

Protected Areas do not Outweigh Species Richness or Functional Diversity and Traits of Birds in Brazil

Supplemental materials. Section 2. R codes used for analyses.

```
library(ggplot2)
library(vegan)
library(dplyr)
library(FD)
library(ggpubr)
library(tidyr)

### ACCUMULATION CURVES ###
#####

f1<-read.csv("pr_sp.csv", sep=";", header=T)

# Anhembi
anh<-f1[f1$localidade == "Anhembi",]
curva.anh<-table(as.numeric(anh$lists),anh$sp)
acum.anh <- specaccum(curva.anh)

# Barra do Jacaré
BdJ<-f1[f1$localidade == "Barra do Jacaré",]
curva.bdj<-table(as.numeric(BdJ$lists),BdJ$sp)
acum.bdj <- specaccum(curva.bdj)

# Caiuá
cai<-f1[f1$localidade == "Caiuá",]
curva.cai<-table(as.numeric(cai$lists),cai$sp)
acum.cai <- specaccum(curva.cai)

# Centenário do Sul
cen<-f1[f1$localidade == "Centenário do Sul",]
curva.cen<-table(as.numeric(cen$lists),cen$sp)
acum.cen <- specaccum(curva.cen)

# Cornélio Procópio
CP<-f1[f1$localidade == "Cornélio Procópio",]
curva.cp<-table(as.numeric(CP$lists),CP$sp)
acum.cp <- specaccum(curva.cp)

# Diamante do Norte
dia<-f1[f1$localidade == "Diamante do Norte",]
curva.dia<-table(as.numeric(dia$lists),dia$sp)
acum.dia <- specaccum(curva.dia)

# Ibiporã
Ibpr<-f1[f1$localidade == "Ibiporã",]
curva.ibpr<-table(as.numeric(Ibpr$lists),Ibpr$sp)
acum.ibpr <- specaccum(curva.ibpr)
```

```

# Jacareziho (RVS)
RVS<-f1[f1$localidade == "Jacarezinho-RVS",]
curva.rvs<-table(as.numeric(RVS$lists),RVS$sp)
acum.rvs <- specaccum(curva.rvs)

# Jacarezinho
JAC<-f1[f1$localidade == "Jacarezinho",]
curva.jac<-table(as.numeric(JAC$lists),JAC$sp)
acum.jac <- specaccum(curva.jac)

# 19.Jundiaí
jun<-f1[f1$localidade == "Jundiaí",]
curva.jun<-table(as.numeric(jun$lists),jun$sp)
acum.jun <- specaccum(curva.jun)

# Londrina
Godoy<-f1[f1$localidade == "Londrina",]
curva.godoy<-table(as.numeric(Godoy$lists),Godoy$sp)
acum.godoy <- specaccum(curva.godoy)

# Porto Ferreira
por<-f1[f1$localidade == "Porto Ferreira",]
curva.por<-table(as.numeric(por$lists),por$sp)
acum.por <- specaccum(curva.por)

# Presidente Epitácio
pre<-f1[f1$localidade == "Presidente Epitácio",]
curva.pre<-table(as.numeric(pre$lists),pre$sp)
acum.pre <- specaccum(curva.pre)

# Santa Mariana
MSF<-f1[f1$localidade == "Santa Mariana",]
curva.msf<-table(as.numeric(MSF$lists),MSF$sp)
acum.msf <- specaccum(curva.msf)

# Santa Rita do Passa Quatro
san<-f1[f1$localidade == "Santa Rita do Passa Quatro",]
curva.san<-table(as.numeric(san$lists),san$sp)
acum.san <- specaccum(curva.san)

# Santo Antônio da Platina
SAP<-f1[f1$localidade == "Santo Antônio da Platina",]
curva.sap<-table(as.numeric(SAP$lists),SAP$sp)
acum.sap <- specaccum(curva.sap)

# Teodoro Sampaio
teo<-f1[f1$localidade == "Teodoro Sampaio",]
curva.teo<-table(as.numeric(teo$lists),teo$sp)
acum.teo <- specaccum(curva.teo)

```

FIGURE 2

#####

```
curvas_acum<-read.csv("acum.csv",sep=";",header = T)
curvas_acum$S<-as.numeric(curvas_acum$S)
curvas_acum$id<-as.character(curvas_acum$id)
str(curvas_acum)

# Convert 'id' to numeric
curvas_acum$id <- as.numeric(as.character(curvas_acum$id))
# Filter data for two ranges
curvas_acum_non_1_7 <- curvas_acum[curvas_acum$id >= 1 & curvas_acum$id <= 7, ]
curvas_acum_uc_8_17 <- curvas_acum[curvas_acum$id >= 8 & curvas_acum$id <= 17, ]

# Convert 'id' back to factor with numeric order
curvas_acum_non_1_7$id <- factor(curvas_acum_non_1_7$id, levels =
sort(unique(curvas_acum_non_1_7$id)))
curvas_acum_uc_8_17$id <- factor(curvas_acum_uc_8_17$id, levels =
sort(unique(curvas_acum_uc_8_17$id)))

# GRAPH

non<-ggplot(curvas_acum_non_1_7, aes(x = lists, y = S)) +
  geom_line(aes(color = id, linetype = id), show.legend = T)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Species richness",x="Number of lists")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  theme(legend.text = element_text(size=10),legend.position="right")+
  theme(legend.title=element_blank())+
  geom_ribbon(aes(ymin = S-sd, ymax = S+sd, fill =id),alpha = 0.3)+
  scale_x_continuous(limits = c(0, 50))+
  scale_y_continuous(limits = c(0, 80),breaks=seq(0, 70, by=10))+
  labs(color = "", linetype = "")+
  scale_color_discrete(breaks = sort(as.numeric(as.character(rownames(id)))))

uc<-ggplot(curvas_acum_uc_8_17, aes(x = lists, y = S)) +
  geom_line(aes(color = id, linetype = id), show.legend = T)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="",x="Number of lists")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  theme(legend.text = element_text(size=10),legend.position="right")+
  theme(legend.title=element_blank())+
  geom_ribbon(aes(ymin = S-sd, ymax = S+sd, fill =id),alpha = 0.3)+
  scale_x_continuous(limits = c(0, 50))+
```

```

scale_y_continuous(limits = c(0, 80),breaks=seq(0, 70, by=10))+
labs(color = "", linetype = "")+
scale_color_discrete(breaks = sort(as.numeric(as.character(rownames(id)))))

```

```

### CHAO #####
#####

```

```

chao.anh <- poolaccum(curva.anh, permutations = 100)
chao.bdj <- poolaccum(curva.bdj, permutations = 100)
chao.cai <- poolaccum(curva.cai, permutations = 100)
chao.cen <- poolaccum(curva.cen, permutations = 100)
chao.cp <- poolaccum(curva.cp, permutations = 100)
chao.dia <- poolaccum(curva.dia, permutations = 100)
chao.ibpr <- poolaccum(curva.ibpr, permutations = 100)
chao.rvs <- poolaccum(curva.rvs, permutations = 100)
chao.jac <- poolaccum(curva.jac, permutations = 100)
chao.jun <- poolaccum(curva.jun, permutations = 100)
chao.godoy <- poolaccum(curva.godoy, permutations = 100)
chao.por <- poolaccum(curva.por, permutations = 100)
chao.pre <- poolaccum(curva.pre, permutations = 100)
chao.msf <- poolaccum(curva.msf, permutations = 100)
chao.san <- poolaccum(curva.san, permutations = 100)
chao.sap <- poolaccum(curva.sap, permutations = 100)
chao.teo <- poolaccum(curva.teo, permutations = 100)

```

```

#####
### FUNCTIONAL TRAITS ###
#####

```

```

func<-read.csv("func_div.csv",sep=";",header=T,row.names = 1)
str(func)
func$strat<-as.factor(func$strat)
func$micro<-as.factor(func$micro)
head(func)
locais<-read.csv("sites.csv",sep=";",header=T,row.names = 1)
str(locais)
x <- list("func"=func, "locais"=locais)
str(x)

```

```

res <- dbFD(x$func, x$locais, w.abun = TRUE, print.pco = TRUE, corr = "cailliez", m="min")
str(res)
index<-data.frame(res$FRic,res$FEve, res$FDiv, res$FDis)

```

```

### FIGURE 4 ###

```

```

grafico<-read.csv("frics.csv",sep=";",header=T)

```

```

# Custom label function
custom_labels <- function(x) {

```

```

    ifelse(x >= 10000, format(x, big.mark = ",", scientific = FALSE), as.character(x))
  }

func_fric<-ggplot(grafico, aes(x = Area, y = FRic)) +
  geom_point(aes(shape = protected), size = 3) +
  theme_bw() +
  guides(fill = guide_legend(title = "")) +
  theme(
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.text = element_text(size = 12)
  ) +
  labs(y = "Functional richness", x = "") +
  scale_x_continuous(trans = 'log10', labels = custom_labels) +
  scale_y_continuous(trans = 'log10') +
  theme(legend.position = "none") +
  geom_smooth(method = lm, col = 'black', linewidth = 1, level = 0.95) +
  stat_regline_equation(label.y = -0.3, aes(label = after_stat(rr.label)))

func_feve<-ggplot(grafico,aes(x=Area,y=FEve)) + geom_point(aes(shape=protected),size=3)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Functional evenness",x="")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  scale_x_continuous(trans = 'log10', labels = custom_labels)+
  theme(legend.position="none")+
  geom_smooth(method=lm, col='black', linewidth=1, level=0.95)+
  stat_regline_equation(label.y = 0.6, aes(label = after_stat(rr.label))) #shows R2

func_fdiv<-ggplot(grafico,aes(x=Area,y=FDiv)) + geom_point(aes(shape=protected),size=3)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Functional divergence",x="Fragment area (log)")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  scale_x_continuous(trans = 'log10', labels = custom_labels)+
  theme(legend.position="none")+
  geom_smooth(method=lm, col='black', linewidth=1, level=0.95)+
  stat_regline_equation(label.y = 0.90, aes(label = after_stat(rr.label))) #shows R2

func_fdis<-ggplot(grafico,aes(x=Area,y=FDis)) + geom_point(aes(shape=protected),size=3)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Functional dispersion",x="Fragment area (log)")

```

```

theme(axis.text=element_text(size=12),
      axis.title=element_text(size=14),legend.text = element_text(size=12))+
scale_x_continuous(trans = 'log10', labels = custom_labels)+
theme(legend.position="none")+
geom_smooth(method=lm, col='black', linewidth=1, level=0.95)+
stat_regline_equation(label.y = 0.25, aes(label = after_stat(rr.label)))

func<-ggarrange(func_fric,func_feve,func_fdiv,func_fdis,
               labels = c("(a)", "(b)","(c)","(d)"),
               ncol = 2, nrow = 2)

#####

func<-ggarrange(func_fric,func_feve,func_fdiv,func_fdis,
               labels = c("(a)", "(b)","(c)","(d)"),
               ncol = 2, nrow = 2)

#####
### FIGURE S1 ###
#####

trait<-read.csv("traits.csv",header=T,sep=";")
head(trait)
# median, min, and max
results <- trait %>%
  filter(uc == "S") %>%
  summarise(
    median_value = median(body_mass),
    min_value = min(body_mass),
    max_value = max(body_mass)
  )

str(trait)
trait$localidade<-as.character(trait$localidade)
str(trait)
trait.guild<-data.frame(table(trait$localidade,trait$func))
trait.strat<-data.frame(table(trait$localidade,trait$strat))
trait.micro<-data.frame(table(trait$localidade,trait$micro))
trait.micro$Freq<-na.omit(trait.micro$Freq)

trait.guild$Var1 <- factor(trait.guild$Var1, levels = c("1", "2", "3","4","5",
                                                    "6","7","8","9","10",
                                                    "11","12","13","14","15","16","17"))
trait.strat$Var1 <- factor(trait.strat$Var1, levels = c("1", "2", "3","4","5",
                                                    "6","7","8","9","10",
                                                    "11","12","13","14","15","16","17"))
trait.micro$Var1 <- factor(trait.micro$Var1, levels = c("1", "2", "3","4","5",
                                                    "6","7","8","9","10",
                                                    "11","12","13","14","15","16","17","18"))
trait$localidade <- factor(trait$localidade, levels = c("1", "2", "3","4","5",
                                                    "6","7","8","9","10",
                                                    "11","12","13","14","15","16","17"))

```

```

guild<-ggplot(trait.guild, aes(x = Var1, y = Freq,fill=Var2))+geom_bar(stat="identity")+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Species richness",x="")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  theme(legend.text = element_text(size=10),legend.position=c(0.13, 0.85))+
  theme(legend.title=element_blank())+
  geom_vline(xintercept=9.5, linetype='dashed')

strat<-ggplot(trait.strat, aes(x = Var1, y = Freq,fill=Var2))+geom_bar(stat="identity")+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Species richness",x="")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  theme(legend.text = element_text(size=10),legend.position=c(0.13, 0.85))+
  theme(legend.title=element_blank())+
  geom_vline(xintercept=9.5, linetype='dashed')

micro<-ggplot(trait.micro, aes(x = Var1, y = Freq,fill=Var2))+geom_bar(stat="identity")+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Species richness",x="Forest fragments")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  theme(legend.text = element_text(size=10),legend.position=c(0.13, 0.8))+
  theme(legend.title=element_blank())+
  geom_vline(xintercept=9.5, linetype='dashed')

bm<-ggplot(trait,aes(x=localidade,y=body_mass)) + geom_boxplot()+
  theme_bw()+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Body mass (g)",x="Forest fragments")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))+
  scale_y_continuous(trans='log10')+
  geom_vline(xintercept=9.5, linetype='dashed')

comparison<-ggarrange(guild, strat, micro,bm,
  labels = c("(a)", "(b)","(c)","(d)"),
  ncol = 2, nrow = 2)

#####
### FIGURE 3 ###
#####
norte<-read.csv("pr_sp_frag.csv",sep=";",header=T)

```

```

head(norte)
#mann whitney estimated species
wilcox.test(chaof1 ~ protected, data=norte)

bar_chao<-ggplot(norte, aes(x = protected, y = chaof1)) +
  geom_boxplot() +
  geom_jitter(width = 0.2, shape = 21, fill = "white", size = 2) + # Adiciona pontos com jitter
  theme_bw() +
  guides(fill = guide_legend(title = "")) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(y = "Species richness", x = "Protection") +
  theme(axis.text = element_text(size = 12),
        axis.title = element_text(size = 14), legend.text = element_text(size = 12))

#####
### FIGURE 5 ###
#####

func_prot_fric<-ggplot(grafico,aes(x=protected,y=FRic)) + geom_boxplot()+
  geom_jitter(width = 0.2, shape = 21, fill = "white", size = 2)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Functional richness",x="Protected")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))

func_prot_feve<-ggplot(grafico,aes(x=protected,y=FVe)) + geom_boxplot()+
  geom_jitter(width = 0.2, shape = 21, fill = "white", size = 2)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Functional evenness",x="Protected")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))

func_prot_fdiv<-ggplot(grafico,aes(x=protected,y=FDiv)) + geom_boxplot()+
  geom_jitter(width = 0.2, shape = 21, fill = "white", size = 2)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+
  labs(y="Functional divergence",x="Protected")+
  theme(axis.text=element_text(size=12),
        axis.title=element_text(size=14),legend.text = element_text(size=12))

func_prot_fdis<-ggplot(grafico,aes(x=protected,y=FDis)) + geom_boxplot()+
  geom_jitter(width = 0.2, shape = 21, fill = "white", size = 2)+
  theme_bw()+
  guides(fill = guide_legend(title = ""))+
  theme(panel.grid.major = element_blank(),panel.grid.minor = element_blank())+

```



```

labs(y="Functional dispersion",x="Protected")+
theme(axis.text=element_text(size=12),
      axis.title=element_text(size=14),legend.text = element_text(size=12))

func_prot<-ggarrange(func_prot_fric,func_prot_feve,func_prot_fdiv,func_prot_fdis,
                    labels = c("(a)", "(b)", "(c)", "(d)"),
                    ncol = 2, nrow = 2)

### ANALYSES ###
#####

### MANOVA functional diversity ###
model <- manova(cbind(chao,FRic,FEve,FDiv,FDis)
               ~ protected, data = grafico)
summary(model,tol=0)
summary.aov(model)

# chi-square trophic
chi<-read.csv("chi.csv",sep=";",header=T)

# Create a contingency table for func
contingency_table <- chi %>%
  group_by(UC, func) %>%
  summarise(species_count = n_distinct(sp), .groups = 'drop') %>% # Count unique species per
UC-func combination
  pivot_wider(names_from = func, values_from = species_count, values_fill = list(species_count =
0))
# chi-squared test
chi_squared_result <- chisq.test(contingency_table[,-1]) # Exclude the UC column

# Create a contingency table for strata
contingency_table <- chi %>%
  group_by(UC, strata) %>%
  summarise(species_count = n_distinct(sp), .groups = 'drop') %>%
  pivot_wider(names_from = strata, values_from = species_count, values_fill = list(species_count
= 0))
# chi-squared test
chi_squared_result <- chisq.test(contingency_table[,-1]) # Exclude the UC column

# Create a contingency table for microhabitat
contingency_table <- chi %>%
  group_by(UC, micro) %>%
  summarise(species_count = n_distinct(sp), .groups = 'drop') %>%
  pivot_wider(names_from = micro, values_from = species_count, values_fill = list(species_count
= 0))
# chi-squared test
chi_squared_result <- chisq.test(contingency_table[,-1]) # Exclude the UC column

###
# MANOVA functional traits

```

```

trophic<-aggregate(func ~ localidade, data = trait, length)
trophic$protected<-
c("No","No","No","No","No","No","No","Yes","Yes","Yes","Yes","Yes","Yes",
  "Yes","Yes","Yes")

strata<-aggregate(strat ~ localidade, data = trait, length)
strata$protected<-
c("No","No","No","No","No","No","No","Yes","Yes","Yes","Yes","Yes","Yes",
  "Yes","Yes","Yes")

habitat<-aggregate(micro ~ localidade, data = trait, length)
habitat$protected<-
c("No","No","No","No","No","No","No","Yes","Yes","Yes","Yes","Yes","Yes",
  "Yes","Yes","Yes")

body<-aggregate(body_mass ~ localidade, data = trait, mean)
body$protected<-
c("No","No","No","No","No","No","No","Yes","Yes","Yes","Yes","Yes","Yes",
  "Yes","Yes","Yes")

modeltrait<-read.csv("modeltrait.csv",sep=";",header=T)
modeltrait2<-modeltrait[complete.cases(modeltrait), ]#remove NAs
correlation_matrix <- cor(modeltrait2[, c("func", "body_mass", "micro", "strat")], use =
"complete.obs")
print(correlation_matrix)#no redundancy

model2 <- manova(cbind(func, body_mass, micro,strat)
  ~ protected, data = modeltrait2[,-1])
summary(model2,tol=0)

# ANOVA post-hoc
anova_func <- aov(func ~ protected, data = modeltrait2)
anova_body_mass <- aov(body_mass ~ protected, data = modeltrait2)
anova_micro <- aov(micro ~ protected, data = modeltrait2)
anova_strat <- aov(strat ~ protected, data = modeltrait2)

# Summary of each ANOVA
summary(anova_func)
summary(anova_body_mass)
summary(anova_micro)
summary(anova_strat)

#####
### BODY MASS
#####
wilcox.test(body_mass ~ uc, data=trait)

#####
### GLM ###
#####
df<-read.csv("glm.csv",sep=";",header=T)

```

```
model <- glm(Chao ~ Elevation + Area + Protected +  
             Elevation:Protected + Area:Protected,  
             data = df,  
             family = poisson())  
summary(model)
```

```
#####  
### END ###  
#####
```