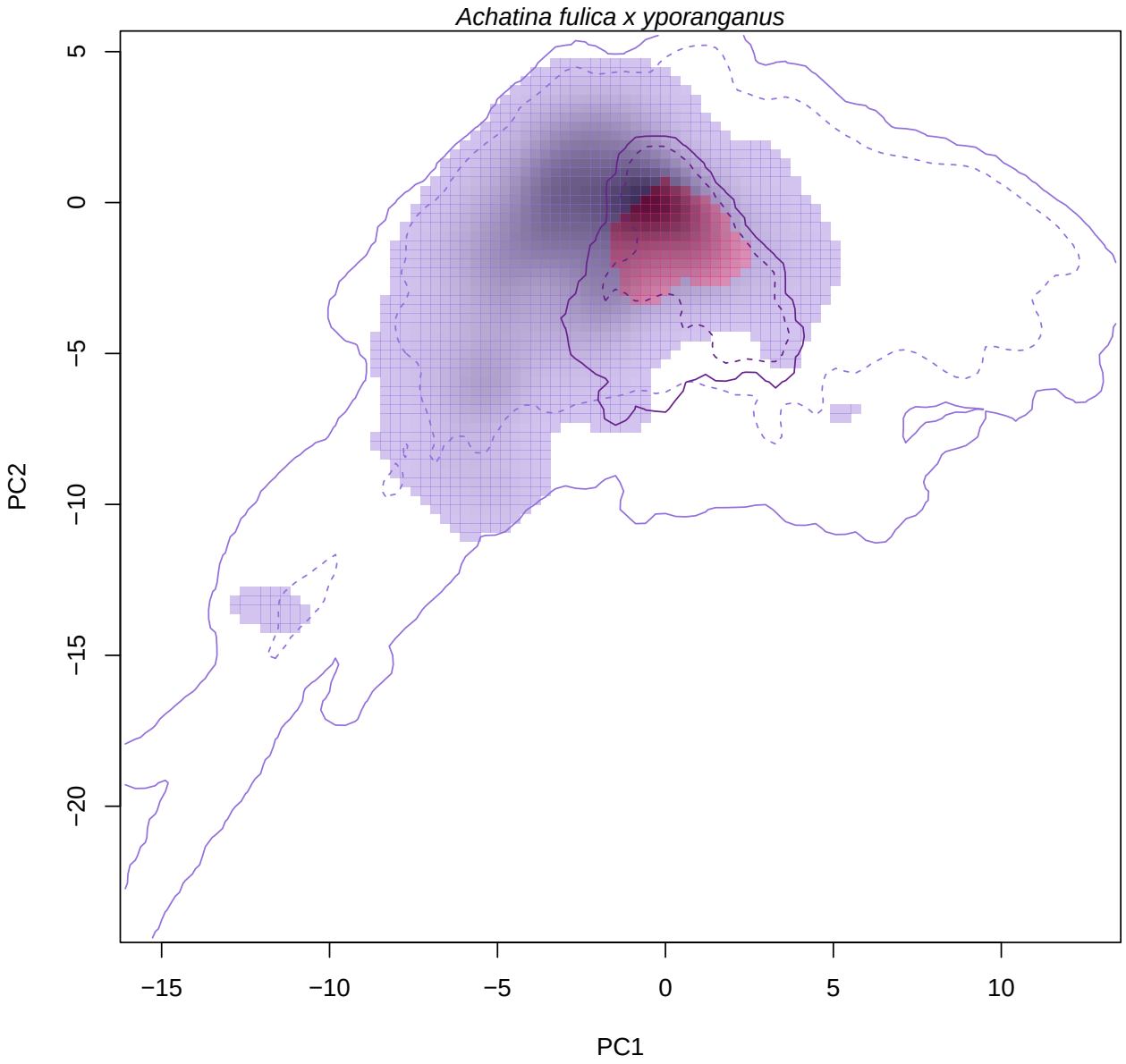

Achatina fulica x granulosus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

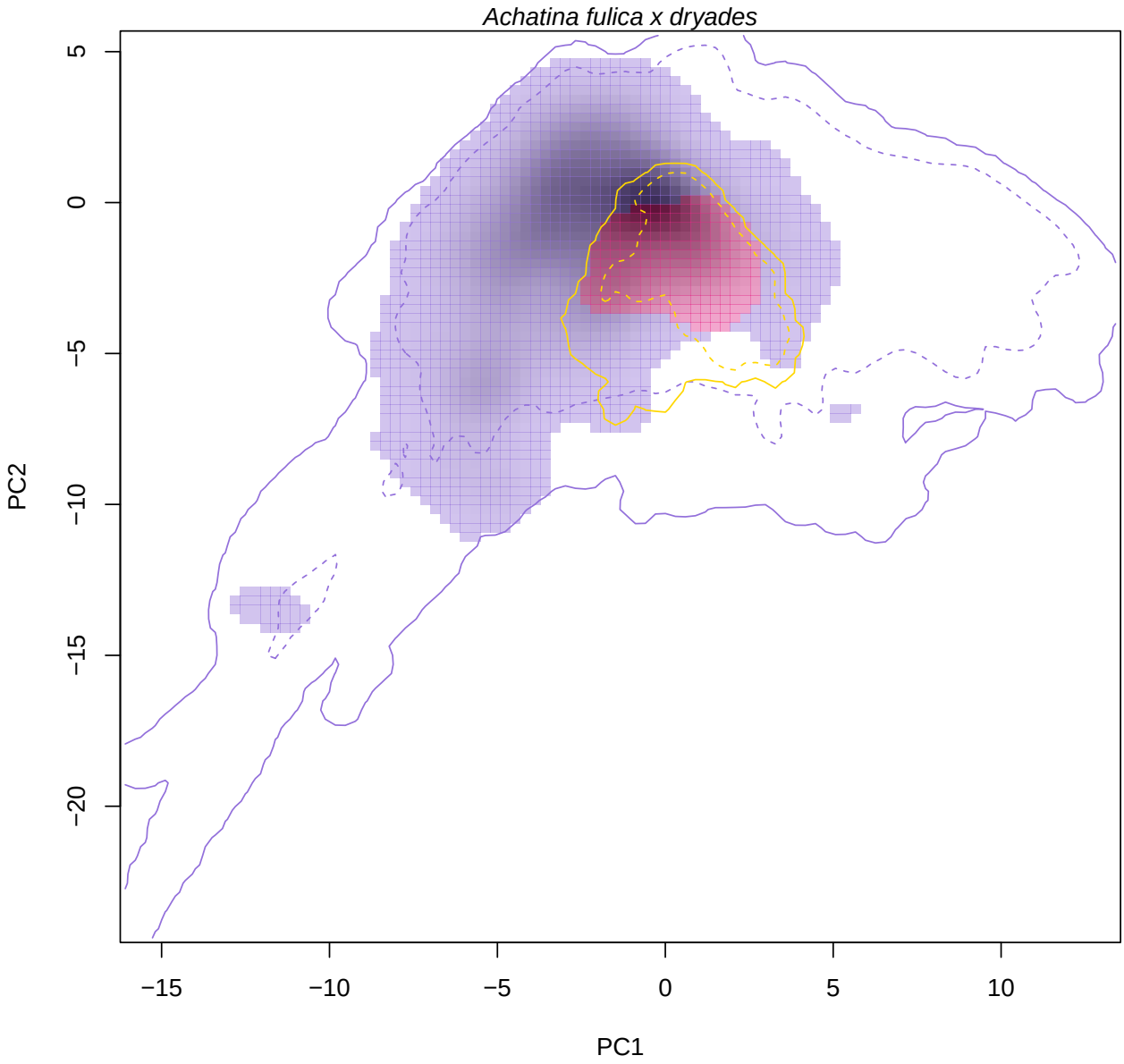

Achatina fulica x haemastomus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

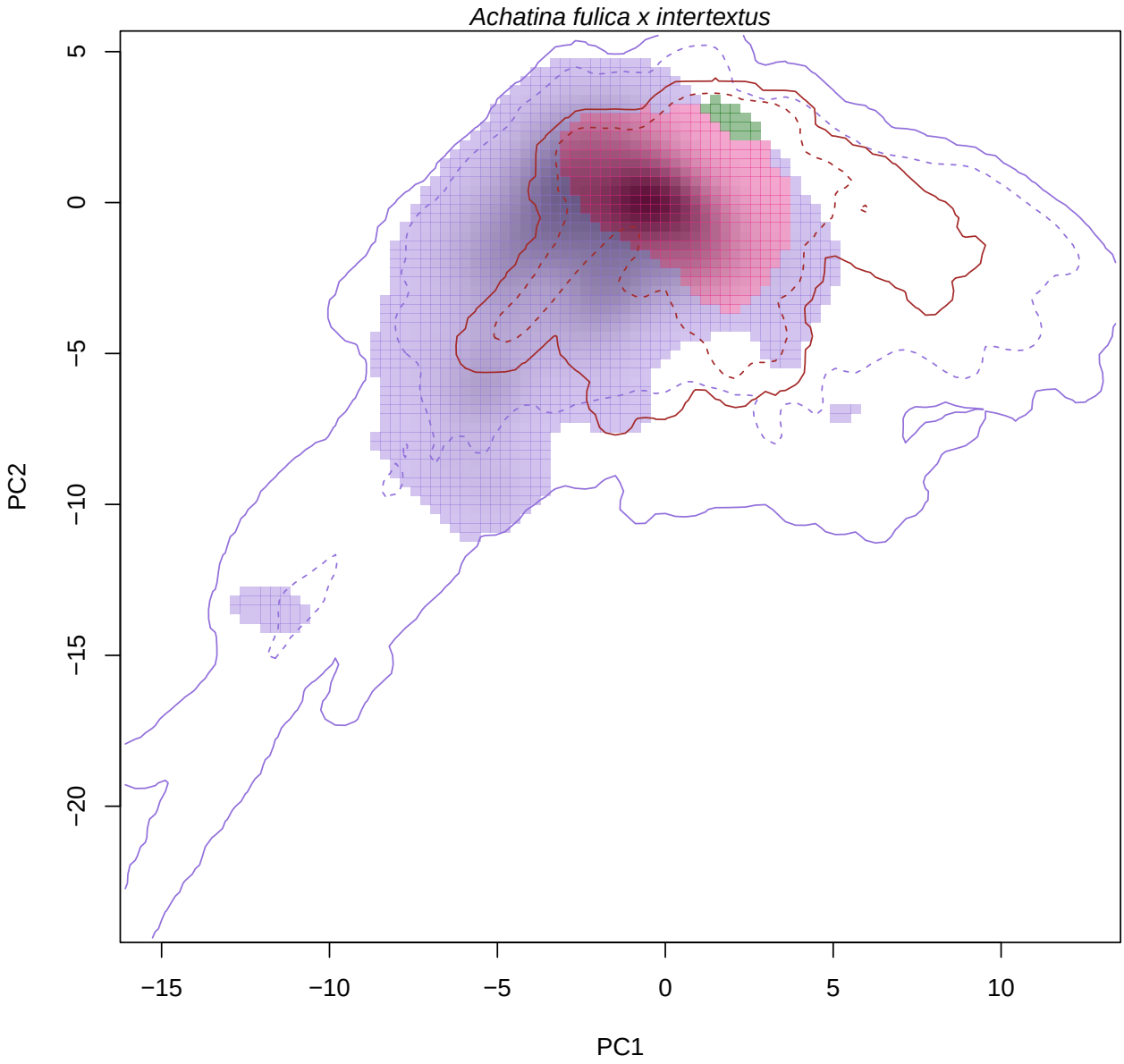

Achatina fulica x musculus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

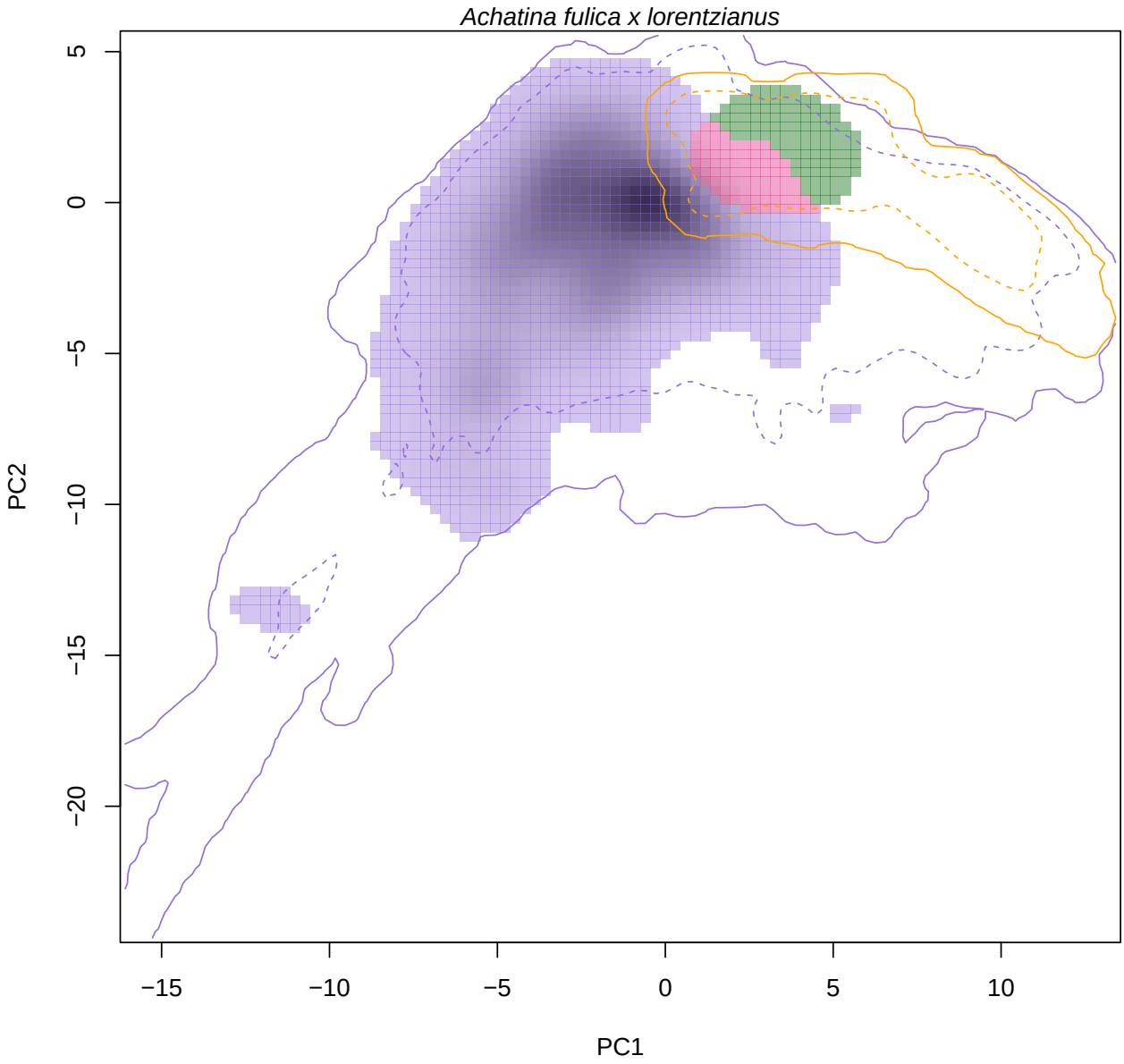

Achatina fulica x paranaguensis
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1


Achatina fulica x sanctipauli
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1


Achatina fulica x yporanganus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

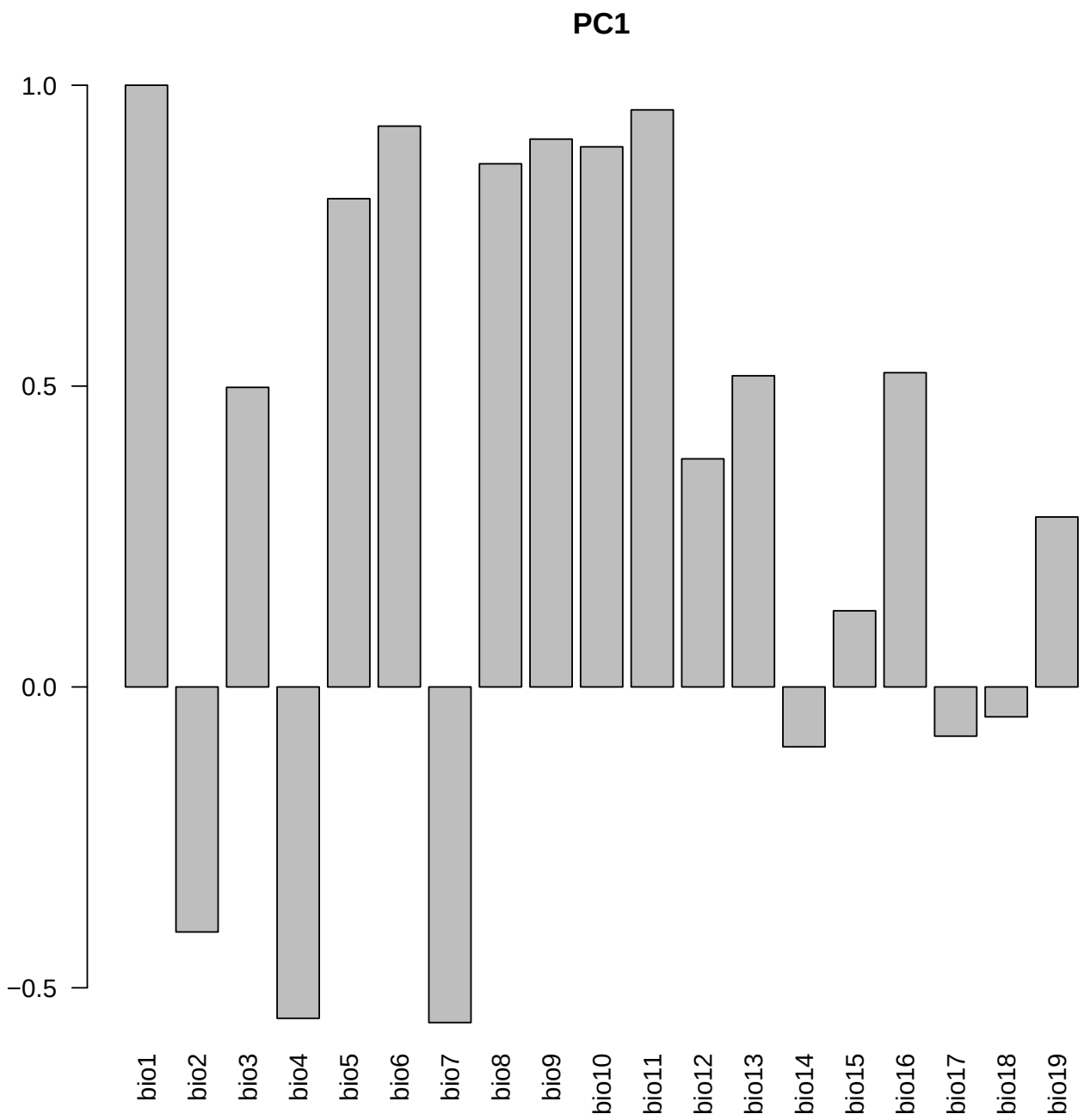

Achatina fulica x dryades
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1

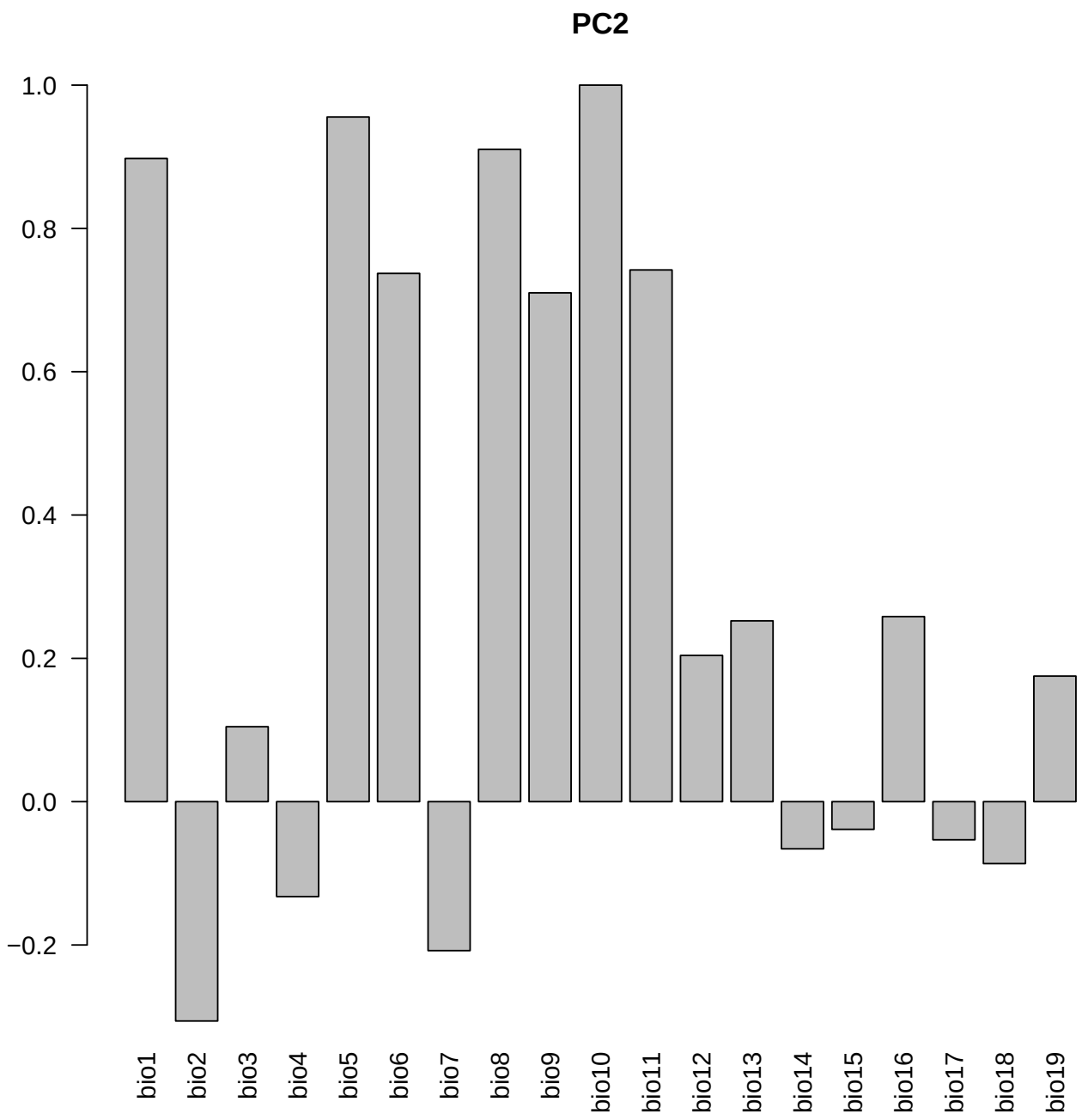

Achatina fulica x intertextus
PC2
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
PC1



Below the loadings plot (contribution of the variables for each axis). Check the variable codes at http://www.worldclim.org/bioclim.

```
loadings <- cbind(cor(data.env, pca.cal$tab[,1]), cor(data.env, pca.cal$tab[,2]))
colnames(loadings) <- c("axis1", "axis2")
loadings <- loadings[c(1, 12:19, 2:11), ]

barplot(loadings[,1], las=2, main="PC1")

barplot(loadings[,2], las=2, main="PC2")
```

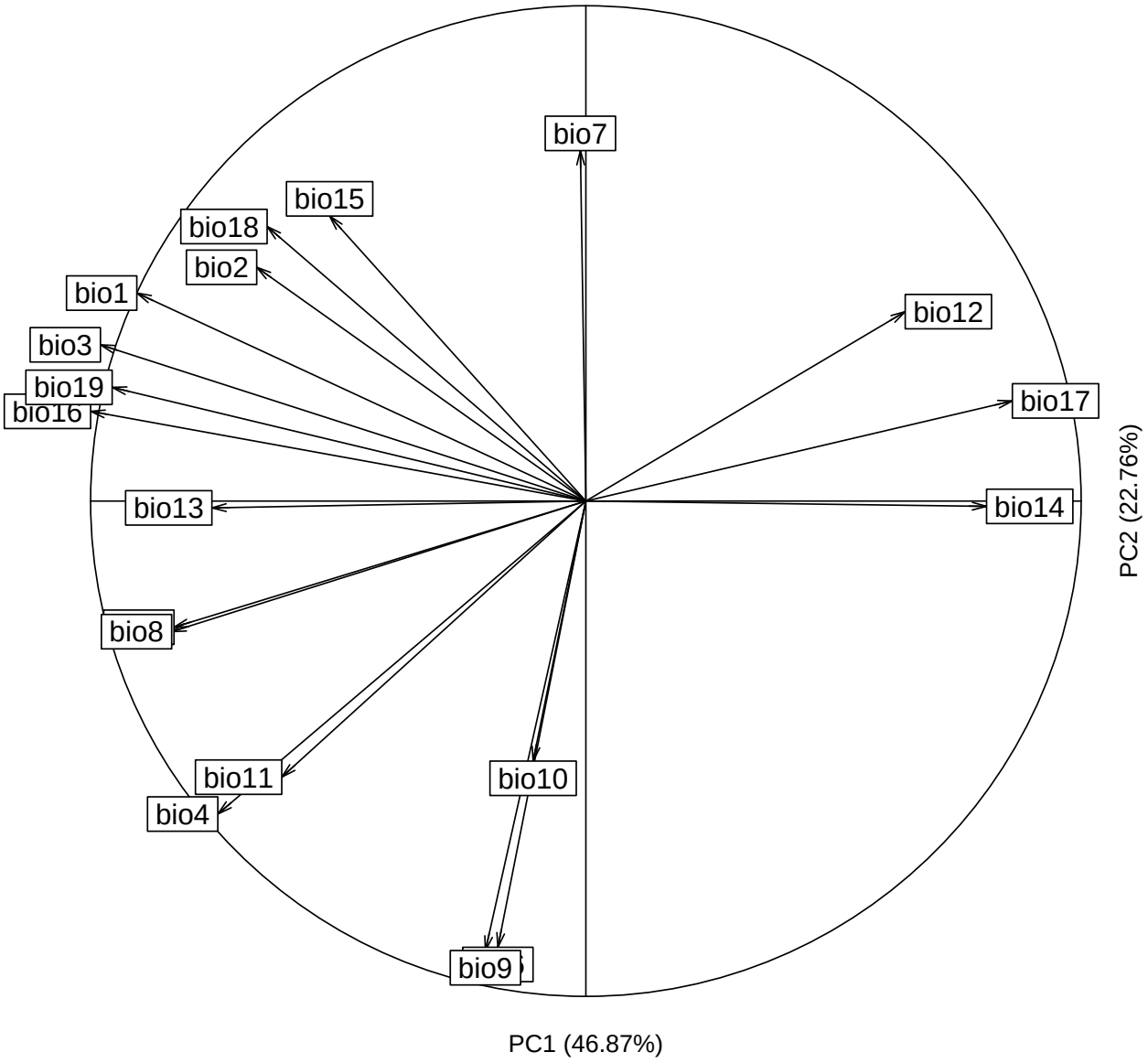

PC1
1.0
0.5
0.0
−0.5
bio1
bio2
bio3
bio4
bio5
bio6
bio7
bio8
bio9
bio10
bio11
bio12
bio13
bio14
bio15
bio16
bio17
bio18
bio19



The arrows representing the contribution of each variable, directly on the environmental space.

```
contrib <- pca.cal$co
eigen <- pca.cal$eig
nomes <- numeric(19)
for(i in 1:19){
  nomes[i] <- paste('bio',i, sep="")
}
s.corcircle(contrib[, 1:2] / max(abs(contrib[, 1:2])),
            grid = F,  label = nomes, clabel = 1.2)
text(0, -1.1, paste("PC1 (", round(eigen[1]/sum(eigen)*100,2),"%)",
                    sep = ""))
text(1.1, 0, paste("PC2 (", round(eigen[2]/sum(eigen)*100,2),"%)",
```

```
                    sep = ""), srt = 90)
```



References

Aiello-Lammens, M. E., Boria, R. A., Radosavljevic, A., Vilela, B., & Anderson, R. P. (2015). spThin: an R package for spatial thinning of species occurrence records for use in ecological niche models. Ecography, 38(5), 541-545.

Broennimann, O., Fitzpatrick, M. C., Pearman, P. B., Petitpierre, B., Pellissier, L., Yoccoz, N. G., & Guisan, A. (2012). Measuring ecological niche overlap from occurrence and spatial environmental data. Global Ecology and Biogeography, 21(4), 481-497.