Ecography

ECOG-03439
Karenyi, N., Sink, K., Nel, R., Clark, A. E. and Altwegg, R. 2018. Imperfect detection distorts depth-related trends in marine macrofaunal species richness. – Ecography doi: 10.1111/ecog.03439

Supplementary material

Supplementary Appendices

Natasha Karenyi, Kerry Sink, Ronel Nel and Res Altwegg. 2016. Detection probability trends distort species richness patterns: a marine macrofauna example. Submitted to Ecography

Current contact details of corresponding author: Natasha Karenyi, Statistics in Ecology, Environment and Conservation, Department of Statistical Sciences, University of Cape Town, Private Bag, Rondebosch, 7701; telephone number: +27 82 214 6545; e-mail address natashakarenyi@gmail.com

Appendix 1: Capture-recapture heterogeneity model code

```
#load packages

library(lattice)

library(coda)

library(rjags)


# read data

histories <- read.csv("Natasha full biota data set.csv")

covariates <- read.csv("Natasha covariates full.csv")


nstations <- length(unique(covariates$ID))          # number of stations

repls <- as.vector(table(covariates$ID))            # grabs per station

maxrepl <- max(repls)                       # maximum number of grabs

aug <- 600                                  # number of species to augment


# put observations into an array: rows = species, col = replicates, dim 3 = stations

y <- array(NA, dim=c(dim(histories)[1]+aug, maxrepl, nstations))


for (j in 1:repls[1]) y[, j, 1] <- c(histories[, j+1], rep(0, aug))

for (j in 1:repls[2]) y[, j, 2] <- c(histories[, repls[1]+j+1], rep(0, aug))
```

```
for (k in 3:dim(y)[3]) {
  for (j in 1:repls[k]) y[, j, k] <- c(histories[, sum(repls[1:k-1])+j+1], rep(0, aug))
} # end k

# calculate number of observed species per station
Nobs <- rep(NA, nstations)
for (i in 1:nstations) Nobs[i] <- sum(apply(y[,1:repls[i],i], 1, sum)>0)
# ********************************
# Specify model in BUGS language
# ********************************
sink("M_single station.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)                  # inclusion probability related to z matrix
alpha <- log(mean.p / (1-mean.p))    # heterogeneity of detection per species (logit)
mean.p ~ dunif(0, 1)                 # ave. detection probability of species

tau <- 1 / (sd * sd)                       #
sd ~ dunif(0, 3)                           #sd of norm dbn~species det prob on logit scale

# Likelihood
for (i in 1:M){
z[i] ~ dbern(omega)                        #occurrence = ecological process
eps[i] ~ dnorm(0, tau)               # estimate norm dbn for species det prob
```

```
y[i] ~ dbin(p.eff[i], nrepls)                    #detection = observation process
p.eff[i] <- z[i] * p[i]                          #can only be detected if z=1
p[i] <- 1 / (1 + exp(-lp[i]))                    #detection probability
lp[i] <- alpha + eps[i]#det prob has norm dbt variance influenced by species heterogeneity
} #i



# Derived quantities
N <- sum(z[])                                    #species richness
logN <- log(N)                                   # log(N)
} # end model
",fill = TRUE)
sink()


# looping over all stations
# ------------------------
output <- list(logN = matrix(NA, nrow=length(repls), ncol=5), N = matrix(NA,
nrow=length(repls), ncol=5), mean.p = matrix(NA, nrow=length(repls), ncol=5), sd =
matrix(NA, nrow=length(repls), ncol=5), omega = matrix(NA, nrow=length(repls), ncol=5),
Gelman = matrix(NA, nrow=length(repls), ncol=5))


n.adapt <- rep(50000, 42)                        #number of iterations discarded as burn-in
n.iter <- rep(400000, 42)                        # number of iterations
for(i in 1:length(repls)) {


   # Bundle data
   win.data <- list(y = apply(y[,,i],1,sum, na.rm=T), M = dim(y)[1], nrepls=repls[i])
   # Initial values
   inits <- function() list(z = rep(1, dim(y)[1]), sd = runif(1, 0.1, 0.9))
```

```
    # Parameters monitored
    params <- c("logN", "N", "mean.p", "sd", "omega")

    print(paste("now doing station", i))

    # Call JAGS
    # Compile the model.
    jm <- jags.model("M_single station.txt", win.data, inits, n.chains=3, n.adapt=n.adapt[i])
    jc2 <- coda.samples(jm, params, n.iter=n.iter[i])

    # Summarize the posteriors and save
    # saves log N, its mean, SD, 2.5th, 50th, and 97.5th percentile
    output$logN[i,] <- c(as.numeric(summary(jc2)$statistics[2,1:2]),
as.numeric(summary(jc2)$quantiles[2,c(1,3,5)]))

    # saves N, its SD, 2.5th, 50th, and 97.5th percentile
    output$N[i,] <- c(as.numeric(summary(jc2)$statistics[1,1:2]),
as.numeric(summary(jc2)$quantiles[1,c(1,3,5)]))

    # saves mean.p, its SD, 2.5th, 50th, and 97.5th percentile
    output$mean.p[i,] <- c(as.numeric(summary(jc2)$statistics[4,1:2]),
as.numeric(summary(jc2)$quantiles[4,c(1,3,5)]))

    # saves sd, its SD, 2.5th, 50th, and 97.5th percentile
    output$sd[i,] <- c(as.numeric(summary(jc2)$statistics[3,1:2]),
as.numeric(summary(jc2)$quantiles[3,c(1,3,5)]))

    # saves omega, its SD, 2.5th, 50th, and 97.5th percentile
```

```
    output$omega[i,] <- c(as.numeric(summary(jc2)$statistics[5,1:2]),
as.numeric(summary(jc2)$quantiles[5,c(1,3,5)]))


    # MCMC diagnostics

    # Rubin-Gelman statistic for N, mean.p, omega, and sd

    output$Gelman[i,] <- as.numeric(gelman.diag(jc2)$psrf[,1])    }
```

Appendix 2: Linear Model code for determining species richness-depth relationship

```
# ++++++++++++++++++++++++++++++++++++++++++++++++++
# Analyse species richness in relation to covariates
# ++++++++++++++++++++++++++++++++++++++++++++++++++
depth <- aggregate(covariates$depth, by=list(covariates$ID), mean)
depths <- scale(depth$x)

mud <- aggregate(covariates$X..mud, by=list(covariates$ID), mean)
muds <- scale(mud$x)

vf <- aggregate(covariates$vf, by=list(covariates$ID), mean)
vfs <- scale(vf$x)

# Define model in BUGS language using all physical variables
sink("bma.txt")
cat("
model {

# Priors
beta0 ~ dnorm(0, 0.0000001)          # intercept
beta.m ~ dnorm(0, 0.0000001)         # mud intercept
beta.d ~ dnorm(0, 0.0000001)         # depth intercept
beta.d2 ~ dnorm(0, 0.0000001)        # depth^2 intercept
beta.vf ~ dnorm(0, 0.0000001)        # very fine sand intercept

tau.station <- pow(sd.station, -2)
sd.station ~ dunif(0, 100)           # Species heterogeneity
# Likelihood
```

```
for (i in 1:n){

   logN[i] ~ dnorm(mu.station[i], tau.error[i])

   mu.station[i] ~ dnorm(mu[i], tau.station)

   mu[i] <- beta0  + beta.m * muds[i]  + beta.d * depths[i] + beta.d2* pow(depths[i],2) +
beta.vf*vfs[i]

   tau.error[i] <- pow(se.estimate[i], -2)

}

} # end model

",fill=TRUE)

sink()


# Bundle data

N.data <- list(logN = output$logN[,1], se.estimate = output$logN[,2],

    n = dim(output$logN)[1], depths=as.vector(depths), muds=as.vector(muds), vfs =
as.vector(vfs))


# Initial values

inits <- function() list(beta0 = rnorm(1), beta.m = rnorm(1), beta.d = rnorm(1), beta.d2 =
rnorm(1), beta.vf = rnorm(1), sd.station = runif(1,0,10))


# Parameters monitored

params <- c("beta0", "beta.m", "beta.d", "beta.d2", "beta.vf")


# Call JAGS

# Compile the model.

N.mod <- jags.model("bma.txt", N.data, inits, n.chains=3, n.adapt=1500)


# Draw samples from the posterior
```

```
Nc <- coda.samples(N.mod, params, n.iter=50000)


# View the Markov chains
plot(Nc, ask=T)
summary(Nc)
```

Appendix 3: Code for calculating WAIC for model selection for the hierarchical linear model of the species richness – depth relationship

The marginal probability distribution of y_i for the hierarchical models fitted is

$$f(y_i|\beta,\sigma_s^2) = \left(2\pi\left(\frac{\tau_i+\tau_s}{\tau_i\tau_s}\right)\right)^{-0.5} e^{\left(\left(\frac{\tau_i+\tau_s}{2}\right)\left(\frac{\tau_i y_i+\tau_s f_i}{\tau_i+\tau_s}\right)^2-\left(\frac{\tau_i y_i^2+\tau_s f_i^2}{2}\right)\right)}$$

for $i = 1, \ldots, n$. The fitted value for observation i is denoted as f_i while $\tau_s = 1/\sigma_s^2$ and $\tau_i = 1/\sigma_i^2$. The vectors y_i (argument *y_i*), σ_i (argument *se.estimate_i*), f_i (argument *f_i*) and σ_s (argument *sd.station.p*) are used as inputs in the R functions below in order to calculate WAIC for the models fitted in the main text. The R functions *l_p_y_i* and *p_y_i* are helper functions for the function *WAIC*. The code for the required R functions are as follows:

```
l_p_y_i<-function(y_i, se.estimate_i, f_i, sd.station.p){
        #this function returns the log of the marginal likelihood function evaluated at the posterior sample values.
        #y_i = the i'th value of y
        #se.estimate_i = the ith value of se.estimate
        #f_i = the mu[i] - obtained from posterior samples (post.mu) Note that this is not a vector and is in general a matrix.
        #sd.station.p is obtained from posterior samples

        s2_i<- se.estimate_i^2
        tau_i<- 1/s2_i
        s2_s<- sd.station.p^2
        tau_s<- 1/s2_s
        c1<- tau_i + tau_s

        #The following terms are inside the exponent
        t1<- 0.5*c1
        t2<- ((tau_i*y_i + tau_s*f_i)/c1)^2
        t3<- 0.5*(tau_i*(y_i^2) + tau_s*(f_i^2))
        exponent<- t1*t2 - t3
        t4<- -0.5*log( 2*pi*c1/(tau_i*tau_s) )
        t5<- t4 + exponent

        return(t5)
}

p_y_i<-function(y_i, se.estimate_i, f_i, sd.station.p){
        #returns the marginal likelihood function evaluated at the posterior sample values
        exp(l_p_y_i(y_i, se.estimate_i, f_i, sd.station.p))
}

WAIC<-function(y, se.estimate, f, sd.station){
        #the following function calculates the WAIC for the hierarchical model fitted

        n<- length(y) #the number of observations

        lpd.s<-0
        for (i in 1:n){
                lpd.s<- lpd.s +
                log(mean( p_y_i(y_i=y[i], se.estimate_i=se.estimate[i], f_i=f[,i],sd.station.p=sd.station)))
        }

        p.waic<-0
        for (i in 1:n){
                p.waic<- p.waic + (sd( l_p_y_i(y_i=y[i], se.estimate_i=se.estimate[i], f_i=f[,i], sd.station.p=sd.station))^2 )
        }
```

```
        return( -2*(lpd.s - p.waic ))
}

WAIC_calc = WAIC(y=y_i, se.estimate=se.estimate_i, f=post.mu, sd.station = sd.station.p)
```

Appendix 4: Description of Van Veen Grab Sampler

The van Veen grab sampler is an instrument used to sample sediments on the seafloor. It consists of two arms attached to two buckets. During sampling, the grab is sent down the water column in the open position (first picture). The grab is held open by chains attached to each arm attached to hooks. The weight of the grab (65kg) keeps the tension on these chains as the grab descends to the seafloor. Once the grab hits the seafloor, the tension on the chains is released and they fall from the hooks. When the grab is then lifted off the seafloor, the buckets close (second picture) and a 0.2m^2 sample is taken from the seafloor up to a depth of 20 cm depending on the sediment type.