Spatiotemporal interactions between ocelots and agoutis analysis

2024-12-10

The following vignette provides the steps to run the Bayesian 2-species occupancy analyses used to assess the mechanisms of co-existence of ocelots and agoutis in the Amazonian region (Vallejo-Vargas, et al.). In addition we included the script to evaluate the model with the Bayesian posterior predictive check. The Baysian model is fit using package nimble (Valpine et al. (2017)).

Load Libraries

Load necessary libraries.

```r
library(tibble)
library(coda)
library(nimble)
library(ggplot2)
library(R.utils)
library(dplyr)
library(reshape2)
library(MCMCvis)
library(tidyr)
library(ggstance)
library(ggpubr)
library(egg)
library(tinytex)
```

Load input data

We load the file containing the objects used as input for the model.

```r
load("Ocelot_agouti_data2.RData")
```

Source additional functions

We load additional custom functions for processing MCMC output.

```r
source("functions.R")
```

Define model

We define the model with the nimble package (Valpine et al. (2017)) to fit Bayesian model. Then we use the model "modelCodeTEAM" to determine the effect of agoutis occurrence on ocelots occupancy, simultaneously we use the model to assess the effect of ocelot occurrence on agoutis detection in three diel periods (day, night, and twilight).

```
modelCodeTEAM <- nimbleCode({

  #---Priors for the interaction terms

  for (sp in 1:n.species) {
    beta.p.night[sp]  ~ dlogis(0, 1)
    beta.p.twilight[sp]  ~ dlogis(0, 1)
    beta.p.psi[sp]  ~ dlogis(0, 1)
    beta.p.psi.night[sp]   ~ dlogis(0, 1)
    beta.p.psi.twilight[sp]   ~ dlogis(0, 1)
    beta.psi.psi[sp] ~ dlogis(0, 1)

  }#sp


  for (sp in 1:n.species) {
    for(pa in 1:3){
      for(prd in 1:n.periods[pa]){
        p0[sp,pa,prd] ~ dunif(0, 1)
        psi0[sp,pa,prd] ~ dunif(0, 1)
      }  #prd
    }#pa
  }#sp

   #---SPECIES 1 (ocelot)

  for (i in 1:n.sites) {

    ### Occupancy probability  (including effect of prey z on predator psi)
    psi[i, 1] <-
      ilogit(logit(psi0[1,PA[i],period[i]]) + beta.psi.psi[1] * z[i,2])


    for (j in 1:n.strata) {
      ### Detection probability
      p[i, 1, j] <- ilogit(logit(p0[1,PA[i],period[i]]) + # intercept
                             beta.p.night[1] * night[j] + # effect of night
                             beta.p.twilight[1] * twilight[j]) # effect of twilight

      ### Detections
      y[i, 1, j] ~ dbinom(z[i, 1] * p[i, 1, j], n.occasions[i, 1, j])


    }#j
    z[i, 1] ~ dbern(psi[i, 1])

  }#i


  #---SPECIES 2 (agouti)

  for (i in 1:n.sites) {
    ### Occupancy probability
    psi[i, 2] <-
      ilogit(logit(psi0[2,PA[i],period[i]])) #+ beta.psi.psi[2] * z[i,1])

    for (j in 1:n.strata) {
      ### Detection probability
      p[i, 2, j] <- ilogit(logit(
        p0[2,PA[i],period[i]]) + # intercept
        beta.p.night[2] * night[j] + # effect of night
        beta.p.twilight[2] * twilight[j] +  # effect of twilight
        beta.p.psi[2] * z[i, 1]  +  # effect of predator occurrence
        beta.p.psi.night[2] * z[i, 1] * night[j] + # effect of predator occurrence during
          #night
        beta.p.psi.twilight[2] * z[i, 1] * twilight[j] # effect of predator occurrence
        #during twilight
      )

      ### Detections
      y[i, 2, j] ~ dbinom(z[i, 2] * p[i, 2, j], n.occasions[i, 2, j])

    }#j
    z[i, 2] ~ dbern(psi[i, 2])

  }#i


})
```

Fit the nimble model

We set the number of iterations to 50000, and discard the first 30000 iterations. In addition we thin by 5 (retaining only every 5th iteration). Note that fitting will be slow when a high number of iterations is chosen.

```
runtimeTEAM <- system.time(
  nimOutputTEAM <- nimbleMCMC(
    code = modelCodeTEAM,
    constants = nimConstantsTEAM,
    data = nimDataTEAM,
    inits = nimInitsTEAM,
    monitors = c(
      "psi0",
      "p0",
      "z",
      "beta.p.psi",
      "beta.psi.psi",
      "beta.p.night",
      "beta.p.twilight",
      "beta.p.psi.night",
      "beta.p.psi.twilight"
    ),
    niter = 50000,
    nburnin = 30000,
    thin = 5, # takes only sample, or choose 5. It remove autocorrelation
    nchains = 3,
    samplesAsCodaMCMC = TRUE
  )
)
```

Process MCMC output

We process the output from nimbleMCMC for easier handling of the posterior samples.

The function ProcessCodaOutput also provides Rhat values for assessing convergence of the chains (Gelman (1996)). Rhat values < 1.1 indicate that convergence has been achieved.

```
## It is important to exclude NAs to calculate Rhat
ctD<-nimOutputTEAM[[1]]

## Extract the names where the values are different than NA
ct.3D<-names(which(!is.na(colSums(ctD))))

# Remove the  iteration where there are NAS

chainsD <- list()
chainsD[[1]] <- as.mcmc(nimOutputTEAM[[1]][, c(ct.3D)])
chainsD[[2]] <- as.mcmc(nimOutputTEAM[[2]][, c(ct.3D)])
chainsD[[3]] <- as.mcmc(nimOutputTEAM[[3]][, c(ct.3D)])

nimOutputTEAM2 <- as.mcmc.list(chainsD)

myNimbleOutputCoda <- ProcessCodaOutput(nimOutputTEAM2, DIC = F)
```

```
## Calculating statistics.......
##
## Done.
```

```
#Check Rhat values

#myNimbleOutputCodaD$Rhat

summary(as.numeric(myNimbleOutputCoda$Rhat$beta.p.twilight))
```

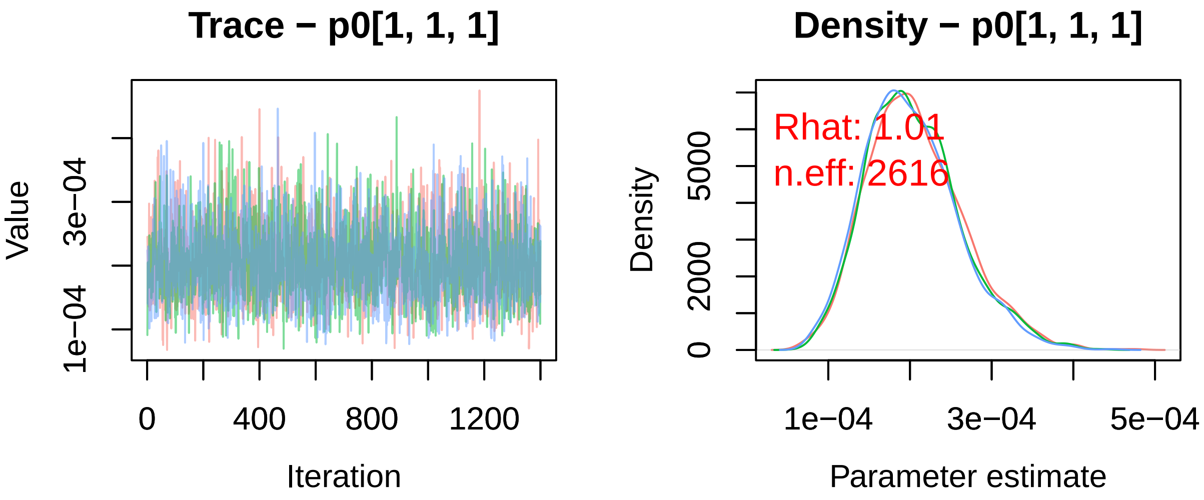
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.005   1.006   1.007   1.007   1.008   1.009
```

```
myNimbleOutputCoda2<-myNimbleOutputCoda
```

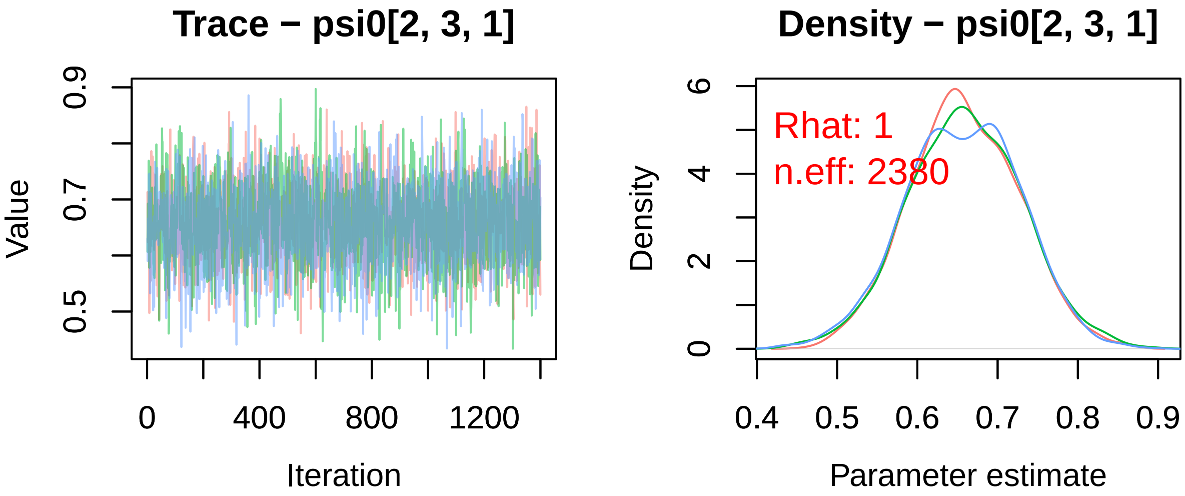
Trace plots.

Generate and inspect trace plots to visually assess convergence and mixing of the MCMC chains. Store plots if needed. We provide a few trace plots examples of the parameters of interest. To see all the trace plot of a parameter write only the parameter without indexes (e.g., "p0").

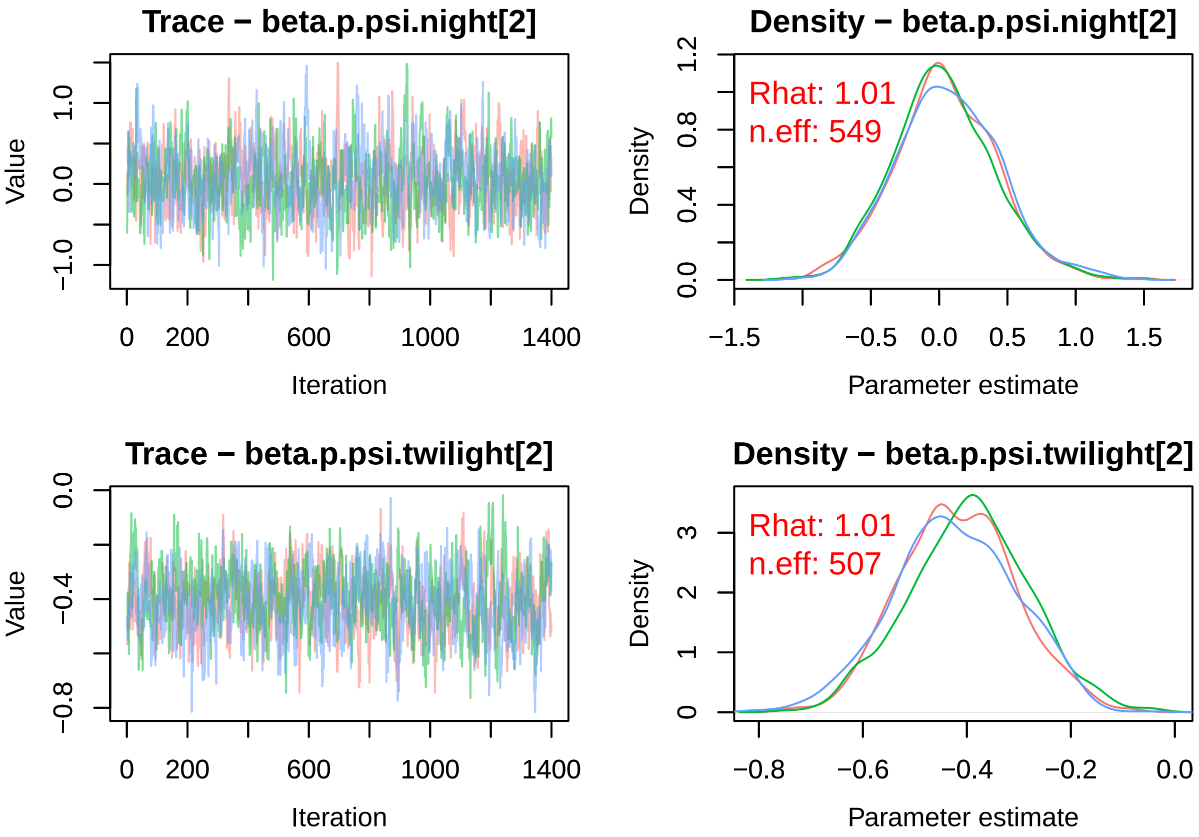
```r
#Check with MCMCvis
# MCMCtrace(object = nimOutputTEAM2,
#           pdf = FALSE,
#           ind = TRUE,
#           open_pdf = TRUE,
#           Rhat = TRUE, # add Rhat
#           n.eff = TRUE, # add eff sample size
#           #filename = "p0.pdf",
#           params = c("p0"))#,
#           )
# dev.off() # when storing the pdf use dev.off

MCMCtrace(object = nimOutputTEAM2,
          pdf = FALSE,
          ind = TRUE,
          open_pdf = TRUE,
          Rhat = TRUE, # add Rhat
          n.eff = TRUE, # add eff sample size
          #filename = "p0.pdf",
          params = "p0[1, 1, 1]",
          exact = TRUE,
          ISB = FALSE)
```



```r
MCMCtrace(object = nimOutputTEAM2,
          pdf = FALSE,
          ind = TRUE,
          open_pdf = TRUE,
          Rhat = TRUE, # add Rhat
          n.eff = TRUE, # add eff sample size
          #filename = "psiv.pdf",
          params = "psi0[2, 3, 1]", # Species 2, Protected area 3, Sampling Period 1
          exact = TRUE,
          ISB = FALSE)
```



```r
MCMCtrace(object = nimOutputTEAM2,
          pdf = FALSE,
          ind = TRUE,
          open_pdf = FALSE,
          Rhat = TRUE, # add Rhat
          n.eff = TRUE, # add eff sample size
          #filename = "betas.pdf",
          params = c(
            "beta.p.psi",
            "beta.psi.psi", "beta.p.night", "beta.p.psi.night[2]",
            "beta.p.psi.twilight[2]", "beta.p.twilight"),
            exact = TRUE,
            ISB = FALSE)
```



Extract parameters

Extract the occupancy of the predator accounting for the effect in the model “b.psi.psi”, with the presence or absence of the prey.

Ocelot occupancy - sampling year 1

```
z.list<-c(0,1) # presence or absence of the prey
z.prey<-1

pas<- length(unique(PA))
pa<-1;prd<-1
prediction.psi.pred<-data.frame()

for (pa in  1:pas) {

  prds<-sam.period[sam.period$PA.num == pa, ]$period.num

  prd<-1
  for(prd in 1:length(prds)) {

    prd.pa<- prds[prd]

    temp<-lapply(z.list,function(z.prey){
      out <- ilogit(
        logit(myNimbleOutputCoda2$sims.list$psi0[,1,pa,prd.pa]) +
          myNimbleOutputCoda2$sims.list$beta.psi.psi[,1] * z.prey
      )
      out<-tibble(psi=out,zprey=z.prey)
    })

    # do.call(rbind, temp)
    prediction.psi<-do.call(rbind,temp)
    prediction.psi$PA<-c(unique(PA)[pa])
    prediction.psi$period<-prd.pa
    prediction.psi.pred<-rbind(prediction.psi.pred, prediction.psi)
  }
}

## Estimate lower and upper confidence intervals
prediction.psi.pred$psi<-as.numeric(prediction.psi.pred$psi)

pred.psi.pred<-prediction.psi.pred %>% group_by(zprey,PA, period) %>%
  mutate(UCI=quantile(psi, 0.975))%>%
  mutate(LCI=quantile(psi, 0.025)) %>%
  mutate(mean.psi=mean(psi))

## Merge with the correct years
sam.period$period.num<-as.numeric(sam.period$period.num)

sam.period<-sam.period %>%  rename(YearCor=period)
sam.period<-as.data.frame(sam.period)

pred.psi.pred<-as.data.frame(pred.psi.pred)
pred.psi.pred$period<-as.numeric(pred.psi.pred$period)

pred.psi.pred<-pred.psi.pred %>%
               left_join(sam.period[-4], by=c("PA"="PA", "period"="period.num"))

means.pred.psi<-pred.psi.pred %>%
                select(zprey,LCI, UCI, mean.psi, PA, period, YearCor) %>%
                distinct()
```

Agouti occupancy - all sampling periods

```
pas<- length(unique(PA))
pa<-1;prd<-1
prediction.psi.prey<-data.frame()
PAs<-unique(PA)
#sam.period

for (pa in  1:pas) {

  prds<-sam.period[sam.period$PA.num == pa, ]$period.num
  prd<-1
  for(prd in 1:length(prds)) {

    prd.pa<- prds[prd]


    prediction.psi <- data.frame(psi=ilogit(
      logit(myNimbleOutputCoda2$sims.list$psi0[,2,pa,prd.pa])))

    prediction.psi$PA<-PAs[pa]
    prediction.psi$period<-prd.pa
    prediction.psi.prey<-rbind(prediction.psi.prey, prediction.psi)
  }
}

pred.psi.prey<-prediction.psi.prey %>% group_by(PA, period) %>%
  mutate(UCI=quantile(psi, 0.975))%>%
  mutate(LCI=quantile(psi, 0.025)) %>%
  mutate(mean.psi=mean(psi))

# Merge with the correct years
pred.psi.prey<-pred.psi.prey %>%
              left_join(sam.period[-4], by=c("PA"="PA", "period"="period.num"))

psipred.mean<-pred.psi.prey %>%
  select(PA, UCI, LCI, mean.psi, period) %>%
  distinct()
```

Prey detection - sampling period 1

```
diel.list<-list(list(day=1,night=0,twilight=0),
                list(day=0,night=1,twilight=0),
                list(day=0,night=0,twilight=1))
z.list<-c(0,1)
diel<-diel.list[[2]]
z.pred<-1

pas<- length(unique(PA))
pa<-1;prd<-1
prediction.all<-data.frame()

for (pa in  1:pas) {

  temp<-lapply(z.list,function(z.pred){
    out2<-  lapply(diel.list,function(diel){
      out <- ilogit(
        logit(myNimbleOutputCoda2$sims.list$p0[,2,pa,prd]) +
          myNimbleOutputCoda2$sims.list$beta.p.night[,2] * diel$night  +
          myNimbleOutputCoda2$sims.list$beta.p.twilight[,2] * diel$twilight +
          myNimbleOutputCoda2$sims.list$beta.p.psi[,2] * z.pred  +
          myNimbleOutputCoda2$sims.list$beta.p.psi.night[,2] * diel$night * z.pred +
          myNimbleOutputCoda2$sims.list$beta.p.psi.twilight[,2] * diel$twilight * z.pred
      )
      out<-tibble(p=out,zpred=z.pred,diel=names(diel)[diel==1])
    })
    do.call(rbind,out2)
  })

  prediction<-do.call(rbind,temp)
  prediction$PA<-c(unique(PA)[pa])
  prediction.all<-rbind(prediction.all, prediction)
}

prey.mean.p<-prediction.all %>% group_by(PA,zpred, diel) %>%
  mutate(UCI=quantile(p, 0.975))%>%
  mutate(LCI=quantile(p, 0.025)) %>%
  mutate(mean.p=mean(p)) %>%
  select(mean.p, UCI,LCI,PA,zpred) %>%
  distinct()
```

```
## Adding missing grouping variables: ‘diel‘
```

Prey detection - all sampling periods

```
diel.list<-list(list(day=1,night=0,twilight=0),
               list(day=0,night=1,twilight=0),
               list(day=0,night=0,twilight=1))
z.list<-c(0,1)
diel<-diel.list[[2]]
z.pred<-1

pas<- length(unique(PA))
pa<-1;prd<-1
prediction.all.y<-data.frame()


for (pa in  1:pas) {

  prds<-sam.period[sam.period$PA.num == pa, ]$period.num

  prd<-1
  for(prd in 1:length(prds)) {

  prd.pa<- prds[prd]

  temp<-lapply(z.list,function(z.pred){
    out2<-  lapply(diel.list,function(diel){
      out <- ilogit(
        logit(myNimbleOutputCoda2$sims.list$p0[,2,pa,prd]) +
          myNimbleOutputCoda2$sims.list$beta.p.night[,2] * diel$night  +
          myNimbleOutputCoda2$sims.list$beta.p.twilight[,2] * diel$twilight +
          myNimbleOutputCoda2$sims.list$beta.p.psi[,2] * z.pred  +
          myNimbleOutputCoda2$sims.list$beta.p.psi.night[,2] * diel$night * z.pred +

          myNimbleOutputCoda2$sims.list$beta.p.psi.twilight[,2] * diel$twilight * z.pred
      )
      out<-tibble(p=out,zpred=z.pred,diel=names(diel)[diel==1])
    })
    do.call(rbind,out2)
  })

  prediction<-do.call(rbind,temp)
  prediction$PA<-c(unique(PA)[pa])
  prediction$period<-prd.pa
  prediction.all.y<-rbind(prediction.all.y, prediction)
  }

}


prediction.all.y.p<-prediction.all.y %>% group_by(PA,zpred, diel, period) %>%
  mutate(UCI=quantile(p, 0.975))%>%
  mutate(LCI=quantile(p, 0.025)) %>%
  mutate(mean.p=mean(p)) %>%
  select(mean.p, UCI,LCI,PA,zpred, diel, period) %>%
  distinct()
```

Predator detection - sampling period 1

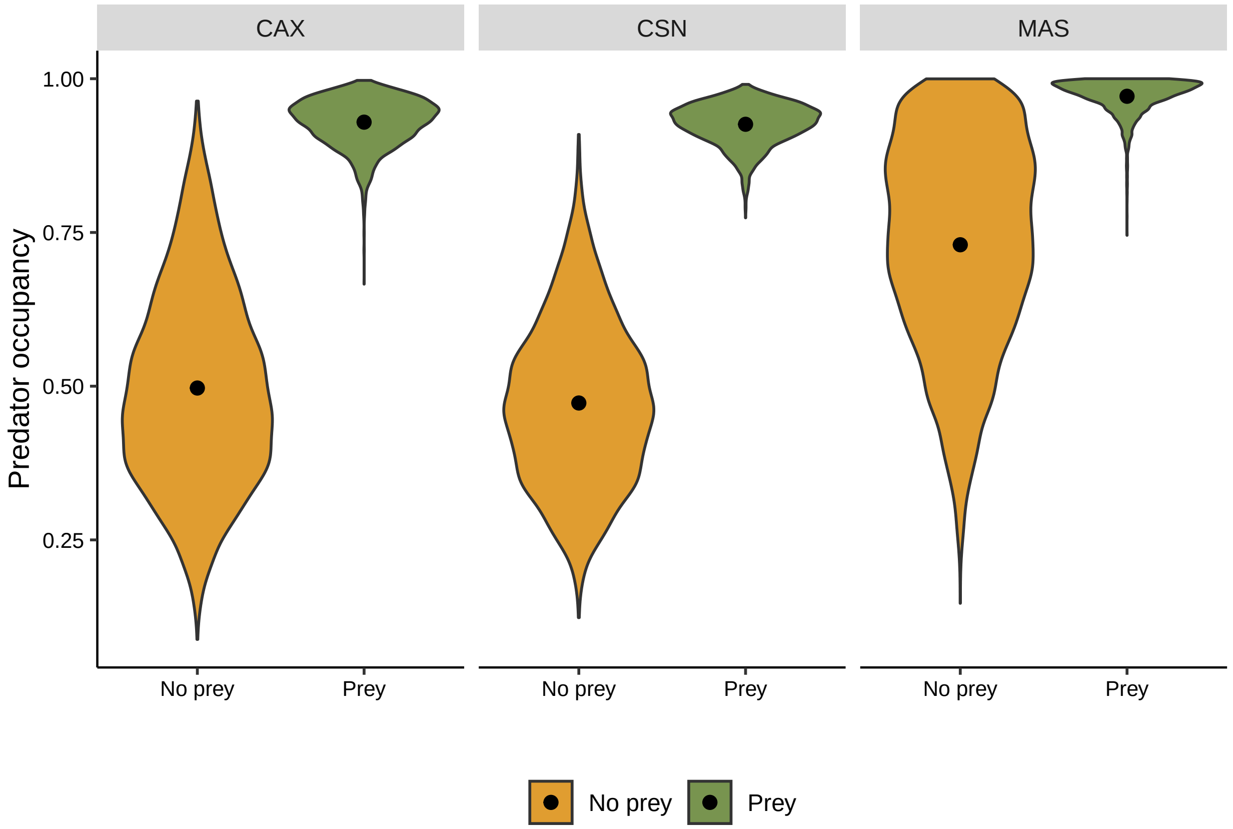
```
diel.list<-list(list(day=1,night=0,twilight=0),
                list(day=0,night=1,twilight=0),
                list(day=0,night=0,twilight=1))
diel<-diel.list[[2]]

pas<- length(unique(PA))
prediction.all.pred<-data.frame()
pa<-1;prd<-1


for (pa in  1:pas) {
  out2<-  lapply(diel.list,function(diel){
    out <- ilogit(
      logit(myNimbleOutputCoda2$sims.list$p0[,1,pa,prd]) +
        myNimbleOutputCoda2$sims.list$beta.p.night[,1] * diel$night  +
        myNimbleOutputCoda2$sims.list$beta.p.twilight[,1] * diel$twilight
    )
    out<-tibble(p=out, diel=names(diel)[diel==1])
  })

  prediction.pred<-do.call(rbind,out2)
  prediction.pred$PA<-c(unique(PA)[pa])
  prediction.all.pred<- rbind(prediction.all.pred, prediction.pred)
}
```

Plot posterior occupancy and detection probabilities

Plot ocelot occupancy, sampling period 1 (Figure 2)

Ocelot occupancy probability across in sampling period 1 in sites were agouti were present vs. absent (Supplementary information, Figure S. 1).

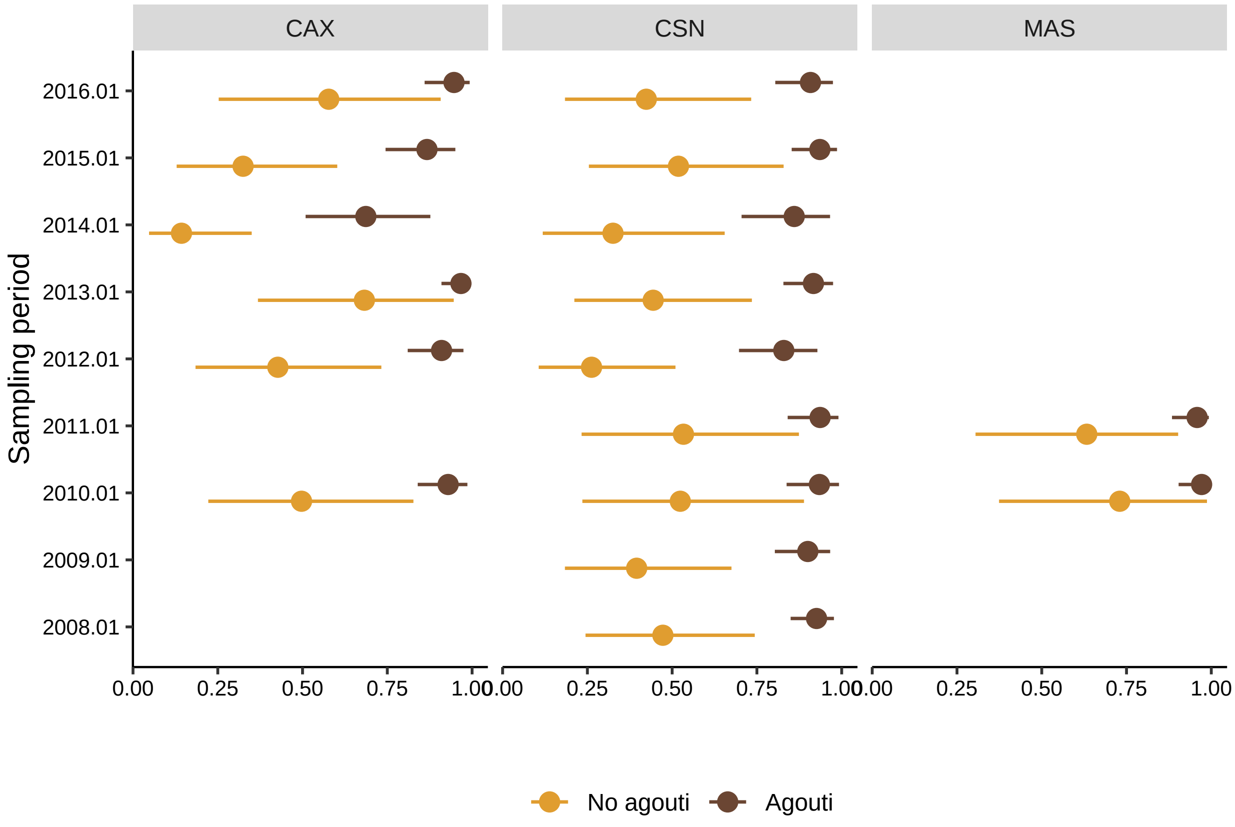
```
# Colors
day<- "#FBDB73"
twilight<-"#D69870"
night<-"#334851"


## Plot psi Ocelot, year 1 (Figure 2)
plot.psi.pred<-pred.psi.pred %>%  filter (period==1) %>%
  ggplot(aes(x=as.factor(zprey),y=psi, fill= as.factor(zprey))) +
  geom_violin( scale= "width")+
  scale_fill_manual(values= c("#e09d30", "#78944f"),
                    labels = c("No prey", "Prey"))+
  scale_x_discrete(labels= c("No prey", "Prey"))+
  theme(panel.background = element_blank(), #strip.background =element_rect(fill="white"),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        #plot.title = element_text(size = 7, hjust = 0.5, vjust = -8),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Predator occupancy")+
  xlab("")+
  facet_wrap(~PA)

plot.pred.psi<-plot.psi.pred+stat_summary(fun=mean, geom="point", size=2, color="black",
                                          position = position_dodge(width = 0.9))

plot.pred.psi
```



Plot ocelot occupancy

Ocelot occupancy probability across all sampling period in sites were agouti were present vs. absent (Figure S. 1).

```
pred.psi.plot<- pred.psi.pred %>% select(PA, mean.psi, LCI,UCI, YearCor,zprey) %>%
                                  distinct()

f.psi<-pred.psi.plot %>%
          ggplot()+
          geom_pointrangeh(aes(x= mean.psi, xmin = LCI, xmax = UCI,
          y=as.factor(YearCor), group= as.factor(zprey), color= as.factor(zprey)),
          position = position_dodge2(width=0.5), size=0.6)+
          scale_color_manual(values= c("#e09d30","#6b4633" ),
                             labels = c("No agouti", "Agouti"))+
        theme(panel.background = element_blank(),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Sampling period")+
  xlab("")+
  facet_grid(~PA)

f.psi
```

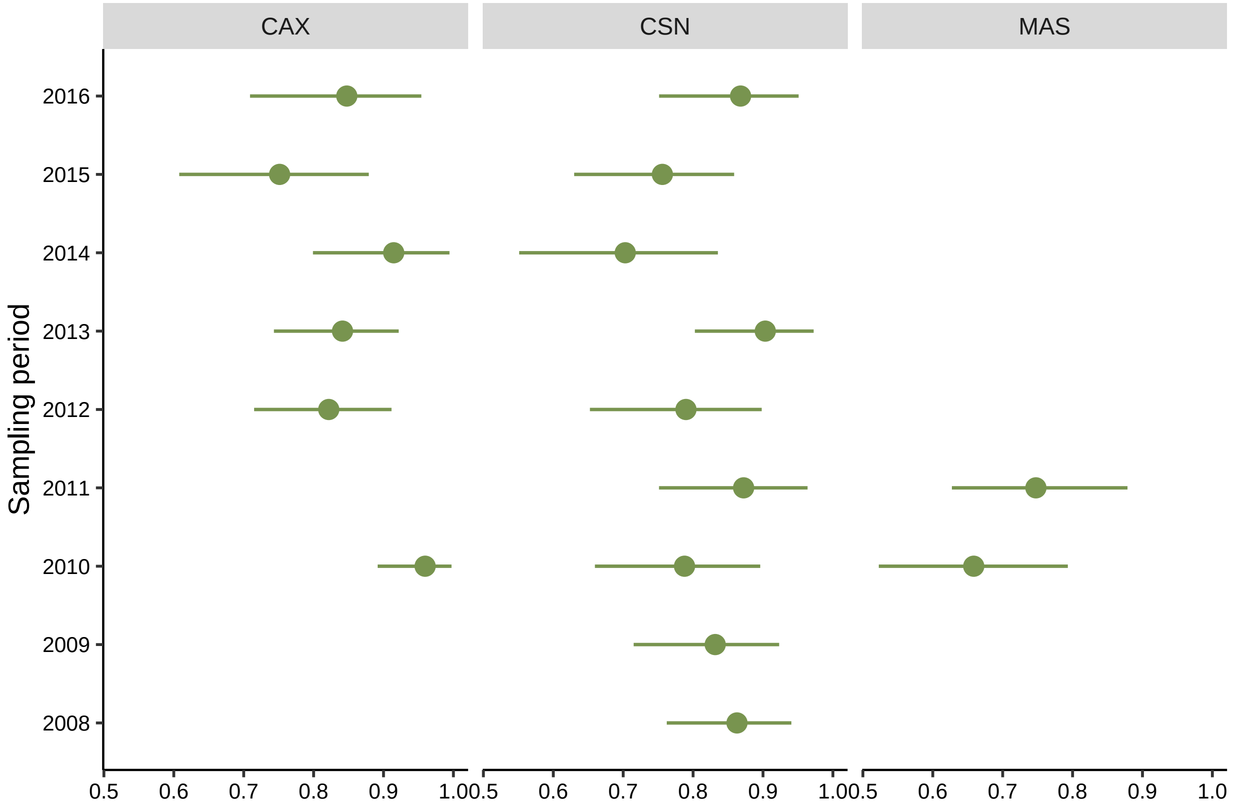


Plot agouti occupancy - all sampling periods

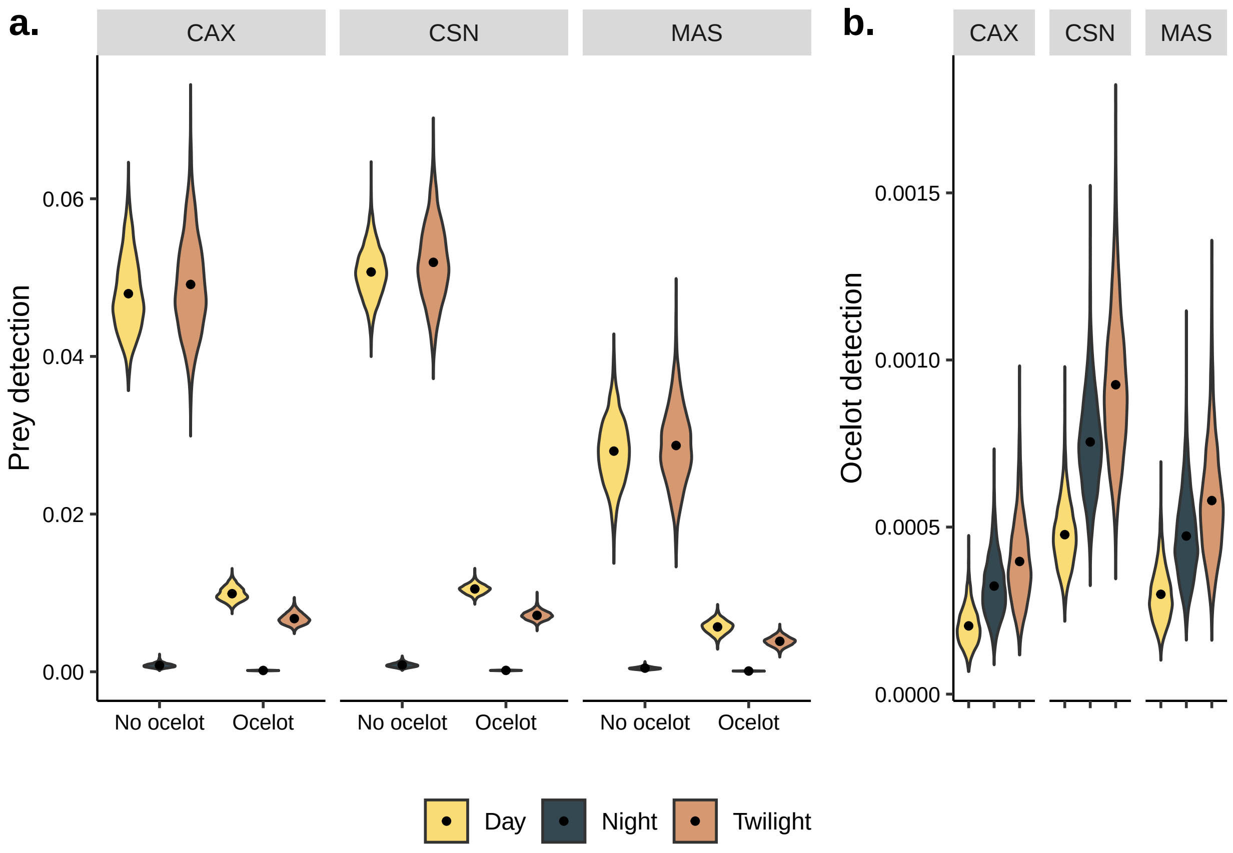
Agouti occupancy probability across all sampling period in sites (Figure S. 2).

```
prey.psi.plot<- pred.psi.prey %>% select(PA, mean.psi, LCI,UCI, YearCor) %>%
                                  distinct()

prey.psi.plot<-prey.psi.plot %>% tidyr :: separate(YearCor, c("Year", "no")) %>%
                                select(!no)

p.estimates.psi<- prey.psi.plot %>%
  ggplot()+
  geom_pointrangeh(aes(x= mean.psi, xmin =UCI, xmax = LCI,
                       y=Year), col= "#78944f",
                   position = position_dodge2(width=0.5), size=0.6)+
  scale_color_manual(values= c( "#78944f"),
                     labels = c("Agouti "))+
  theme(panel.background = element_blank(), #strip.background =element_rect(fill="white"),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        #plot.title = element_text(size = 7, hjust = 0.5, vjust = -8),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Sampling period")+
  xlab("")+
  facet_grid(~PA)
```

```
p.estimates.psi
```



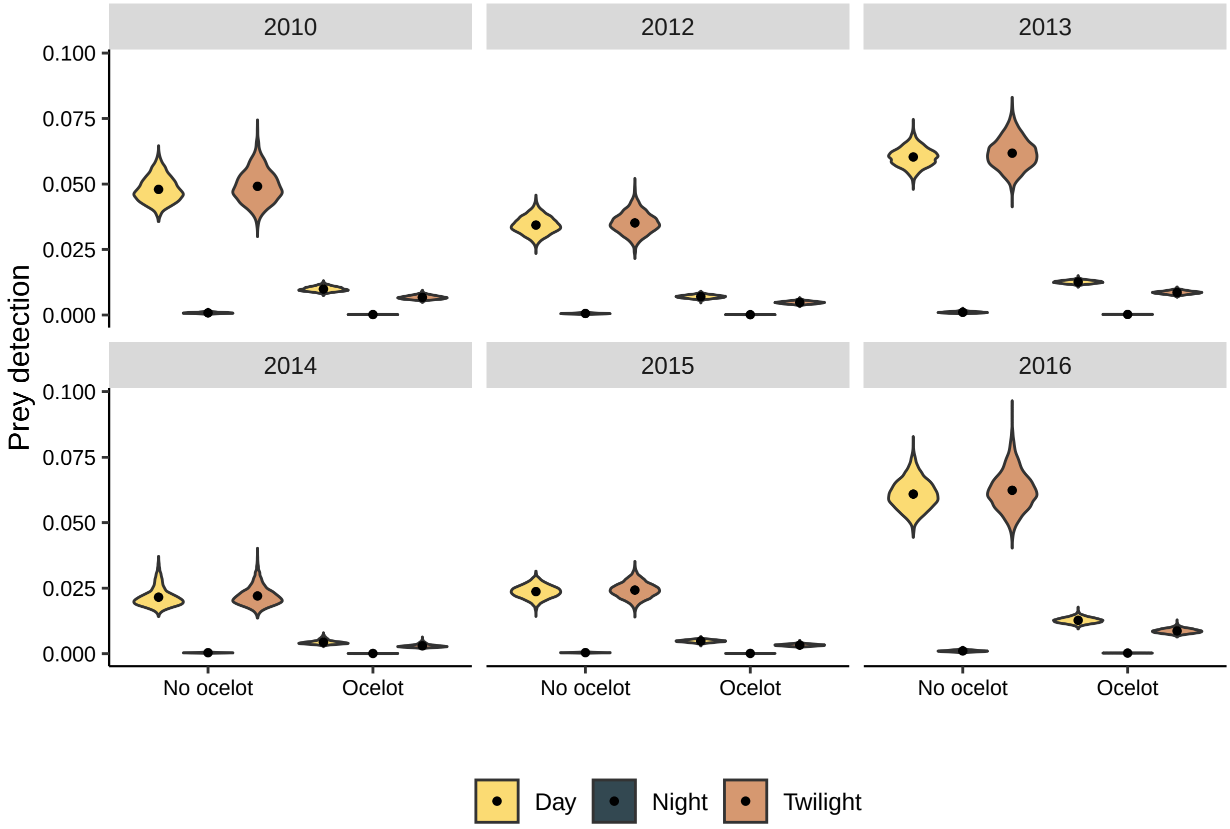
```
##  Plot prey detection: day, night, twiligh & presence/absence of predator one year####
prediction.all$zpred<-as.factor(prediction.all$zpred)
p.vi.prey.p<-ggplot(data=prediction.all, aes(x=zpred,y=p,fill=diel)) +
  geom_violin( scale= "width")+
  scale_fill_manual(values= c(day, night, twilight),
                    labels = c("Day", "Night", "Twilight"))+
  scale_x_discrete(labels= c("No ocelot", "Ocelot"))+
  theme(panel.background = element_blank(), #strip.background =element_rect(fill="white"),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        #plot.title = element_text(size = 7, hjust = 0.5, vjust = -8),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Prey detection")+
  xlab("")+
  facet_wrap(~PA)

plot.prey<-p.vi.prey.p+stat_summary(fun=mean, geom="point", size=1, color="black",
                                    position = position_dodge(width = 0.9))


###  Plot Predator detection ####
p.pred.det<-ggplot(data=prediction.all.pred, aes(x=as.factor(diel),y=p,fill=diel)) +
  geom_violin(scale = "width")+
  scale_fill_manual(values= c(day, night, twilight),
                     labels = c("Day", "Night", "Twilight"))+
  scale_x_discrete(labels= c("", "", ""))+
  theme(panel.background = element_blank(),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Ocelot detection")+
  xlab("")+
  facet_wrap(~PA)


plot.pred<-p.pred.det+stat_summary(fun=mean, geom="point", size=1, color="black",
                                   position = position_dodge(width = 0.9))

##  Detection predictions of both species ####
ggpubr::ggarrange(plot.prey,plot.pred, ncol= 2, widths = c(1, 0.5),
                  common.legend = TRUE, legend="bottom",
                  labels = c("a.","b."))
```



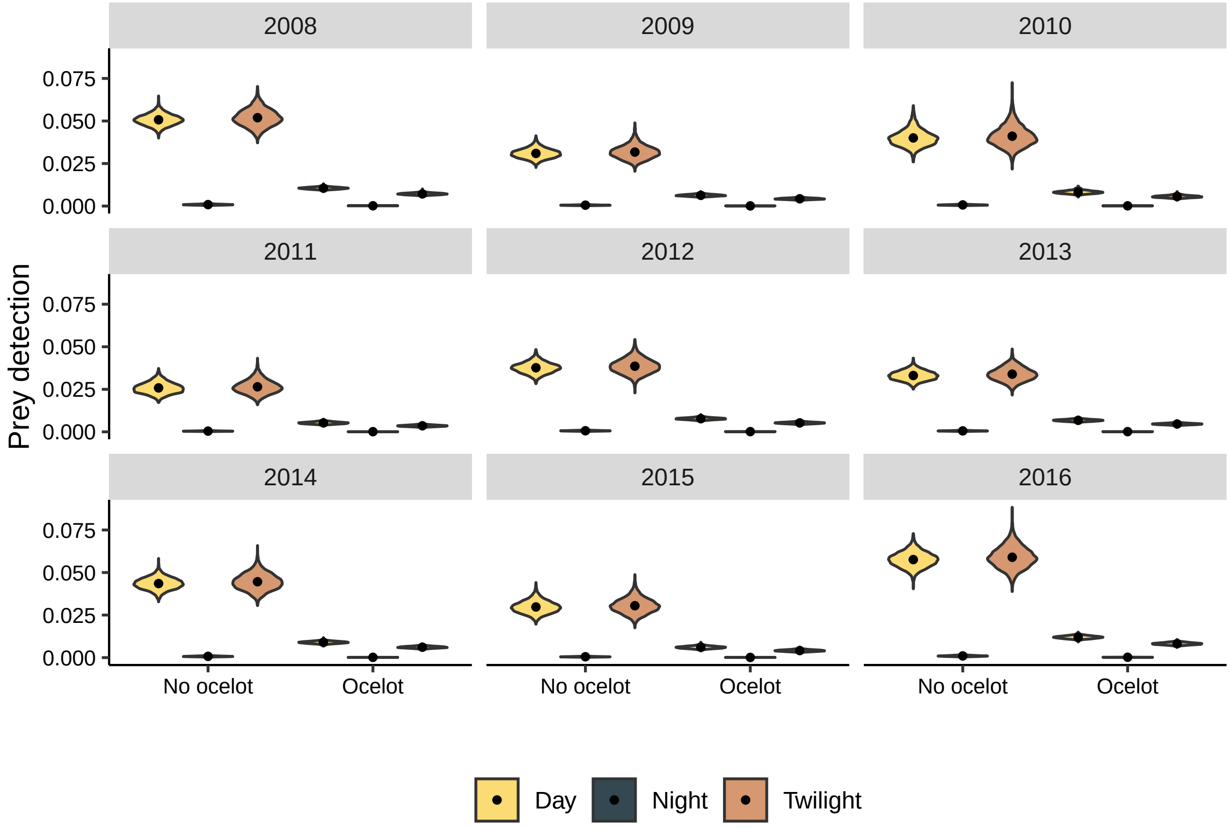
Prey detection all sampling periods and protected areas

Estimates of prey detection by protected area.

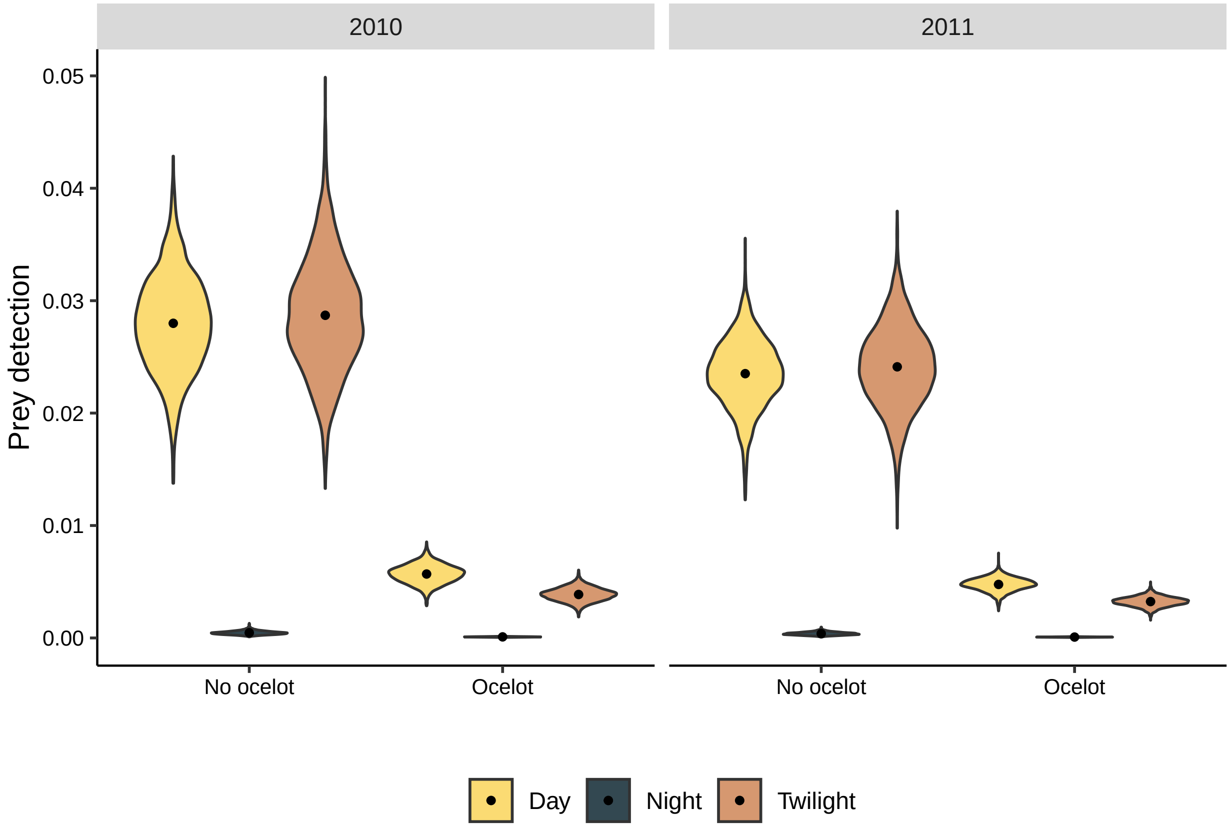
```
prediction.all.y.y<-prediction.all.y %>%
                    left_join(sam.period[-4], by=c("PA"="PA", "period"="period.num")) %>%
                    tidyr::separate(YearCor, c("Year", "no")) %>% select(!no)

## Prey detection in CAX - Figure S. 3
prediction.all.y.y$zpred<-as.factor(prediction.all.y.y$zpred)
p.vi.prey.p.all.y.cax<-prediction.all.y.y %>%  filter (PA== "CAX") %>%
  ggplot(aes(x=zpred,y=p,fill=diel)) +
  geom_violin( scale= "width")+
  scale_fill_manual(values= c(day, night, twilight),
                    labels = c("Day", "Night", "Twilight"))+
  scale_x_discrete(labels= c("No ocelot", "Ocelot"))+
  theme(panel.background = element_blank(), #strip.background =element_rect(fill="white"),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        #plot.title = element_text(size = 7, hjust = 0.5, vjust = -8),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Prey detection")+
  xlab("")+
  facet_wrap(~Year)

p.vi.prey.p.all.y.cax+stat_summary(fun=mean, geom="point", size=1, color="black",
                                   position = position_dodge(width = 0.9))
```



```
## Prey detection in CSN Figure S. 4
p.vi.prey.p.all.y.csn<-prediction.all.y.y %>%  filter (PA== "CSN") %>%
  ggplot(aes(x=zpred,y=p,fill=diel)) +
  geom_violin( scale= "width")+
  scale_fill_manual(values= c(day, night, twilight),
                    labels = c("Day", "Night", "Twilight"))+
  scale_x_discrete(labels= c("No ocelot", "Ocelot"))+
  theme(panel.background = element_blank(),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),
        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Prey detection")+
  xlab("")+
  facet_wrap(~Year)

p.vi.prey.p.all.y.csn+stat_summary(fun=mean, geom="point", size=1, color="black",
                                   position = position_dodge(width = 0.9))
```



```
## Prey detection in MAS Figure S. 5
p.vi.prey.p.all.y.mas<-prediction.all.y.y %>%  filter (PA== "MAS") %>%
  ggplot(aes(x=zpred,y=p,fill=diel)) +
  geom_violin( scale= "width")+
  scale_fill_manual(values= c(day, night, twilight),
                    labels = c("Day", "Night", "Twilight"))+
  scale_x_discrete(labels= c("No ocelot", "Ocelot"))+
  theme(panel.background = element_blank(),
        axis.line = element_line(linewidth =  0.4, colour = "black", linetype=1),
        legend.position="bottom",
        legend.title = element_blank(),

        axis.text =  element_text(size=8, color =  "black" ))+
  ylab("Prey detection")+
  xlab("")+
  facet_wrap(~Year)

p.vi.prey.p.all.y.mas+stat_summary(fun=mean, geom="point", size=1, color="black",
                                   position = position_dodge(width = 0.9))
```



Bayesian posterior predictive check

Simulate data

Simulate data considering the posterior estimates from several iterations. Here the constants for the model are the same as the fitted model. This means that we simulate data considering the same number of sites, number ocassions, protected areas and sampling periods (year)

We only simulated data for 25 iterations to decrease the time for running the code. In the manuscript the simulation corresponded to 100 iterations. We provided the data of the simulations below for comparison.

```
### Define the data to be used in the model
```

```r
## sample 100 iteration from all the iterations of the output model (4000)

# Define your sequence
sequence <- 1:(dim(nimOutputTEAM$chain1)[1])

# Select random numbers from the sequence
random_numbers <- sample(sequence, size = 25)  # Change number of samples as needed


sims_data<-data.frame() #matrix(NA,ncol=7, nrow=100))

i<-1

## loop for 100 samples to simulate data
for(i in 1:length(random_numbers) ){ # length(random_numbers)){

  a<-random_numbers[i]

  nimData <- list(
  beta.p.night = myNimbleOutputCoda2$sims.list$beta.p.night[a,]
  ,
  beta.p.twilight = myNimbleOutputCoda2$sims.list$beta.p.twilight[a,]
  ,
  beta.p.psi  = myNimbleOutputCoda2$sims.list$beta.p.psi[a,]
  ,
  beta.p.psi.night =  myNimbleOutputCoda2$sims.list$beta.p.psi.night[a,]
  ,
  beta.p.psi.twilight = myNimbleOutputCoda2$sims.list$beta.p.psi.twilight[a,]
  ,
  beta.psi.psi = myNimbleOutputCoda2$sims.list$beta.psi.psi[a,]
  ,
  psi0 = myNimbleOutputCoda2$sims.list$psi0[a,,,]
  ,
  p0 = myNimbleOutputCoda2$sims.list$p0[a,,,]

)

true.parms <- nimData

# Initialize the stochastic nodes of the model
nimInits <- list()


### Run the version of the model  ====
ptm <- proc.time()
simModel <- nimbleModel( code = modelCodeTEAM,
                         constants = nimConstantsTEAM,
                         data = nimData,
                         check = F)



## Identify nodes and simulate
```

```
nodesToSim <- simModel$getDependencies(names(nimData),
                                       self = F, downstream = T,
                                       returnScalarComponents = TRUE)

# Simulate:
simModel$simulate(nodesToSim,includeData = FALSE)

mod.name<- paste0("simModel", i)
assign(mod.name, simModel)

value <- get(mod.name)

temp<- as.data.frame(apply( value$y[,,], c(2,3), sum))
temp$sp <- c("Ocelot", "Agouti")
colnames(temp)<- c("day", "night", "twilight", "Specie")

temp$simulation<- i
sims_data <- rbind(sims_data, temp)

}
```

Load simulated data

Data were simulated using 100 simulations using posteriors of model-estimated parameters.

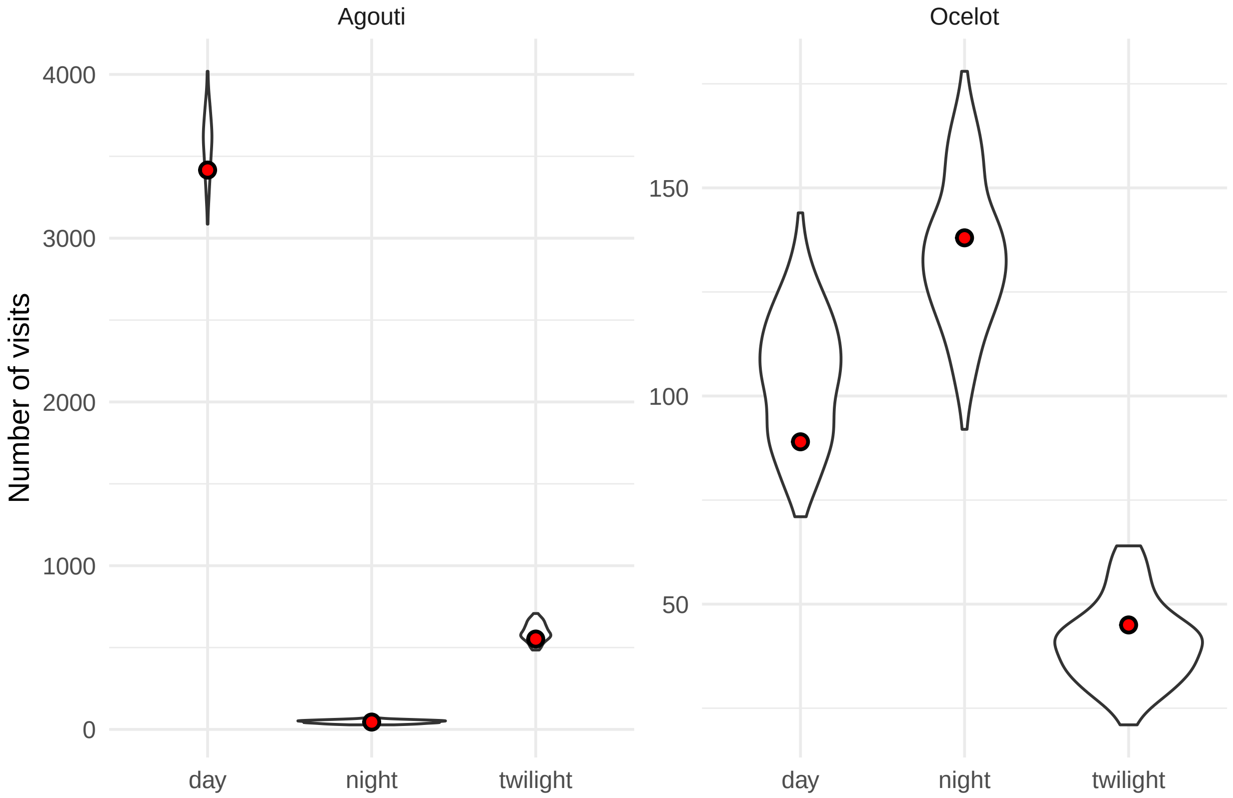
```
load("100_it_SimsData.Rdata")

sims_data2<- melt(sims_data , id.vars= c("Specie", "simulation"))

# Summarise the number of real observations by species, diel categories to
# compare with the simulations

Tdata<-apply( nimDataTEAM$y[,,], c(2,3), sum)
Tdata<-as.data.frame(Tdata)
Tdata<-Tdata %>% mutate(Specie = c("Ocelot", "Agouti"))
Tdata<-melt(Tdata, id.vars= "Specie")
```

Plot simulated and real data

In this plot we used the 100 simulated data sets to plot the distribution with violin plots, and in red we plotted the real observation of ocelots and agoutis in our data set.

```
# Figure S. 7.
PlotsimViolin<-sims_data2 %>%
ggplot()+geom_violin(aes(y=value, x=variable, group=variable)) +
  geom_point(data=Tdata, aes(x=variable, y=value), size=2,color = "black",
             fill = "red", stroke = 1, shape= 21)+
  facet_wrap(~Specie, scales= "free")+
  xlab("")+
  ylab("Number of visits")+
  theme_minimal()
```

```
PlotsimViolin
```



References

Gelman, A. (1996). Inference and monitoring convergence. *Markov Chain Monte Carlo in Practice*, 131–144.

Valpine, P. de, Turek, D., Paciorek, C. J., Anderson-Bergman, C., Lang, D. T., & Bodik, R. (2017). Programming with models: Writing statistical algorithms for general model structures with NIMBLE. *Journal of Computational and Graphical Statistics*, *26*(2), 403–413.