**Supplementary material**

**Predator and scavenger movements among and within endangered seabird colonies: opportunities for pathogen spread**

Amandine Gamble, Romain Bazire, Karine Delord, Christophe Barbraud, Audrey Jaeger, Hubert Gantelet, Eric Thibault, Camille Lebarbenchon, Erwan Lagadec, Pablo Tortosa, Henri Weimerskirch, Jean-Baptiste Thiebot, Romain Garnier, Jérémy Tornos, and Thierry Boulinier

**Appendix S3. R notebook**

Movements of an apex consumer among and within endangered seabird colonies: opportunities for pathogen spread

Gamble et al.

2019

R code for figures and analyses

Written by Amandine Gamble ([amandine.gamble@gmail.com](mailto:amandine.gamble@gmail.com))

Document created with R Markdown in RStudio 1.2.1335 and R 3.6.1

# Working space

# needed files:  
# immuno\_functions.r  
# data/sku\_epidemio.csv  
# data/sku\_gps.csv  
# data/ams\_species\_altitude.csv  
# data/yna\_nestlings.csv  
  
# data  
sku\_epidemio=read.csv("data/sku\_epidemio.csv",sep=";",header=T)  
sku\_gps=read.table("data/sku\_gps.csv",header=T,sep=";",dec=",")  
species\_altitude=read.table("data/ams\_species\_altitude.csv",header=T,sep=";",dec=",")  
albatross\_nestlings=read.csv("data/yna\_nestlings.csv",sep=";",dec=".")  
# if nest seen empty once during the month,recorded as empty  
# if individual pcr-positive once during the month,recorded as positive  
  
# functions  
source("R/immuno\_functions.r")  
  
# packages  
  
library(ggplot2);library(gridExtra);library(cowplot)  
  
ggtheme=theme\_bw() +  
 theme(panel.grid=element\_blank(),  
 panel.border=element\_blank(),  
 axis.title=element\_text(size=8),  
 axis.text=element\_text(size=8,color="black"),  
 axis.line=element\_line(size=0.25,color="black"),  
 axis.ticks=element\_line(size=0.25,color="black"),  
 legend.text=element\_text(size=8),  
 plot.title=element\_text(size=8),  
 legend.title=element\_blank(),  
 legend.margin=margin(0,0,0,0),  
 legend.key=element\_rect(fill=NA,color=NA),  
 legend.key.size=unit(0.4,"cm"),  
 plot.margin=margin(5,5,5,5),  
 legend.box="horizontal",  
 legend.background=element\_rect(colour="transparent",fill=ggplot2::alpha("white",0)))  
  
library(mixtools);library(MASS)  
library(lmodel2)  
library(fmsb);library(psych)  
library(car)  
library(sp);library(geosphere);library(rgdal);library(adehabitatHR)  
library(igraph)  
library(move);library(moveVis)

# 1. Eco-epidemiological data

if(grepl("/",as.character(sku\_epidemio$date[1]))){  
 sku\_epidemio$date=as.Date(sku\_epidemio$date,format="%d/%m/%Y")  
}else{sku\_epidemio$date=as.Date(sku\_epidemio$date)}  
  
# log transformation  
sku\_epidemio$titer\_log=log2(sku\_epidemio$titer/10)  
sku\_epidemio[sku\_epidemio$titer\_log<0 & !is.na(sku\_epidemio$titer\_log),"titer\_log"]=-1  
sku\_epidemio$titer\_log=sku\_epidemio$titer\_log+1

Counts

# number of samples from other sites  
sum(sku\_epidemio$project=="collaboration\_JGS")

## [1] 25

# number of samples analysed with MAT  
sum(!is.na(sku\_epidemio$titer))

## [1] 100

# number of samples analysed with ELISA   
sum(!is.na(sku\_epidemio$OD\_cor))

## [1] 98

# number of samples analysed with both assays  
sum(!is.na(sku\_epidemio$OD\_cor) & !is.na(sku\_epidemio$titer))

## [1] 98

# number of samples analysed with MAT only  
sum(is.na(sku\_epidemio$OD\_cor) & !is.na(sku\_epidemio$titer))

## [1] 2

# number of samples analysed with ELISA only  
sum(!is.na(sku\_epidemio$OD\_cor) & is.na(sku\_epidemio$titer))

## [1] 0

# number of samples analysed with ELISA on Amsterdam  
sum(!is.na(sku\_epidemio$OD\_cor) & sku\_epidemio$project=="ecopath")

## [1] 73

# number of samples analysed with MAT on Amsterdam  
sum(!is.na(sku\_epidemio$titer)& sku\_epidemio$project=="ecopath")

## [1] 75

# number of samples analysed with MAT only on Amsterdam  
sum(is.na(sku\_epidemio$OD\_cor) & !is.na(sku\_epidemio$titer)& sku\_epidemio$project=="ecopath")

## [1] 2

# number of samples analysed with ELISA only on Amsterdam  
sum(!is.na(sku\_epidemio$OD\_cor) & is.na(sku\_epidemio$titer)& sku\_epidemio$project=="ecopath")

## [1] 0

# ggplot()+geom\_point(data=subset(sku\_epidemio,stage=="adult"),aes(x=tarsus,y=skull,color=as.factor(detect\_pos)))

## 1.1. ELISA cut-off

Method described in Garnier et al., 2017, Functional Ecology

# fits two normal distributions to the corrected OD values  
mixnorm<-normalmixEM(subset(sku\_epidemio,!is.na(OD\_cor))$OD\_cor,eps=1e-12,lambda=0.5,mu=c(0,1),sigma=c(0.1,1),k=2)

## number of iterations= 19

# fits one normal distribution to the same data as above  
norm1<-fitdistr(subset(sku\_epidemio,!is.na(OD\_cor))$OD\_cor,"normal")  
  
# manually calculate AIC   
AIC1n<-2\*2-2\*norm1$loglik  
AIC2n<-4\*2-2\*mixnorm$loglik  
c(AIC1n,AIC2n)

## [1] 147.53198 61.86505

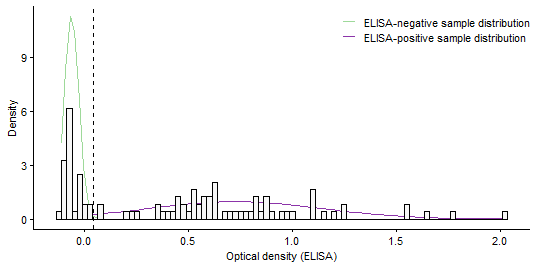
if (AIC2n<AIC1n-2){print("Two normal distributions")}else{print("One unique normal distribution")}

## [1] "Two normal distributions"

cut95=mixnorm$mu[1]+2\*mixnorm$sigma[1]  
cut99=mixnorm$mu[1]+3\*mixnorm$sigma[1]  
c(cut95,cut99)

## [1] 0.005364065 0.040241703

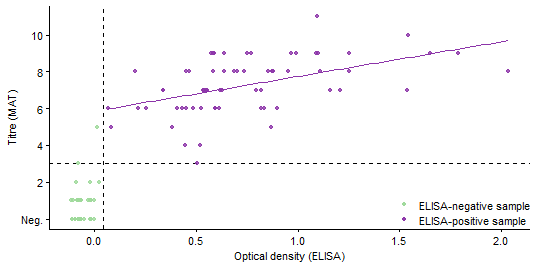
ggplot() +   
 stat\_function(aes(x=subset(sku\_epidemio,!is.na(OD\_cor))$OD\_cor,colour="#8c34a9"),fun=dnorm,args=list(mean=mixnorm$mu[1],sd=mixnorm$sigma[1]),size=0.5)+  
 stat\_function(aes(x=subset(sku\_epidemio,!is.na(OD\_cor))$OD\_cor,colour="#9ed99a"),fun=dnorm,args=list(mean=mixnorm$mu[2],sd=mixnorm$sigma[2]),size=0.5)+  
 geom\_histogram(aes(x=subset(sku\_epidemio,!is.na(OD\_cor))$OD\_cor,y=..density..),  
 binwidth=.025,size=0.25,  
 colour="black",fill="gray97")+  
 geom\_vline(aes(xintercept=cut99),linetype="dashed",size=0.5)+  
 ylab("Density")+xlab("Optical density (ELISA)")+  
 ggtheme+  
 scale\_color\_manual(values=c("#9ed99a","#8c34a9"),labels=c("ELISA-negative sample distribution","ELISA-positive sample distribution"))+  
 theme(legend.justification=c(1,1),legend.position=c(1,1))



ggsave("r/exports/sku\_epidemio\_OD\_distribution.pdf",width=14,height=7,units="cm",dpi=300)  
  
sku\_epidemio$titer\_pos=sku\_epidemio$sero\_pos=NA  
sku\_epidemio[!is.na(sku\_epidemio$OD\_cor) & sku\_epidemio$OD\_cor>cut99,"sero\_pos"]="1"  
sku\_epidemio[!is.na(sku\_epidemio$OD\_cor) & sku\_epidemio$OD\_cor<=cut99,"sero\_pos"]="0"  
sku\_epidemio[!is.na(sku\_epidemio$titer) & sku\_epidemio$titer>40,"titer\_pos"]="1"  
sku\_epidemio[!is.na(sku\_epidemio$titer) & sku\_epidemio$titer<=40,"titer\_pos"]="0"  
  
remove(mixnorm,AIC2n,norm1,AIC1n)

## 1.2. Correlation between immunoassays

# graphic  
ggplot() +   
 geom\_vline(aes(xintercept=cut99),linetype="dashed",size=0.5)+  
 geom\_hline(aes(yintercept=3),linetype="dashed",size=0.5)+  
 geom\_smooth(data=subset(subset(sku\_epidemio,!is.na(OD\_cor)),OD\_cor>cut99),aes(y=titer\_log,x=OD\_cor),se=F,size=0.5,colour="#8c34a9",method="lm",fill="#af8dc3",alpha=0.1)+  
 geom\_point(data=subset(subset(sku\_epidemio,!is.na(OD\_cor)),OD\_cor>cut99),aes(y=titer\_log,x=OD\_cor,colour="#9ed99a"),size=1,stroke=0.5,alpha=0.8)+  
 geom\_point(data=subset(subset(sku\_epidemio,!is.na(OD\_cor)),OD\_cor<=cut99),aes(y=titer\_log,x=OD\_cor,colour="#8c34a9"),size=1,stroke=0.5,alpha=0.8)+  
 xlab("Optical density (ELISA)")+ylab("Titre (MAT)")+  
 scale\_x\_continuous(breaks=seq(0,2,0.5))+  
 scale\_y\_continuous(breaks=seq(0,12,2),labels=c("Neg.","2","4","6","8","10","12"))+  
 ggtheme+  
 scale\_color\_manual(values=c("#9ed99a","#8c34a9"),labels=c("ELISA-negative sample","ELISA-positive sample"))+  
 theme(legend.justification=c(1,0.01),legend.position=c(1,0.01))



ggsave("r/exports/sku\_epidemio\_immuno\_correlation.pdf",width=14,height=7,units="cm",dpi=300)

### 1.2.1. Linear regression

On quantitative data

# sample size - positive samples only  
nrow(subset(subset(sku\_epidemio,!is.na(OD\_cor)),OD\_cor>cut99))

## [1] 63

# classic linear regression  
summary(lm(OD\_cor~titer\_log,subset(subset(sku\_epidemio,!is.na(OD\_cor)),OD\_cor>cut99)))

##   
## Call:  
## lm(formula = OD\_cor ~ titer\_log, data = subset(subset(sku\_epidemio,   
## !is.na(OD\_cor)), OD\_cor > cut99))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.64532 -0.20290 -0.07924 0.15640 1.19124   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.16924 0.21535 -0.786 0.435   
## titer\_log 0.12652 0.02893 4.373 4.86e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3506 on 61 degrees of freedom  
## Multiple R-squared: 0.2387, Adjusted R-squared: 0.2262   
## F-statistic: 19.13 on 1 and 61 DF, p-value: 4.855e-05

# type II linear regression  
lmodel2(OD\_cor~titer\_log,data=subset(subset(sku\_epidemio,!is.na(OD\_cor)),OD\_cor>cut99),nperm=99)

## RMA was not requested: it will not be computed.

##   
## Model II regression  
##   
## Call: lmodel2(formula = OD\_cor ~ titer\_log, data =  
## subset(subset(sku\_epidemio, !is.na(OD\_cor)), OD\_cor > cut99),  
## nperm = 99)  
##   
## n = 63 r = 0.4885705 r-square = 0.2387011   
## Parametric P-values: 2-tailed = 4.855318e-05 1-tailed = 2.427659e-05   
## Angle between the two OLS regression lines = 20.71455 degrees  
##   
## Permutation tests of OLS, MA, RMA slopes: 1-tailed, tail corresponding to sign  
## A permutation test of r is equivalent to a permutation test of the OLS slope  
## P-perm for SMA = NA because the SMA slope cannot be tested  
##   
## Regression results  
## Method Intercept Slope Angle (degrees) P-perm (1-tailed)  
## 1 OLS -0.1692401 0.1265210 7.210807 0.01  
## 2 MA -0.2179676 0.1332091 7.587649 0.01  
## 3 SMA -1.1341646 0.2589616 14.518475 NA  
##   
## Confidence intervals  
## Method 2.5%-Intercept 97.5%-Intercept 2.5%-Slope 97.5%-Slope  
## 1 OLS -0.5998669 0.2613866 0.06867194 0.1843701  
## 2 MA -0.6660945 0.2230584 0.07267611 0.1947167  
## 3 SMA -1.6021392 -0.7591957 0.20749531 0.3231934  
##   
## Eigenvalues: 2.408584 0.1189244   
##   
## H statistic used for computing C.I. of MA: 0.003581438

### 1.2.2. Cohen’s Kappa

On qualitative data

# sample size - samples analysed with both assays  
nrow(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor)))

## [1] 98

## MAT and ELISA concordance table  
pp=sum(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos==1 & subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos==1,na.rm=T)  
nn=sum(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos==0 & subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos==0,na.rm=T)  
pn=sum(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos==1 & subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos==0,na.rm=T)  
np=sum(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos==0 & subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos==1,na.rm=T)  
  
concordance=as.matrix(rbind(c(nn,np),c(pn,pp)));concordance

## [,1] [,2]  
## [1,] 34 1  
## [2,] 1 62

## Kappa coefficient  
cohen.kappa(cbind(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos,subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos))

## Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha, levels = levels)  
##   
## Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries   
## lower estimate upper  
## unweighted kappa 0.89 0.96 1  
## weighted kappa 0.89 0.96 1  
##   
## Number of subjects = 98

Kappa.test(subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos,subset(sku\_epidemio,!is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos)

## $Result  
##   
## Estimate Cohen's kappa statistics and test the null hypothesis  
## that the extent of agreement is same as random (kappa=0)  
##   
## data: subset(sku\_epidemio, !is.na(titer\_log) & !is.na(OD\_cor))$titer\_pos and subset(sku\_epidemio, !is.na(titer\_log) & !is.na(OD\_cor))$sero\_pos  
## Z = 8.7164, p-value < 2.2e-16  
## 95 percent confidence interval:  
## 0.8945916 1.0165195  
## sample estimates:  
## [1] 0.9555556  
##   
##   
## $Judgement  
## [1] "Almost perfect agreement"

remove(pp,nn,pn,np,concordance)

## 1.3. Prevalences

# seroprevalence in adults according to ELISA  
subset(get\_prev(subset(sku\_epidemio,stage=="adult"),"sero\_pos",c("zone","campaign")),n>0)

##   
## Attaching package: 'plyr'

## The following object is masked from 'package:adehabitatLT':  
##   
## id

## The following object is masked from 'package:adehabitatMA':  
##   
## join

## zone campaign n pos prev CImin CImax  
## 2 carcass 2012/2013 17 0 0.00 0.00 0.20  
## 7 ETX\_club 2015/2016 10 10 1.00 0.69 1.00  
## 8 ETX\_club 2016/2017 10 10 1.00 0.69 1.00  
## 12 MAE 2016/2017 8 6 0.75 0.35 0.97  
## 13 PDT 2011/2012 16 16 1.00 0.79 1.00  
## 15 PDT 2015/2016 10 10 1.00 0.69 1.00  
## 16 PDT 2016/2017 10 10 1.00 0.69 1.00  
## 18 speedwell 2012/2013 8 0 0.00 0.00 0.37

subset(get\_prev(subset(sku\_epidemio,stage=="adult" & campaign!="2012/2013"),"sero\_pos",c("campaign")),n>0)

## campaign n pos prev CImin CImax  
## 1 2011/2012 16 16 1.00 0.79 1.00  
## 2 2015/2016 20 20 1.00 0.83 1.00  
## 3 2016/2017 28 26 0.93 0.76 0.99

# seroprevalence in adults according to MAT  
subset(get\_prev(subset(sku\_epidemio,stage=="adult"),"titer\_pos",c("zone","campaign")),n>0)

## zone campaign n pos prev CImin CImax  
## 2 carcass 2012/2013 17 0 0.0 0.00 0.20  
## 7 ETX\_club 2015/2016 10 10 1.0 0.69 1.00  
## 8 ETX\_club 2016/2017 10 10 1.0 0.69 1.00  
## 12 MAE 2016/2017 10 8 0.8 0.44 0.97  
## 13 PDT 2011/2012 16 16 1.0 0.79 1.00  
## 15 PDT 2015/2016 10 10 1.0 0.69 1.00  
## 16 PDT 2016/2017 10 10 1.0 0.69 1.00  
## 18 speedwell 2012/2013 8 0 0.0 0.00 0.37

subset(get\_prev(subset(sku\_epidemio,stage=="adult" & campaign!="2012/2013"),"titer\_pos",c("campaign")),n>0)

## campaign n pos prev CImin CImax  
## 1 2011/2012 16 16 1.00 0.79 1.00  
## 2 2015/2016 20 20 1.00 0.83 1.00  
## 3 2016/2017 30 28 0.93 0.78 0.99

# prevalence in adults according to PCR in clocal swabs  
subset(get\_prev(subset(sku\_epidemio,stage=="adult"),"detect\_pos",c("zone","campaign")),n>0)

## zone campaign n pos prev CImin CImax  
## 7 ETX\_club 2015/2016 10 3 0.30 0.07 0.65  
## 8 ETX\_club 2016/2017 10 2 0.20 0.03 0.56  
## 12 MAE 2016/2017 10 0 0.00 0.00 0.31  
## 13 PDT 2011/2012 16 1 0.06 0.00 0.30  
## 15 PDT 2015/2016 10 1 0.10 0.00 0.45  
## 16 PDT 2016/2017 10 8 0.80 0.44 0.97

# seroprevalence in chicks according to ELISA  
subset(get\_prev(subset(sku\_epidemio,stage=="chick"),"sero\_pos",c("zone","campaign")),n>0)

## zone campaign n pos prev CImin CImax  
## 1 PDT 2011/2012 3 0 0.00 0 0.71  
## 2 PDT 2016/2017 6 1 0.17 0 0.64

# seroprevalence in chicks according to MAT  
subset(get\_prev(subset(sku\_epidemio,stage=="chick"),"titer\_pos",c("zone","campaign")),n>0)

## zone campaign n pos prev CImin CImax  
## 1 PDT 2011/2012 3 0 0.00 0 0.71  
## 2 PDT 2016/2017 6 1 0.17 0 0.64

# prevalence in chicks according to PCR in clocal swabs  
subset(get\_prev(subset(sku\_epidemio,stage=="chick"),"detect\_pos",c("zone","campaign")),n>0)

## zone campaign n pos prev CImin CImax  
## 1 PDT 2011/2012 3 0 0.0 0.00 0.71  
## 2 PDT 2016/2017 6 3 0.5 0.12 0.88

## 1.4. Adults

sku\_epidemio\_ams=subset(sku\_epidemio,project=="ecopath")  
  
# mean titer (MAT)  
mean(subset(sku\_epidemio\_ams,stage=="adult")$titer\_log,na.rm=T)

## [1] 7.151515

sd(subset(sku\_epidemio\_ams,stage=="adult")$titer\_log,na.rm=T)

## [1] 1.693812

sum(!is.na(subset(sku\_epidemio\_ams,stage=="adult")$titer\_log))

## [1] 66

# mean OD (ELISA)  
mean(subset(sku\_epidemio\_ams,stage=="adult")$OD\_cor,na.rm=T)

## [1] 0.732968

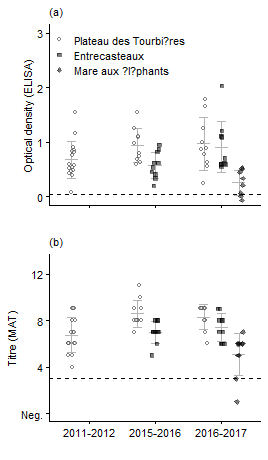
sd(subset(sku\_epidemio\_ams,stage=="adult")$OD\_cor,na.rm=T)

## [1] 0.4168822

sum(!is.na(subset(sku\_epidemio\_ams,stage=="adult")$OD\_cor))

## [1] 64

# effect of season and site on OD  
sku\_epidemio\_ams$zone=as.factor(as.character(sku\_epidemio\_ams$zone))  
sku\_epidemio\_ams$zone=factor(sku\_epidemio\_ams$zone,levels(sku\_epidemio\_ams$zone)[c(3,1,2)])  
  
dummies=sku\_epidemio\_ams[1:3,]  
dummies$campaign=c("2011/2012","2011/2012","2015/2016")  
dummies$zone=c("ETX\_club","MAE","MAE")  
dummies$OD\_cor=100  
dummies$titer\_log=10000  
  
# ELISA  
sku\_stat\_elisa=get\_stat(subset(sku\_epidemio\_ams,stage=="adult"),"OD\_cor",c("campaign","zone"))  
sku\_stat\_elisa[sku\_stat\_elisa$n==0,"mean"]=99  
sku\_stat\_elisa[sku\_stat\_elisa$n==0,"sd"]=0  
sku\_stat\_elisa$zone=factor(sku\_stat\_elisa$zone,levels(sku\_stat\_elisa$zone)[c(3,1,2)])  
  
d=0.8  
p1=ggplot() +   
 geom\_hline(aes(yintercept=cut99),linetype="dashed",size=0.5)+  
 geom\_errorbar(data=sku\_stat\_elisa,aes(x=campaign,ymin=mean-sd,ymax=mean+sd,group=zone),  
 position=position\_dodge(d),colour="gray70",width=0.4,size=0.5)+  
 geom\_errorbar(data=sku\_stat\_elisa,aes(x=campaign,ymin=mean,ymax=mean,group=zone),  
 position=position\_dodge(d),colour="gray70",width=0.6,size=0.5)+  
 geom\_point(data=rbind(subset(sku\_epidemio\_ams,stage=="adult"),dummies),aes(x=campaign,y=OD\_cor,shape=zone,fill=zone),  
 position=position\_jitterdodge(dodge.width=d,jitter.width=0.3),colour="black",size=1,stroke=0.5,alpha=0.5)+  
 ylab("Optical density (ELISA) ")+  
 xlab("")+  
 scale\_x\_discrete(labels=(c("2011-2012","2015-2016","2016-2017")))+  
 scale\_shape\_manual(labels=c("Plateau des Tourbi?res","Entrecasteaux","Mare aux ?l?phants"),values=c(21,22,23))+  
 scale\_fill\_manual(labels=c("Plateau des Tourbi?res","Entrecasteaux","Mare aux ?l?phants"),values=c("white","black","black"))+  
 ggtheme+  
 theme(legend.justification=c(0,1),legend.position=c(0.01,1),  
 plot.margin=margin(5,5,5,17),  
 axis.text.x=element\_blank())+  
 coord\_cartesian(ylim=c(0,3))+  
 ggtitle("(a)")  
  
# MAT  
sku\_stat\_mat=get\_stat(subset(sku\_epidemio\_ams,stage=="adult"),"titer\_log",c("campaign","zone"))  
sku\_stat\_mat[sku\_stat\_mat$n==0,"mean"]=99  
sku\_stat\_mat[sku\_stat\_mat$n==0,"sd"]=0  
sku\_stat\_mat$zone=factor(sku\_stat\_mat$zone,levels(sku\_stat\_mat$zone)[c(3,1,2)])  
  
d=0.8  
p2=ggplot() +   
 geom\_hline(aes(yintercept=3),linetype="dashed",size=0.5)+  
 geom\_errorbar(data=sku\_stat\_mat,aes(x=campaign,ymin=mean-sd,ymax=mean+sd,group=zone),  
 position=position\_dodge(d),colour="gray70",width=0.4,size=0.5)+  
 geom\_errorbar(data=sku\_stat\_mat,aes(x=campaign,ymin=mean,ymax=mean,group=zone),  
 position=position\_dodge(d),colour="gray70",width=0.6,size=0.5)+  
 geom\_point(data=rbind(subset(sku\_epidemio\_ams,stage=="adult"),dummies),aes(x=campaign,y=titer\_log,shape=zone,fill=zone),  
 position=position\_jitterdodge(dodge.width=d,jitter.width=0.3),colour="black",size=1,stroke=0.5,alpha=0.5)+  
 ylab("Titre (MAT)")+  
 xlab("")+  
 scale\_x\_discrete(labels=(c("2011-2012","2015-2016","2016-2017")))+  
 scale\_shape\_manual(labels=c("Plateau des Tourbi?res","Entrecasteaux","Mare aux ?l?phants"),values=c(21,22,23))+  
 scale\_fill\_manual(labels=c("Plateau des Tourbi?res","Entrecasteaux","Mare aux ?l?phants"),values=c("white","black","black"))+  
 scale\_y\_continuous(breaks=seq(0,12,4),labels=c("Neg.","4","8","12"))+  
 ggtheme+  
 coord\_cartesian(ylim=c(0,13))+  
 theme(legend.position="none")+  
 ggtitle("(b)")  
  
p=grid.arrange(p1,p2,nrow=2)



ggsave("r/exports/sku\_epidemio\_antibody\_levels.pdf",p,width=7,height=12,units="cm",dpi=300)  
  
remove(p1,p2,p)  
remove(d,dummies)  
  
# Wilcoxon with Bonferroni correction  
  
ntests=8  
  
# among sites  
w=wilcox.test(OD\_cor~zone,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="MAE" | zone=="ETX\_club") & campaign=="2016/2017"))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="MAE" & campaign=="2016/2017"))

## [1] 10

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="ETX\_club" & campaign=="2016/2017"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # ETX/MAE 2016

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by zone  
## W = 80, p-value = 4.571e-05  
## alternative hypothesis: true location shift is not equal to 0

## [1] 0.0003656474

w=wilcox.test(OD\_cor~zone,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="MAE" | zone=="PDT") & campaign=="2016/2017"))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="MAE" & campaign=="2016/2017"))

## [1] 10

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2016/2017"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # PDT/MAE 2016

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by zone  
## W = 76, p-value = 0.0005485  
## alternative hypothesis: true location shift is not equal to 0

## [1] 0.004387769

w=wilcox.test(OD\_cor~zone,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="ETX\_club" | zone=="PDT") & campaign=="2016/2017"))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2016/2017"))

## [1] 10

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="ETX\_club" & campaign=="2016/2017"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # PDT/ETX 2016

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by zone  
## W = 57, p-value = 0.6305  
## alternative hypothesis: true location shift is not equal to 0

## [1] 1

w=wilcox.test(OD\_cor~zone,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="ETX\_club" | zone=="PDT") & campaign=="2015/2016"))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2015/2016"))

## [1] 10

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="ETX\_club" & campaign=="2015/2016"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # PDT/ETX 2015

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by zone  
## W = 83, p-value = 0.0115  
## alternative hypothesis: true location shift is not equal to 0

## [1] 0.09196995

# among years   
  
w=wilcox.test(OD\_cor~campaign,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="ETX\_club" & (campaign=="2016/2017" | campaign=="2015/2016" ))))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="ETX\_club" & campaign=="2016/2017"))

## [1] 10

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="ETX\_club" & campaign=="2015/2016"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # ETX 2015/2016

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by campaign  
## W = 28, p-value = 0.1051  
## alternative hypothesis: true location shift is not equal to 0

## [1] 0.8409795

w=wilcox.test(OD\_cor~campaign,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="PDT" & (campaign=="2016/2017" | campaign=="2015/2016" ))))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2016/2017"))

## [1] 10

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2015/2016"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # PDT 2015/2016

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by campaign  
## W = 50, p-value = 1  
## alternative hypothesis: true location shift is not equal to 0

## [1] 1

w=wilcox.test(OD\_cor~campaign,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="PDT" & (campaign=="2016/2017" | campaign=="2011/2012" ))))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2011/2012"))

## [1] 16

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2016/2017"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # PDT 2011/2016

##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by campaign  
## W = 46, p-value = 0.07722  
## alternative hypothesis: true location shift is not equal to 0

## [1] 0.6177266

w=wilcox.test(OD\_cor~campaign,subset(subset(sku\_epidemio\_ams,stage=="adult"),(zone=="PDT" & (campaign=="2015/2016" | campaign=="2011/2012" ))))  
nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2011/2012"))

## [1] 16

nrow(subset(subset(sku\_epidemio\_ams,stage=="adult"),zone=="PDT" & campaign=="2015/2016"))

## [1] 10

w;p.adjust(w$p.value,method="bonferroni",n=ntests) # PDT 2011/2015

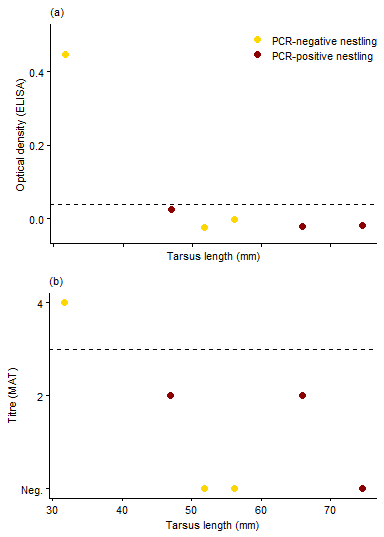
##   
## Wilcoxon rank sum test  
##   
## data: OD\_cor by campaign  
## W = 41, p-value = 0.04085  
## alternative hypothesis: true location shift is not equal to 0

## [1] 0.3267723

remove(sku\_stat\_elisa,sku\_stat\_mat,w,ntests)

## 1.5. Chicks

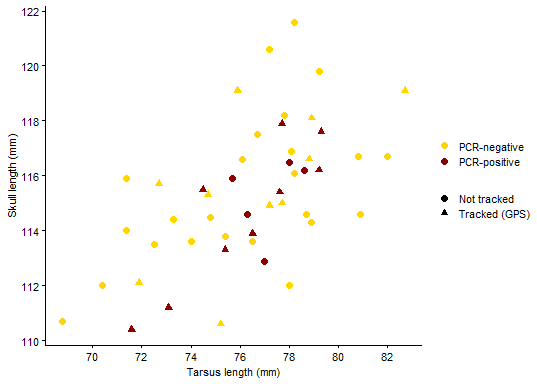
# OD / tarsus  
p1=ggplot(subset(sku\_epidemio,stage=="chick" & campaign=="2016/2017"),aes(x=as.numeric(as.character(tarsus)),y=OD\_cor,colour=as.factor(detect\_pos))) +   
 geom\_hline(aes(yintercept=cut99),linetype="dashed",size=0.5)+  
 geom\_point(size=2,stroke=0.5)+  
 ylab("Optical density (ELISA)")+xlab("Tarsus length (mm)")+  
 scale\_y\_continuous(breaks=seq(0,0.5,0.2),limits=c(-0.04,0.5))+  
 scale\_color\_manual(values=c("gold","red4"),labels=c("PCR-negative nestling","PCR-positive nestling"))+  
 ggtheme+  
 theme(legend.justification=c(1,1),legend.position=c(1,1),  
 plot.margin=margin(5,5,5,11),  
 axis.text.x=element\_blank())+  
 ggtitle("(a)")  
  
p2=ggplot(subset(sku\_epidemio,stage=="chick" & campaign=="2016/2017"),aes(x=as.numeric(as.character(tarsus)),y=titer\_log,colour=as.factor(detect\_pos))) +   
 geom\_hline(aes(yintercept=3),linetype="dashed",size=0.5)+  
 geom\_point(size=2,stroke=0.5)+  
 ylab("Titre (MAT)")+xlab("Tarsus length (mm)")+  
 scale\_y\_continuous(breaks=seq(0,4,2),labels=c("Neg.","2","4"))+  
 scale\_color\_manual(values=c("gold","red4"),labels=c("PCR-negative nestling","PCR-positive nestling"))+  
 ggtheme+  
 theme(legend.position="none")+  
 ggtitle("(b)")  
  
p=grid.arrange(p1,p2,nrow=2)



ggsave("r/exports/sku\_epidemio\_chicks.pdf",p,width=10,height=14,units="cm",dpi=300)  
  
remove(p1,p2,p)

## 1.6. Miscellaneous

sku\_epidemio\_ams$tracked=0  
sku\_epidemio\_ams[!is.na(sku\_epidemio\_ams$logger),"tracked"]=1  
  
ggplot(subset(sku\_epidemio\_ams,stage=="adult" & !is.na(detect\_pos)),aes(x=tarsus,y=skull,colour=as.factor(detect\_pos),shape=as.factor(tracked))) +   
 geom\_point(size=2,stroke=0.5)+  
 scale\_y\_continuous(breaks=seq(110,122,2))+  
 scale\_x\_continuous(breaks=seq(70,82,2))+  
 ylab("Skull length (mm)")+xlab("Tarsus length (mm)")+  
 scale\_color\_manual(values=c("gold","red4"),labels=c("PCR-negative","PCR-positive"))+  
 scale\_shape\_manual(values=c(16,17),labels=c("Not tracked","Tracked (GPS)"))+  
 ggtheme+  
 theme(legend.direction="vertical",legend.box="vertical")



ggsave("r/exports/sku\_size.pdf",width=14,height=10,units="cm",dpi=300)

# 2. Tracking data

if(grepl("/",as.character(sku\_gps$date[1]))){  
 sku\_gps$date=as.Date(sku\_gps$date,format="%d/%m/%Y")  
}else{sku\_gps$date=as.Date(sku\_gps$date)}  
  
sku\_gps$longitude=as.numeric(as.character(sku\_gps$longitude))  
sku\_gps$latitude=as.numeric(as.character(sku\_gps$latitude))  
  
sku\_gps$time=as.character(sku\_gps$time)  
sku\_gps=sku\_gps[with(sku\_gps,order(logger,date,time)),]  
  
sku\_gps$datetime=as.POSIXct(paste(sku\_gps$date, sku\_gps$time), format="%Y-%m-%d %H:%M:%S")  
  
sku\_gps$nesting=1  
  
for(i in 2:nrow(sku\_gps)){  
   
 # ouf of nesting area -> not nesting  
 if(sku\_gps$nest[i]==0){   
 sku\_gps$nesting[i]=0  
 }   
 # first nest=1 (in nesting area) after a serie of 0s -> gps\_no\_nest  
 else if(sku\_gps$nest[i]==1 &   
 sku\_gps$nest[i-1]==0 & sku\_gps$logger[i]==sku\_gps$logger[i-1]){  
 sku\_gps$nesting[i]=0  
 }   
}  
  
sku\_gps$etx=0  
sku\_gps[which(sku\_gps$longitude>77.515671 & sku\_gps$longitude<77.539308 & sku\_gps$latitude<(-37.851135) & sku\_gps$latitude>(-37.869850)),"etx"]=1  
sku\_gps[which(is.na(sku\_gps$longitude)),"etx"]=NA  
sku\_gps[which(sku\_gps$nest==1),"etx"]=0  
  
sku\_gps$fernand=0  
sku\_gps[which(sku\_gps$longitude>77.509971 & sku\_gps$longitude<77.523328 & sku\_gps$latitude>(-37.851135) & sku\_gps$latitude<(-37.832814)),"fernand"]=1  
sku\_gps[which(is.na(sku\_gps$longitude)),"fernand"]=NA  
sku\_gps[which(sku\_gps$nest==1),"fernand"]=0

Counts

loggers=as.character(unique(sku\_gps$logger))  
  
sku\_gps\_counts=as.data.frame(matrix(ncol=9,nrow=length(loggers)))  
colnames(sku\_gps\_counts)=c("logger","loc\_all","loc\_not\_nesting","loc\_not\_nest","etx","fernand","date\_first","date\_last","day\_nb")  
  
for (i in 1:length(loggers)){  
 sku\_gps\_counts$logger[i]=loggers[i]  
 sku\_gps\_counts$loc\_all[i]=sum(sku\_gps$logger==loggers[i] & !is.na(sku\_gps$latitude))  
 sku\_gps\_counts$loc\_not\_nesting[i]=sum(sku\_gps$logger==loggers[i] & !is.na(sku\_gps$latitude) & sku\_gps$nesting==0)  
 sku\_gps\_counts$loc\_not\_nest[i]=sum(sku\_gps$logger==loggers[i] & !is.na(sku\_gps$latitude) & sku\_gps$nest==0)  
 sku\_gps\_counts$etx[i]=sum(sku\_gps$logger==loggers[i] & !is.na(sku\_gps$latitude) & sku\_gps$etx==1)  
 sku\_gps\_counts$fernand[i]=sum(sku\_gps$logger==loggers[i] & !is.na(sku\_gps$latitude) & sku\_gps$fernand==1)  
 sku\_gps\_counts$date\_first[i]=min(subset(sku\_gps,logger==loggers[i])$date)  
 sku\_gps\_counts$date\_last[i]=max(subset(sku\_gps,logger==loggers[i])$date)  
 sku\_gps\_counts$day\_nb[i]=sku\_gps\_counts$date\_last[i]-sku\_gps\_counts$date\_first[i]  
}  
  
sku\_gps\_counts$date\_first=as.Date(sku\_gps\_counts$date\_first,origin="1970-01-01")  
sku\_gps\_counts$date\_last=as.Date(sku\_gps\_counts$date\_last,origin="1970-01-01")  
sku\_gps\_counts=sku\_gps\_counts[with(sku\_gps\_counts,order(date\_first)),]  
  
sku\_gps\_counts$etx\_prop=sku\_gps\_counts$etx/sku\_gps\_counts$loc\_all  
sku\_gps\_counts$fernand\_prop=sku\_gps\_counts$fernand/sku\_gps\_counts$loc\_all  
  
sku\_gps\_counts

## logger loc\_all loc\_not\_nesting loc\_not\_nest etx fernand  
## 10 GPS\_HAR20\_15 2748 703 644 457 148  
## 11 GPS\_HAR21\_15 902 91 81 33 8  
## 12 GPS\_HAR22\_15 910 0 0 0 0  
## 13 GPS\_HAR23\_15 3088 1058 944 430 312  
## 14 GPS\_HAR24\_15 16 0 0 0 0  
## 15 GPS\_HAR25\_15 1075 180 167 46 29  
## 16 GPS\_HAR26\_15 2044 0 0 0 0  
## 17 GPS\_HAR27\_15 127 0 0 0 0  
## 18 GPS\_HAR29\_15 88 10 9 4 4  
## 1 GPS\_HAR01\_16 6122 1118 1035 378 247  
## 2 GPS\_HAR02\_16 1177 450 432 272 92  
## 3 GPS\_HAR03\_16 1518 392 311 70 134  
## 4 GPS\_HAR04\_16 1285 248 216 166 44  
## 5 GPS\_HAR06\_16 912 176 133 115 16  
## 6 GPS\_HAR07\_16 571 50 39 11 5  
## 7 GPS\_HAR08\_16 728 73 62 47 2  
## 8 GPS\_HAR09\_16 1970 448 364 360 0  
## 9 GPS\_HAR10\_16 1713 380 316 190 44  
## date\_first date\_last day\_nb etx\_prop fernand\_prop  
## 10 2015-12-03 2016-03-14 102 0.16630277 0.053857351  
## 11 2015-12-03 2016-01-25 53 0.03658537 0.008869180  
## 12 2015-12-03 2015-12-06 3 0.00000000 0.000000000  
## 13 2015-12-03 2016-01-11 39 0.13924870 0.101036269  
## 14 2015-12-03 2015-12-03 0 0.00000000 0.000000000  
## 15 2015-12-03 2016-01-29 57 0.04279070 0.026976744  
## 16 2015-12-06 2016-01-17 42 0.00000000 0.000000000  
## 17 2015-12-06 2015-12-06 0 0.00000000 0.000000000  
## 18 2015-12-06 2015-12-06 0 0.04545455 0.045454545  
## 1 2016-11-26 2017-02-01 67 0.06174453 0.040346292  
## 2 2016-11-26 2016-12-08 12 0.23109601 0.078164826  
## 3 2016-12-01 2017-02-24 85 0.04611331 0.088274045  
## 4 2016-12-01 2017-02-25 86 0.12918288 0.034241245  
## 5 2016-12-02 2017-02-26 86 0.12609649 0.017543860  
## 6 2016-12-02 2017-01-25 54 0.01926445 0.008756567  
## 7 2016-12-02 2017-01-15 44 0.06456044 0.002747253  
## 8 2016-12-02 2017-01-18 47 0.18274112 0.000000000  
## 9 2016-12-02 2017-01-19 48 0.11091652 0.025685931

enough\_data=sku\_gps\_counts[sku\_gps\_counts$loc\_not\_nesting>0 & sku\_gps\_counts$day\_nb>0,"logger"]  
length(enough\_data)

## [1] 13

enough\_data\_counts=sku\_gps\_counts[sku\_gps\_counts$loc\_not\_nesting>0 & sku\_gps\_counts$day\_nb>0,]  
quantile(enough\_data\_counts$day\_nb)

## 0% 25% 50% 75% 100%   
## 12 47 54 85 102

sku\_gps=subset(sku\_gps,is.element(logger,enough\_data))  
sku\_gps$logger=as.character(sku\_gps$logger)

## 2.1. Representativeness

Method described in Lascelles et al.,2016,Diversity and Distributions

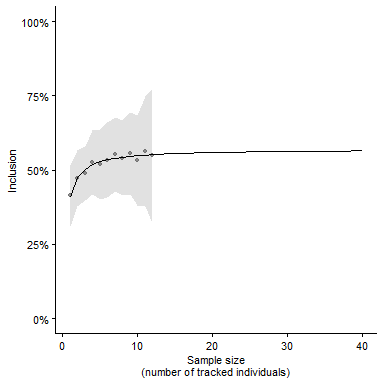
DataGroup=subset(sku\_gps,nest==0)  
  
DataGroup$ID <- as.character(DataGroup$logger)  
DataGroup$Longitude <- as.numeric(as.character(DataGroup$longitude))  
DataGroup$Latitude <- as.numeric(as.character(DataGroup$latitude))  
DataGroup$DateTime <- paste(DataGroup$date,DataGroup$time,sep=" ")  
DataGroup$DateTime <- as.POSIXct(strptime(DataGroup$DateTime,format="%Y-%m-%d %H:%M:%S"))  
DataGroup$TrackTime <- as.double(DataGroup$DateTime)  
  
# convert to SpatialPointsDataFrame and project  
if(class(DataGroup)!= "SpatialPointsDataFrame")   
{  
 mid\_point<-data.frame(centroid(cbind(DataGroup$Longitude,DataGroup$Latitude)))  
 DataGroup.Wgs <- SpatialPoints(data.frame(DataGroup$Longitude,DataGroup$Latitude),proj4string=CRS("+proj=longlat + datum=wgs84"))  
 DgProj <- CRS(paste("+proj=laea +lon\_0=",mid\_point$lon," +lat\_0=",mid\_point$lat,sep=""))  
 DataGroup.Projected <- spTransform(DataGroup.Wgs,CRS=DgProj)  
 DataGroup <- SpatialPointsDataFrame(DataGroup.Projected,data=DataGroup)  
}else{DgProj<-DataGroup@proj4string}  
  
DataGroup$X <- DataGroup@coords[,1]  
DataGroup$Y <- DataGroup@coords[,2]  
  
# representativeness by boostraping  
Scale=0.250  
Iteration=200  
  
UIDs <- unique(DataGroup$ID)  
Ntrips <- length(UIDs)  
Output <- data.frame(SampleSize=1:Ntrips,InclusionMean=rep(0,Ntrips),InclusionSD=rep(0,Ntrips))  
UDLev <- 50 # isopleth  
for (N in 1:(Ntrips - 1))  
{  
 par(mfrow=c(1,2),mai=c(0.4,0.4,0.4,0.4))  
 Coverage <- NULL  
 Inclusion <- NULL  
 History <- NULL  
 BoundBox <- bbox(DataGroup)  
 for(i in 1:Iteration)  
 {  
 RanNum <- sample(UIDs,N,replace=F)  
 SelectedCoords <- coordinates(DataGroup[DataGroup$ID %in% RanNum,])  
 NotSelected <- DataGroup[!DataGroup$ID %in% RanNum,]  
   
 Temp <- data.frame(SelectedCoords[,1],SelectedCoords[,2])  
 Ext <- (min(Temp[,1]) + 3 \* diff(range(Temp[,1])))  
 if(Ext < (Scale \* 1000 \* 2)) {BExt <- ceiling((Scale \* 1000 \* 3)/(diff(range(Temp[,1]))))} else {BExt <- 3}  
   
 xy <- SpatialPoints(coords=data.frame(SelectedCoords[,1],SelectedCoords[,2]))  
 KDE.Surface <- kernelUD(xy,h=Scale\*1000,grid=180,extent=BExt,same4all=FALSE)  
 KDE.UD <- getverticeshr(KDE.Surface,percent=50)  
 KDE.UD@proj4string <- DgProj  
 Overlain <- over(as(NotSelected,"SpatialPoints"),as(KDE.UD,"SpatialPolygons"))  
 Inclusion[i] <- length(which(Overlain == 1))/nrow(NotSelected)  
   
 #plot(NotSelected,xlim=c(BoundBox[1,]),ylim=c(BoundBox[2,]),cex=0.5,col="grey")  
 #points(SelectedCoords,xlim=c(BoundBox[1,]),ylim=c(BoundBox[2,]),cex=0.5,col="red")  
 #plot(KDE.UD,add=T)  
 }  
 Output[N,]$InclusionMean <- mean(Inclusion)  
 Output[N,]$InclusionSD <- sd(Inclusion)  
}  
  
Output <- Output[1:nrow(Output)-1,]  
  
M1 <- nls((InclusionMean ~ (a\*SampleSize)/(1+b\*SampleSize)),data=Output,start=list(a=1,b=0.1))  
summary(M1)

##   
## Formula: InclusionMean ~ (a \* SampleSize)/(1 + b \* SampleSize)  
##   
## Parameters:  
## Estimate Std. Error t value Pr(>|t|)   
## a 1.4262 0.1228 11.61 3.98e-07 \*\*\*  
## b 2.5013 0.2329 10.74 8.23e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01016 on 10 degrees of freedom  
##   
## Number of iterations to convergence: 9   
## Achieved convergence tolerance: 2.078e-06

NPred <- max(c(100,nrow(Output)+20))  
PredData <- data.frame(SampleSize=seq(1:NPred))  
P2 <- predict(M1,PredData)  
  
Asymptote <- (summary(M1)$coefficients[1]/summary(M1)$coefficients[2])  
RepresentativeValue <- P2[nrow(Output)]/(summary(M1)$coefficients[1]/summary(M1)$coefficients[2])\*100  
RepresentativeValue

## [1] 96.77577

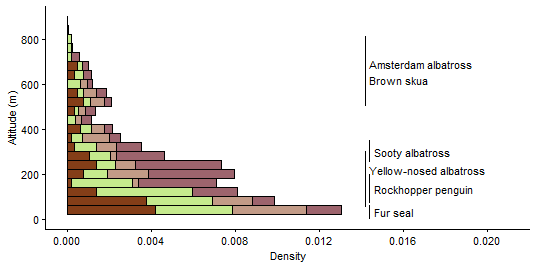
# graphic  
ggplot() +   
 geom\_ribbon(data=Output,aes(x=SampleSize,ymin=InclusionMean-InclusionSD,ymax=InclusionMean+InclusionSD),fill="gray70",alpha=0.4)+  
 geom\_point(data=Output,aes(x=SampleSize,y=InclusionMean),colour="gray40",size=1,alpha=0.7)+  
 geom\_line(aes(y=P2,x=seq(1,length(P2),1)),colour="#000000",size=0.5)+  
 ylab("Inclusion")+xlab("Sample size\n(number of tracked individuals)")+  
 ggtheme+  
 xlim(c(1,40))+  
 scale\_y\_continuous(labels=scales::percent,lim=c(0,1))



ggsave("r/exports/sku\_gps\_inclusion.pdf",width=10,height=10,units="cm",dpi=300)  
  
remove(BoundBox,mid\_point,Output,SelectedCoords,Temp,BExt,Coverage,DataGroup,DataGroup.Projected,DataGroup.Wgs,DgProj,Ext,i,Inclusion,Iteration,KDE.Surface,KDE.UD,N,NotSelected,Ntrips,Overlain,RanNum,Scale,UDLev,UIDs,xy)

## 2.2. Altitude

species\_altitude$density=as.numeric(c(0.0144,0.0144,0.0142,0.0144,0.0142,0.0142))  
species\_altitude$lab\_x=as.numeric(c(0.0146,0.0146,0.0144,0.0146,0.0144,0.0144))  
species\_altitude$lab\_position=as.numeric(c(30,135,220,300,620,690))  
species\_altitude$color=as.numeric(c("A","B","B","B","C","D"))  
  
ggplot(data=subset(sku\_gps,nest==0))+  
 geom\_histogram(aes(x=altitude,y=..density..,fill=as.factor(logger)),  
 binwidth=40,  
 colour="black",size=0.25)+  
 geom\_segment(data=species\_altitude,aes(x=altitude\_min,xend=altitude\_max,y=density,yend=density),size=0.5)+  
 geom\_text(data=species\_altitude,aes(label=label,x=lab\_position,y=as.numeric(as.character(lab\_x))),  
 hjust="outward",size=3)+  
 coord\_flip()+  
 scale\_fill\_manual(values=c("#9d646d","#c29b87","#c5ea8d","#843e18"))+  
 xlab("Altitude (m)")+ylab("Density")+  
 scale\_y\_continuous(limits=c(0,0.0210),breaks=seq(0,0.1,0.004))+  
 scale\_x\_continuous(limits=c(0,900),breaks=seq(0,900,200))+  
 ggtheme+theme(legend.position="none")



ggsave("r/exports/sku\_gps\_altitude.pdf",width=14,height=7,units="cm",dpi=300)

## 2.3. Animation

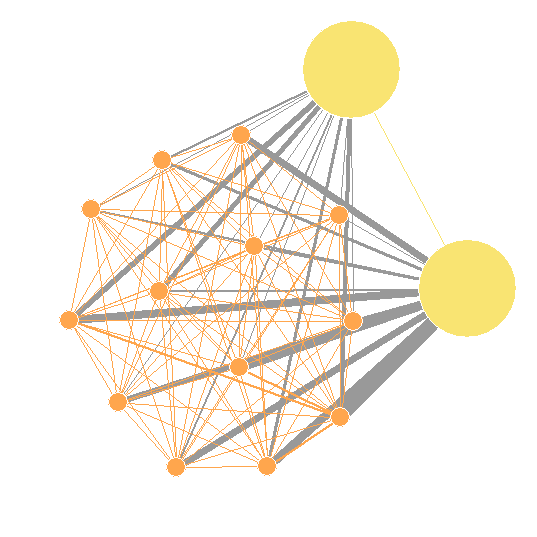
sku\_gps\_sel=subset(sku\_gps,grepl("\_16",logger))  
sku\_gps\_sel=subset(sku\_gps,date>=as.Date("2016-12-03") & date<=as.Date("2016-12-03"))  
  
m=move(sku\_gps\_sel$longitude,sku\_gps\_sel$latitude,proj=CRS("+proj=longlat +ellps=WGS84"),time=sku\_gps\_sel$datetime,animal=sku\_gps\_sel$logger,data=sku\_gps\_sel)  
  
# find the command or directory to convert tool of ImageMagick  
conv\_dir=get\_libraries()  
  
out\_dir=paste0(getwd(),"/export")  
img\_title="Movements of breeding skuas on Amsterdam Island - 3 December 2016"  
img\_sub=""   
img\_caption="Projection: Geographical, WGS84; Sources: Gamble et al. 2019; Google Maps"  
  
# warning, does not work because of new Google API terms of service  
# animate\_move(m,out\_dir,conv\_dir,tail\_elements=10,paths\_mode="true\_data",frames\_nmax=0,log\_level=1,extent\_factor=0.0002,out\_format="mov",overwrite=T,time\_size=5,frames\_fps=5,map\_elements=F,out\_name="sku\_gps\_animation")  
  
remove(sku\_gps\_sel,m,conv\_dir,out\_dir,img\_caption,img\_sub,img\_title)

## 2.4. Network

enough\_data=sku\_gps\_counts[sku\_gps\_counts$day\_nb>0 & sku\_gps\_counts$loc\_not\_nesting>0,"logger"]  
length(enough\_data)

## [1] 13

sku\_gps=subset(sku\_gps,is.element(logger,enough\_data))  
sku\_gps$logger=as.character(sku\_gps$logger)  
  
sku\_gps\_counts=subset(sku\_gps\_counts,is.element(logger,enough\_data))  
  
remove(enough\_data)  
   
etx=cbind(subset(sku\_gps\_counts,select=c(logger)),rep("etx",nrow(sku\_gps\_counts)),subset(sku\_gps\_counts,select=c(etx\_prop)))  
colnames(etx)=c("from","to","width")  
fernand=cbind(subset(sku\_gps\_counts,select=c(logger)),rep("fernand",nrow(sku\_gps\_counts)),subset(sku\_gps\_counts,select=c(fernand\_prop)))  
colnames(fernand)=c("from","to","width")  
edges=rbind(fernand,etx)  
edges$type="skutoyna"  
  
skutosku=as.data.frame(matrix(ncol=3,nrow=length(sku\_gps\_counts$logger)^2))  
colnames(skutosku)=c("from","to","width")  
skutosku$type="skutosku"  
  
l=1  
for (i in (1:length(sku\_gps\_counts$logger))){  
 for (j in (1:length(sku\_gps\_counts$logger))){  
 skutosku$from[l]=sku\_gps\_counts$logger[i]  
 skutosku$to[l]=sku\_gps\_counts$logger[j]  
 skutosku$width[l]=sku\_gps\_counts$etx\_prop[i]\*sku\_gps\_counts$etx\_prop[j]+sku\_gps\_counts$fernand\_prop[i]\*sku\_gps\_counts$fernand\_prop[j]  
 l=l+1  
 }  
}  
  
edges=rbind(edges,skutosku)  
remove(etx,fernand,skutosku)  
edges=subset(edges,as.character(from)!=as.character(to))  
  
nodes=as.data.frame(cbind(sku\_gps\_counts$logger,rep(1,length(sku\_gps\_counts$logger)),rep("sku",length(sku\_gps\_counts$logger))))  
colnames(nodes)=c("id","size","species")  
  
nodes$id=as.character(nodes$id)  
nodes$species=as.character(nodes$species)  
nodes$size=as.numeric(nodes$size)  
  
edges$to=as.character(edges$to)  
  
# link albatross plots to make them clother on the graphic (visualization purpose only)  
edges=rbind(edges,c("etx","fernand",0.0001,"ynatoyna"))  
  
edges$from=as.factor(edges$from)  
edges$to=as.factor(edges$to)  
  
# comments for node size proportional of skuas' visits  
nodes=rbind(nodes,c("fernand",5,"yna")) #;nodes[nodes$id=="fernand","size"]=sum(sku\_gps\_counts$fernand)/500  
nodes=rbind(nodes,c("etx",5,"yna")) #;nodes[nodes$id=="etx","size"]=sum(sku\_gps\_counts$etx)/500  
  
nodes$size=as.numeric(nodes$size)  
edges$width=as.numeric(edges$width)\*3  
  
nodes$species=as.factor(nodes$species)  
  
ams\_net=graph\_from\_data\_frame(d=edges,vertices=nodes,directed=T)   
  
# sizes  
V(ams\_net)$size=V(ams\_net)$size\*10  
E(ams\_net)$width=E(ams\_net)$width\*20  
  
# colors  
E(ams\_net)$edge.color="gray80"  
  
colrs=c("#ffa64d","#f9e472");  
V(ams\_net)$color=colrs[as.numeric(as.factor(V(ams\_net)$species))]  
  
colrs=c("#ffa64d","grey60","#f9e472")  
E(ams\_net)$color=colrs[as.numeric(as.factor(E(ams\_net)$type))]  
  
V(ams\_net)$frame.color="white"  
  
# graphic  
par(mar=c(0,0,0,0))  
plot(ams\_net,edge.arrow.size=0,vertex.label=NA)



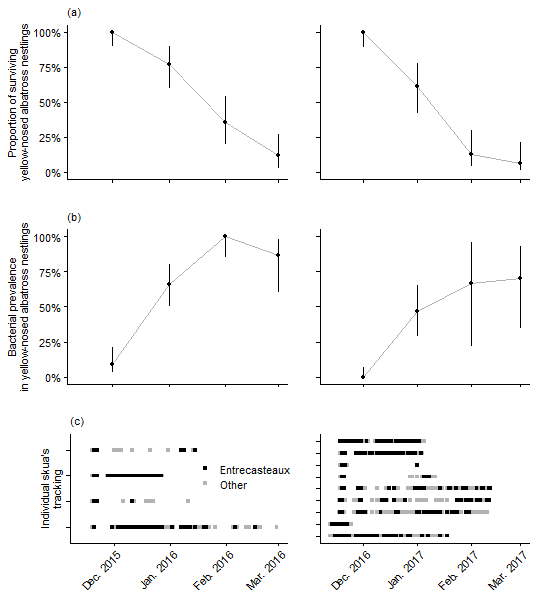
tiff("r/exports/sku\_gps\_network.tiff",width=14,height=14,units="cm",res=300)  
par(mar=c(0,0,0,0)+.1)  
plot(ams\_net,edge.arrow.size=0,vertex.label=NA)   
dev.off()

## png   
## 2

remove(colrs,nodes,edges)

# 3. Albatross monitoring

if(grepl("/",as.character(albatross\_nestlings$month[1]))){  
 albatross\_nestlings$month=as.Date(albatross\_nestlings$month,format="%d/%m/%Y")  
}else{albatross\_nestlings$month=as.Date(albatross\_nestlings$month)}  
  
p11=ggplot(data=subset(albatross\_nestlings,month<as.Date("2016-07-01")))+  
 geom\_line(aes(x=month,y=survival\_rate),size=0.5,colour="gray70")+  
 geom\_point(aes(x=month,y=survival\_rate),size=1)+  
 geom\_errorbar(aes(x=month,ymin=survival\_ci\_min,ymax=survival\_ci\_max),width=0,size=0.5)+  
 scale\_x\_date(breaks=c(as.Date("2015-12-15"),  
 as.Date("2016-01-15"),  
 as.Date("2016-02-15"),  
 as.Date("2016-03-15")),  
 labels=c("Dec. 2015","Jan. 2016","Feb. 2016","Mar. 2016"),  
 limits=c(as.Date("2015-11-26"),  
 as.Date("2016-03-15")))+  
 ylab("Proportion of surviving\nyellow-nosed albatross nestlings")+xlab("")+  
 scale\_y\_continuous(labels=scales::percent)+  
 coord\_cartesian(ylim=c(0,1))+  
 ggtitle("(a)")+  
 ggtheme+theme(axis.text.x=element\_blank())  
  
p12=ggplot(data=subset(albatross\_nestlings,month<as.Date("2017-07-01")))+  
 geom\_line(aes(x=month,y=survival\_rate),size=0.5,colour="gray70")+  
 geom\_point(aes(x=month,y=survival\_rate),size=1)+  
 geom\_errorbar(aes(x=month,ymin=survival\_ci\_min,ymax=survival\_ci\_max),width=0,size=0.5)+  
 scale\_x\_date(breaks=c(as.Date("2016-12-15"),  
 as.Date("2017-01-15"),  
 as.Date("2017-02-15"),  
 as.Date("2017-03-15")),  
 labels=c("Dec. 2016","Jan. 2017","Feb. 2017","Mar. 2017"),  
 limits=c(as.Date("2016-11-26"),  
 as.Date("2017-03-15")))+  
 ylab(" ")+xlab("")+  
 scale\_y\_continuous(labels=scales::percent)+  
 coord\_cartesian(ylim=c(0,1))+  
 ggtitle(" ")+  
 ggtheme+theme(axis.text.y=element\_blank())+theme(axis.text.x=element\_blank())  
  
p1=plot\_grid(p11,p12,nrow=1,rel\_widths=c(0.55,0.45))  
  
p21=ggplot(data=subset(albatross\_nestlings,month<as.Date("2016-07-01")))+  
 geom\_line(aes(x=month,y=prevalence),size=0.5,colour="gray70")+  
 geom\_point(aes(x=month,y=prevalence),size=1)+  
 geom\_errorbar(aes(x=month,ymin=prevalence\_ci\_min,ymax=prevalence\_ci\_max),width=0,size=0.5)+  
 scale\_x\_date(breaks=c(as.Date("2015-12-15"),  
 as.Date("2016-01-15"),  
 as.Date("2016-02-15"),  
 as.Date("2016-03-15")),  
 labels=c("Dec. 2015","Jan. 2016","Feb. 2016","Mar. 2016"),  
 limits=c(as.Date("2015-11-26"),  
 as.Date("2016-03-15")))+  
 ylab("Bacterial prevalence\nin yellow-nosed albatross nestlings")+xlab("")+  
 scale\_y\_continuous(labels=scales::percent)+  
 coord\_cartesian(ylim=c(0,1))+  
 ggtitle("(b)")+  
 ggtheme+theme(axis.text.x=element\_blank())  
  
p22=ggplot(data=subset(albatross\_nestlings,month<as.Date("2017-07-01")))+  
 geom\_line(aes(x=month,y=prevalence),size=0.5,colour="gray70")+  
 geom\_point(aes(x=month,y=prevalence),size=1)+  
 geom\_errorbar(aes(x=month,ymin=prevalence\_ci\_min,ymax=prevalence\_ci\_max),width=0,size=0.5)+  
 scale\_x\_date(breaks=c(as.Date("2016-12-15"),  
 as.Date("2017-01-15"),  
 as.Date("2017-02-15"),  
 as.Date("2017-03-15")),  
 labels=c("Dec. 2016","Jan. 2017","Feb. 2017","Mar. 2017"),  
 limits=c(as.Date("2016-11-26"),  
 as.Date("2017-03-15")))+  
 ylab(" ")+xlab("")+  
 scale\_y\_continuous(labels=scales::percent)+  
 coord\_cartesian(ylim=c(0,1))+  
 ggtitle(" ")+  
 ggtheme+theme(axis.text.y=element\_blank(),axis.text.x=element\_blank())  
  
p2=plot\_grid(p21,p22,nrow=1,rel\_widths=c(0.55,0.45))  
  
p31=ggplot()+  
 geom\_point(data=subset(sku\_gps,grepl("\_15",logger)),aes(x=date,y=logger,colour="other"),size=1,shape=15)+  
 geom\_point(data=subset(sku\_gps,grepl("\_15",logger) & etx==1),aes(x=date,y=logger,colour="ETX"),size=1,shape=15)+  
 scale\_x\_date(breaks=c(as.Date("2015-12-15"),  
 as.Date("2016-01-15"),  
 as.Date("2016-02-15"),  
 as.Date("2016-03-15")),  
 labels=c("Dec. 2015","Jan. 2016","Feb. 2016","Mar. 2016"),  
 limits=c(as.Date("2015-11-26"),  
 as.Date("2016-03-15")))+  
 scale\_color\_manual(values=c("black","gray70"),labels=c("Entrecasteaux","Other"))+  
 ylab("Individual skua's\ntracking")+xlab("")+  
 ggtitle("(c)")+  
 ggtheme+theme(axis.text.y=element\_blank(),axis.text.x=element\_text(angle=45,hjust=1),plot.margin=margin(5,5,5,30),  
 legend.position=c(1,0.65),legend.justification=c(1,0.5))  
  
p32=ggplot()+  
 geom\_point(data=subset(sku\_gps,grepl("\_16",logger)),aes(x=date,y=logger,colour="other"),size=1,shape=15)+  
 geom\_point(data=subset(sku\_gps,grepl("\_16",logger) & etx==1),aes(x=date,y=logger,colour="ETX"),size=1,shape=15)+  
 scale\_x\_date(breaks=c(as.Date("2016-12-15"),  
 as.Date("2017-01-15"),  
 as.Date("2017-02-15"),  
 as.Date("2017-03-15")),  
 labels=c("Dec. 2016","Jan. 2017","Feb. 2017","Mar. 2017"),  
 limits=c(as.Date("2016-11-26"),  
 as.Date("2017-03-15")))+  
 scale\_color\_manual(values=c("black","gray70"),labels=c("Entrecasteaux","Other"))+  
 ylab(" ")+xlab("")+  
 ggtitle(" ")+  
 ggtheme+theme(axis.text.y=element\_blank(),axis.text.x=element\_text(angle=45,hjust=1),plot.margin=margin(5,5,5,5),  
 legend.position="none")  
  
p3=plot\_grid(p31,p32,nrow=1,rel\_widths=c(0.55,0.45))  
  
p=plot\_grid(p1,p2,p3,align="v",nrow=3,rel\_heights=c(1,1,1))  
  
p



ggsave("r/exports/yna\_outbreak\_timing\_v2.pdf",p,width=14,height=16,units="cm",dpi=300)  
  
remove(p,p11,p12,p1,p21,p22,p2)

rmarkdown::render(“C:/Users/Gamble/PhD Eco-epidemiology/On\_going/data\_and\_analyses/ecopath/sku/ecopath\_sku\_notebook.rmd”,output\_format=“html\_document”)