

Supplementary material

Estimating transient populations of unmarked individuals at a migratory stopover site using generalized N-mixture models

Eunbi Kwon^a, Lawrence M. Houghton^a, Robert E. Settlage^b, Daniel H. Catlin^a, Sarah M. Karpanty^a, and James D. Fraser^a

^a Department of Fish and Wildlife Conservation, Virginia Tech, Blacksburg, VA 24061, USA.

^b Advanced Research Computing, Virginia Tech, Blacksburg, VA 24061, USA.

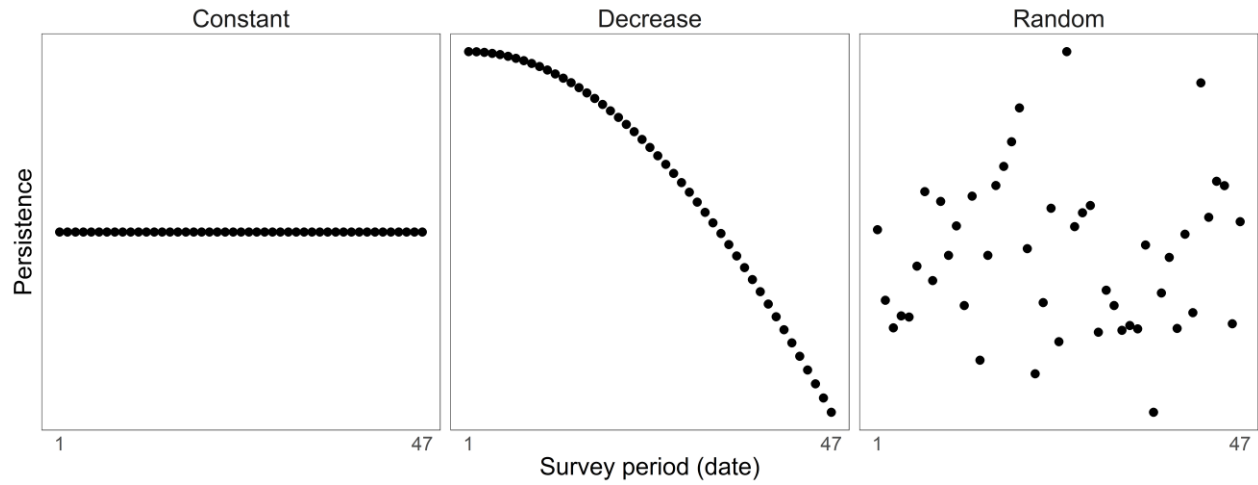
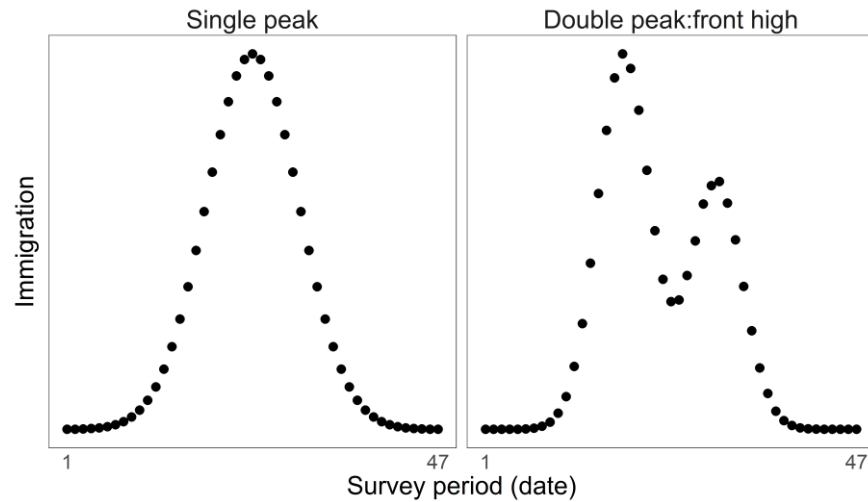
S1. R Code to simulate data and fit generalized N-mixture models

```
# R code was modified from Hostetler, J. A. and Chandler, R. B. (2015), Improved state-space  
# models for inference about spatial and temporal variation in abundance from count data.  
# Ecology, 96: 1713–1723. doi:10.1890/14-1487.1
```

```
library(unmarked)
```

```
# Simulated data mimicked a count survey data from 14 sites (M) and 47 survey occasions (T).  
# Initial abundance of individuals at the site was set to be ‘1’, and individual animals immigrated  
# (gam or gamma) or either died or left the site (om or omega) at preset rates. Mean detection  
# probability (p) was 0.3 or 0.8 and varied with two environmental variables, survey time  
# (‘time’) and tidal height at the time of survey (‘water’).
```

```
# Within-season trends of immigration (gam or gamma) and persistence (om or omega) were  
# allowed to take one of the following structures.
```

Hypothesized temporal structure for *persistence*:# Hypothesized temporal structure for *immigration*:

We presented R script to simulate scenarios where the mean detection probability is set at 0.8
 # ('high detection'). For simulations with the mean detection probability of 0.3 ('low detection'),
 # replace '0.8' in red-font into '0.3'.

Single peak immigration**# Scenario 1.**

```
# Omega is constant at 0.5.
# Gamma peaks in the middle of the survey period.
# Same model was used to generate data and estimate parameters.
```

Simulating data

```
sim2 <- function(lam=c(1), gam=c(0,1), om=c(0.5), p=c(0.8,1,-1), M=14, T=47)
{
  y <- matrix(NA, M, T)
  gamma <- matrix(NA, M, T)
  omega <- matrix(NA, M, T)
  det <- matrix(NA, M, T)

  N <- matrix(NA, M, T)
  S <- matrix(NA, M, T-1)
  G <- matrix(NA, M, T-1)

  site <- matrix(rnorm(M))
  decrease <- matrix(dnorm(seq(from=0, to=4, length.out=47), 0, 7),
                     nrow=14,ncol=47,byrow=TRUE)
  bell <- matrix(dnorm(seq(from=0, to=4, length.out=47), 2, 0.5),
                 nrow=14,ncol=47,byrow=TRUE)
  time <- matrix(rnorm(M*T, 1), M, T)
  water <- matrix(rnorm(M*T, 1), M, T)

  lambda <- exp(lam[1])
  gamma[] <- exp(gam[1] + gam[2]*bell)
  omega[] <- plogis(om[1])
  det[] <- plogis(p[1] + p[2]*time + p[3]*water)

  N[,1] <- rpois(M, lambda)
  for(t in 1:(T-1)) {
    S[,t] <- rbinom(M, N[,t], omega[,t])
    G[,t] <- rpois(M, gamma[,t])
    N[,t+1] <- S[,t] + G[,t]
  }

  y[] <- rbinom(M*T, N, det)

  return(list(N=N, y=y,
```

```

    covs=data.frame(site=site,decrease=decrease,bell=bell,time=time,water=water)))
  }

```

Model fitting for 500 iterations

```

nsim2 <- 500
simout2 <- matrix(NA, nsim2, 7)
predout <- matrix(NA, nrow=nsim2, ncol=47)
Nout <- matrix(NA, nrow=nsim2, ncol=47)

colnames(simout2) <- c('lam0', 'gam0', 'gam1', 'om0', 'p0', 'p1', 'p2')

for(i in 1:nsim2) {
  cat("sim2:", i, "\n")
  lam <- c(1)
  gam <- c(0,1)
  om <- c(0.5)
  p <- c(0.8,1,-1)
  M <- 14
  T <- 47

  sim2out <- sim2(lam, gam, om, p, M=M, T=T)
  y.sim2 <- sim2out$y
  covs <- sim2out$covs
  N <- sim2out$N
  cn <- colnames(covs)
  siteCovs <- covs[,grep("site", cn), drop=FALSE]
  yearlySiteCovs <- list(decrease = covs[,grep("decrease", cn)],
                        bell = covs[,grep("bell", cn)])
  obsCovs <- list(time = covs[,grep("time", cn)],
                  water = covs[,grep("water", cn)])
  umf2 <- unmarkedFramePCO(y = y.sim2, siteCovs=siteCovs,
                          yearlySiteCovs=yearlySiteCovs,
                          obsCovs=obsCovs, numPrimary=T)

  m2 <- pcountOpen(~1, ~bell, ~1, ~time+water, umf2, K=50,
                  control=list(trace=TRUE, REPORT=1))

  e <- coef(m2)
  simout2[i, ] <- e
  cat(" mle =", e, "\n")
}

```

Predict abundance

```

re <- ranef(m2)

```

```

preds <- bup(re)
model <- colSums(preds)
predout[i, ] <- model

true <- colSums(N)
Nout[i, ] <- true

}

```

Scenario 2.

```

# Omega decreases throughout the survey period.
# Gamma peaks in the middle of the survey period.
# Same model was used to generate data and estimate parameters.

```

Simulating data

```

sim2 <- function(lam=c(1), gam=c(0,1), om=c(0.5,1), p=c(0.8,1,-1), M=14, T=47)
{
  y <- matrix(NA, M, T)
  gamma <- matrix(NA, M, T)
  omega <- matrix(NA, M, T)
  det <- matrix(NA, M, T)

  N <- matrix(NA, M, T)
  S <- matrix(NA, M, T-1)
  G <- matrix(NA, M, T-1)

  site <- matrix(rnorm(M))
  decrease <- matrix(dnorm(seq(from=0, to=4, length.out=47), 0, 7),
                     nrow=14, ncol=47, byrow=TRUE)
  bell <- matrix(dnorm(seq(from=0, to=4, length.out=47), 2, 0.5),
                nrow=14, ncol=47, byrow=TRUE)
  time <- matrix(rnorm(M*T, 1), M, T)
  water <- matrix(rnorm(M*T, 1), M, T)

  lambda <- exp(lam[1])
  gamma[] <- exp(gam[1] + gam[2]*bell)
  omega[] <- plogis(om[1] + om[2]*decrease)
  det[] <- plogis(p[1] + p[2]*time + p[3]*water)

  N[,1] <- rpois(M, lambda)
  for(t in 1:(T-1)) {

```

```

      S[,t] <- rbinom(M, N[,t], omega[,t])
      G[,t] <- rpois(M, gamma[,t])
      N[,t+1] <- S[,t] + G[,t]
    }

    y[] <- rbinom(M*T, N, det)

  return(list(N=N, y=y,
             covs=data.frame(site=site,decrease=decrease,bell=bell,time=time,water=water)))
}

```

Model fitting for 500 iterations

```

nsim2 <- 500
simout2 <- matrix(NA, nsim2, 8)
predout <- matrix(NA, nrow=nsim2, ncol=47)
Nout <- matrix(NA, nrow=nsim2, ncol=47)

colnames(simout2) <- c('lam0', 'gam0', 'gam1', 'om0', 'om1', 'p0', 'p1', 'p2')

for(i in 1:nsim2) {
  cat("sim2:", i, "\n")
  lam <- c(1)
  gam <- c(0,1)
  om <- c(0.5,1)
  p <- c(0.8,1,-1)
  M <- 14
  T <- 47

  sim2out <- sim2(lam, gam, om, p, M=M, T=T)
  y.sim2 <- sim2out$y
  covs <- sim2out$covs
  N <- sim2out$N
  cn <- colnames(covs)
  siteCovs <- covs[,grep("site", cn), drop=FALSE]
  yearlySiteCovs <- list(decrease = covs[,grep("decrease", cn)],
                        bell = covs[,grep("bell", cn)])
  obsCovs <- list(time = covs[,grep("time", cn)],
                  water = covs[,grep("water", cn)])
  umf2 <- unmarkedFramePCO(y = y.sim2, siteCovs=siteCovs,
                           yearlySiteCovs=yearlySiteCovs,
                           obsCovs=obsCovs, numPrimary=T)

  m2 <- pcountOpen(~1, ~bell, ~decrease, ~time+water, umf2, K=50,
                   control=list(trace=TRUE, REPORT=1))
}

```

```
e <- coef(m2)
simout2[i, ] <- e
cat(" mle =", e, "\n")
```

Predict abundance

```
re <- ranef(m2)
preds <- bup(re)
model <- colSums(preds)
predout[i, ] <- model

true <- colSums(N)
Nout[i, ] <- true

}
```

Scenario 3.

```
# Omega fluctuates as the function of a stochastic environmental variable ('random').
# Gamma peaks in the middle of the survey period.
# Same model was used to generate data and estimate parameters.
```

Simulating data

```
sim2 <- function(lam=c(1), gam=c(0,1), om=c(0.5,1), p=c(0.8,1,-1), M=14, T=47)

{
  y <- matrix(NA, M, T)
  gamma <- matrix(NA, M, T)
  omega <- matrix(NA, M, T)
  det <- matrix(NA, M, T)

  N <- matrix(NA, M, T)
  S <- matrix(NA, M, T-1)
  G <- matrix(NA, M, T-1)

  site <- matrix(rnorm(M))
  decrease <- matrix(dnorm(seq(from=0, to=4, length.out=47), 0, 7),
                     nrow=14,ncol=47,byrow=TRUE)
  bell <- matrix(dnorm(seq(from=0, to=4, length.out=47), 2, 0.5),
                 nrow=14,ncol=47,byrow=TRUE)
  time <- matrix(rnorm(M*T, 1), M, T)
```

```

water <- matrix(rnorm(M*T, 1), M, T)
random <- matrix(rnorm(T), nrow=14, ncol=47, byrow=TRUE)

lambda <- exp(lam[1])
gamma[] <- exp(gam[1] + gam[2]*bell)
omega[] <- plogis(om[1] + om[2]*random)
det[] <- plogis(p[1] + p[2]*time + p[3]*water)

N[,1] <- rpois(M, lambda)
for(t in 1:(T-1)) {
  S[,t] <- rbinom(M, N[,t], omega[,t])
  G[,t] <- rpois(M, gamma[,t])
  N[,t+1] <- S[,t] + G[,t]
}

y[] <- rbinom(M*T, N, det)

return(list(N=N, y=y,
  covs=data.frame(site=site,decrease=decrease,bell=bell,time=time,
    water=water,random=random)))
}

```

Model fitting for 500 iterations

```

nsim2 <- 500
simout2 <- matrix(NA, nsim2, 8)
predout <- matrix(NA, nrow=nsim2, ncol=47)
Nout <- matrix(NA, nrow=nsim2, ncol=47)

colnames(simout2) <- c('lam0', 'gam0', 'gam1', 'om0', 'om1', 'p0', 'p1', 'p2')

for(i in 1:nsim2) {
  cat("sim2:", i, "\n")
  lam <- c(1)
  gam <- c(0,1)
  om <- c(0.5, 1)
  p <- c(0.8,1,-1)
  M <- 14
  T <- 47

  sim2out <- sim2(lam, gam, om, p, M=M, T=T)
  y.sim2 <- sim2out$y
  covs <- sim2out$covs
  N <- sim2out$N
  cn <- colnames(covs)
}

```



```

siteCovs <- covs[,grep("site", cn), drop=FALSE]
yearlySiteCovs <- list(decrease = covs[,grep("decrease", cn)],
                      bell = covs[,grep("bell", cn)]),
                      random = covs[,grep("random", cn)])
obsCovs <- list(time = covs[,grep("time", cn)],
               water = covs[,grep("water", cn)])
umf2 <- unmarkedFramePCO(y = y.sim2, siteCovs=siteCovs,
                       yearlySiteCovs=yearlySiteCovs,
                       obsCovs=obsCovs, numPrimary=T)

m2 <- pcountOpen(~1, ~bell, ~random, ~time+water, umf2, K=50,
               control=list(trace=TRUE, REPORT=1))

e <- coef(m2)
simout2[i, ] <- e
cat(" mle =", e, "\n")

```

Predict abundance

```

re <- ranef(m2)
preds <- bup(re)
model <- colSums(preds)
predout[i, ] <- model

true <- colSums(N)
Nout[i, ] <- true

}

```

Scenario 4.

```

# Gamma peaks in the middle of the survey period.
# In data-generating model, omega decreases throughout the survey period.
# Fitting model mis-specifies omega as ‘constant’.

```

Simulating data

```

sim2 <- function(lam=c(1), gam=c(0,1), om=c(0.5,1), p=c(0.8,1,-1), M=14, T=47)
{
  y <- matrix(NA, M, T)
  gamma <- matrix(NA, M, T)
  omega <- matrix(NA, M, T)
  det <- matrix(NA, M, T)

```

```

N <- matrix(NA, M, T)
S <- matrix(NA, M, T-1)
G <- matrix(NA, M, T-1)

site <- matrix(rnorm(M))
decrease <- matrix(dnorm(seq(from=0, to=4, length.out=47), 0, 7),
                    nrow=14, ncol=47, byrow=TRUE)
bell <- matrix(dnorm(seq(from=0, to=4, length.out=47), 2, 0.5),
               nrow=14, ncol=47, byrow=TRUE)
time <- matrix(rnorm(M*T, 1), M, T)
water <- matrix(rnorm(M*T, 1), M, T)

lambda <- exp(lam[1])
gamma[] <- exp(gam[1] + gam[2]*bell)
omega[] <- plogis(om[1] + om[2]*decrease)
det[] <- plogis(p[1] + p[2]*time + p[3]*water)

N[,1] <- rpois(M, lambda)
for(t in 1:(T-1)) {
  S[,t] <- rbinom(M, N[,t], omega[,t])
  G[,t] <- rpois(M, gamma[,t])
  N[,t+1] <- S[,t] + G[,t]
}

y[] <- rbinom(M*T, N, det)

return(list(N=N, y=y,
            covs=data.frame(site=site, decrease=decrease, bell=bell, time=time, water=water)))
}

# Model fitting for 500 iterations

nsim2 <- 500
simout2 <- matrix(NA, nsim2, 7)
predout <- matrix(NA, nrow=nsim2, ncol=47)
Nout <- matrix(NA, nrow=nsim2, ncol=47)

colnames(simout2) <- c('lam0', 'gam0', 'gam1', 'om0', 'p0', 'p1', 'p2')

for(i in 1:nsim2) {
  cat("sim2:", i, "\n")
  lam <- c(1)
  gam <- c(0, 1)
  om <- c(0.5, 1)
  p <- c(0.8, 1, -1)

```

```

M <- 14
T <- 47

sim2out <- sim2(lam, gam, om, p, M=M, T=T)
y.sim2 <- sim2out$y
covs <- sim2out$covs
N <- sim2out$N
cn <- colnames(covs)
siteCovs <- covs[,grep("site", cn), drop=FALSE]
yearlySiteCovs <- list(decrease = covs[,grep("decrease", cn)],
                      bell = covs[,grep("bell", cn)])
obsCovs <- list(time = covs[,grep("time", cn)],
               water = covs[,grep("water", cn)])
umf2 <- unmarkedFramePCO(y = y.sim2, siteCovs=siteCovs,
                       yearlySiteCovs=yearlySiteCovs,
                       obsCovs=obsCovs, numPrimary=T)

m2 <- pcountOpen(~1, ~bell, ~1, ~time+water, umf2, K=50,
                control=list(trace=TRUE, REPORT=1))

e <- coef(m2)
simout2[i, ] <- e
cat(" mle =", e, "\n")

# Predict abundance

re <- ranef(m2)
preds <- bup(re)
model <- colSums(preds)
predout[i, ] <- model

true <- colSums(N)
Nout[i, ] <- true

}

```

Scenario 5.

```

# Omega decreases throughout the survey period.
# In data-generating model, gamma peaks in the middle of the survey period.
# Fitting model mis-specifies gamma as 'constant'.

```

Simulating data

```

sim2 <- function(lam=c(1), gam=c(0,1), om=c(0.5,1), p=c(0.8,1,-1), M=14, T=47)
{
  y <- matrix(NA, M, T)
  gamma <- matrix(NA, M, T)
  omega <- matrix(NA, M, T)
  det <- matrix(NA, M, T)

  N <- matrix(NA, M, T)
  S <- matrix(NA, M, T-1)
  G <- matrix(NA, M, T-1)

  site <- matrix(rnorm(M))
  decrease <- matrix(dnorm(seq(from=0, to=4, length.out=47), 0, 7),
                     nrow=14, ncol=47, byrow=TRUE)
  bell <- matrix(dnorm(seq(from=0, to=4, length.out=47), 2, 0.5),
                 nrow=14, ncol=47, byrow=TRUE)
  time <- matrix(rnorm(M*T, 1), M, T)
  water <- matrix(rnorm(M*T, 1), M, T)

  lambda <- exp(lam[1])
  gamma[] <- exp(gam[1] + gam[2]*bell)
  omega[] <- plogis(om[1] + om[2]*decrease)
  det[] <- plogis(p[1] + p[2]*time + p[3]*water)

  N[,1] <- rpois(M, lambda)
  for(t in 1:(T-1)) {
    S[,t] <- rbinom(M, N[,t], omega[,t])
    G[,t] <- rpois(M, gamma[,t])
    N[,t+1] <- S[,t] + G[,t]
  }

  y[] <- rbinom(M*T, N, det)

  return(list(N=N, y=y,
             covs=data.frame(site=site, decrease=decrease, bell=bell, time=time, water=water)))
}

```

Model fitting for 500 iterations

```

nsim2 <- 500
simout2 <- matrix(NA, nsim2, 7)
predout <- matrix(NA, nrow=nsim2, ncol=47)
Nout <- matrix(NA, nrow=nsim2, ncol=47)

```

```

colnames(simout2) <- c('lam0', 'gam0', 'om0', 'om1', 'p0', 'p1', 'p2')

for(i in 1:nsim2) {
  cat("sim2:", i, "\n")
  lam <- c(1)
  gam <- c(0, 1)
  om <- c(0.5, 1)
  p <- c(0.8, 1, -1)
  M <- 14
  T <- 47

  sim2out <- sim2(lam, gam, om, p, M=M, T=T)
  y.sim2 <- sim2out$y
  covs <- sim2out$covs
  N <- sim2out$N
  cn <- colnames(covs)
  siteCovs <- covs[,grep("site", cn), drop=FALSE]
  yearlySiteCovs <- list(decrease = covs[,grep("decrease", cn)],
                        bell = covs[,grep("bell", cn)])
  obsCovs <- list(time = covs[,grep("time", cn)],
                  water = covs[,grep("water", cn)])
  umf2 <- unmarkedFramePCO(y = y.sim2, siteCovs=siteCovs,
                          yearlySiteCovs=yearlySiteCovs,
                          obsCovs=obsCovs, numPrimary=T)

  m2 <- pcountOpen(~1, ~1, ~decrease, ~time+water, umf2, K=50,
                  control=list(trace=TRUE, REPORT=1))

  e <- coef(m2)
  simout2[i, ] <- e
  cat(" mle =", e, "\n")

  # Predict abundance

  re <- ranef(m2)
  preds <- bup(re)
  model <- colSums(preds)
  predout[i, ] <- model

  true <- colSums(N)
  Nout[i, ] <- true

}

```

Scenario 6.

```
# Omega decreases throughout the survey period.
# Gamma peaks in the middle of the survey period.
# In data-generating model, mean detection probability (p) is 0.8 and varies with two
# environmental variables (time and water).
# Fitting model mis-specifies p as 'constant'.
```

Simulating data

```
sim2 <- function(lam=c(1), gam=c(0,1), om=c(0.5,1), p=c(0.8,1,-1), M=14, T=47)
{
  y <- matrix(NA, M, T)
  gamma <- matrix(NA, M, T)
  omega <- matrix(NA, M, T)
  det <- matrix(NA, M, T)

  N <- matrix(NA, M, T)
  S <- matrix(NA, M, T-1)
  G <- matrix(NA, M, T-1)

  site <- matrix(rnorm(M))
  decrease <- matrix(dnorm(seq(from=0, to=4, length.out=47), 0, 7),
    nrow=14, ncol=47, byrow=TRUE)
  bell <- matrix(dnorm(seq(from=0, to=4, length.out=47), 2, 0.5),
    nrow=14, ncol=47, byrow=TRUE)
  time <- matrix(rnorm(M*T, 1), M, T)
  water <- matrix(rnorm(M*T, 1), M, T)

  lambda <- exp(lam[1])
  gamma[] <- exp(gam[1] + gam[2]*bell)
  omega[] <- plogis(om[1] + om[2]*decrease)
  det[] <- plogis(p[1] + p[2]*time + p[3]*water)

  N[,1] <- rpois(M, lambda)
  for(t in 1:(T-1)) {
    S[,t] <- rbinom(M, N[,t], omega[,t])
    G[,t] <- rpois(M, gamma[,t])
    N[,t+1] <- S[,t] + G[,t]
  }

  y[] <- rbinom(M*T, N, det)

  return(list(N=N, y=y,
    covs=data.frame(site=site, decrease=decrease, bell=bell, time=time, water=water)))
}
```

```
}
```

Model fitting for 500 iterations

```
nsim2 <- 500
simout2 <- matrix(NA, nsim2, 6)
predout <- matrix(NA, nrow=nsim2, ncol=47)
Nout <- matrix(NA, nrow=nsim2, ncol=47)

colnames(simout2) <- c('lam0', 'gam0', 'gam1', 'om0', 'om1', 'p0')

for(i in 1:nsim2) {
  cat("sim2:", i, "\n")
  lam <- c(1)
  gam <- c(0, 1)
  om <- c(0.5, 1)
  p <- c(0.8, 1, -1)
  M <- 14
  T <- 47

  sim2out <- sim2(lam, gam, om, p, M=M, T=T)
  y.sim2 <- sim2out$y
  covs <- sim2out$covs
  N <- sim2out$N
  cn <- colnames(covs)
  siteCovs <- covs[,grep("site", cn), drop=FALSE]
  yearlySiteCovs <- list(decrease = covs[,grep("decrease", cn)],
                        bell = covs[,grep("bell", cn)])
  obsCovs <- list(time = covs[,grep("time", cn)],
                  water = covs[,grep("water", cn)])
  umf2 <- unmarkedFramePCO(y = y.sim2, siteCovs=siteCovs,
                          yearlySiteCovs=yearlySiteCovs,
                          obsCovs=obsCovs, numPrimary=T)

  m2 <- pcountOpen(~1, ~bell, ~decrease, ~1, umf2, K=50,
                  control=list(trace=TRUE, REPORT=1))

  e <- coef(m2)
  simout2[i, ] <- e
  cat(" mle =", e, "\n")
}
```

Predict abundance

```
re <- ranef(m2)
preds <- bup(re)
```

```

model <- colSums(preds)
predout[i, ] <- model

true <- colSums(N)
Nout[i, ] <- true

}

```

Double peak immigration

Generating a dummy variable that presents double peaks with slightly lower peak following the
initial peak.

```

mu1 <- log(1)
mu2 <- log(50)
sig1 <- log(3)
sig2 <- log(3)
cpct <- 0.45

bimodalDistFunc <- function (n,cpct, mu1, mu2, sig1, sig2) {
  y0 <- rlnorm(n,mean=mu1, sd = sig1)
  y1 <- rlnorm(n,mean=mu2, sd = sig2)

  flag <- rbinom(n,size=1,prob=cpct)
  y <- y0*(1 - flag) + y1*flag
}

bimodalData <- bimodalDistFunc(n=10000,cpct,mu1,mu2, sig1,sig2)

x <- hist(log(bimodalData), breaks=c(seq(-5,9,length.out=48)))

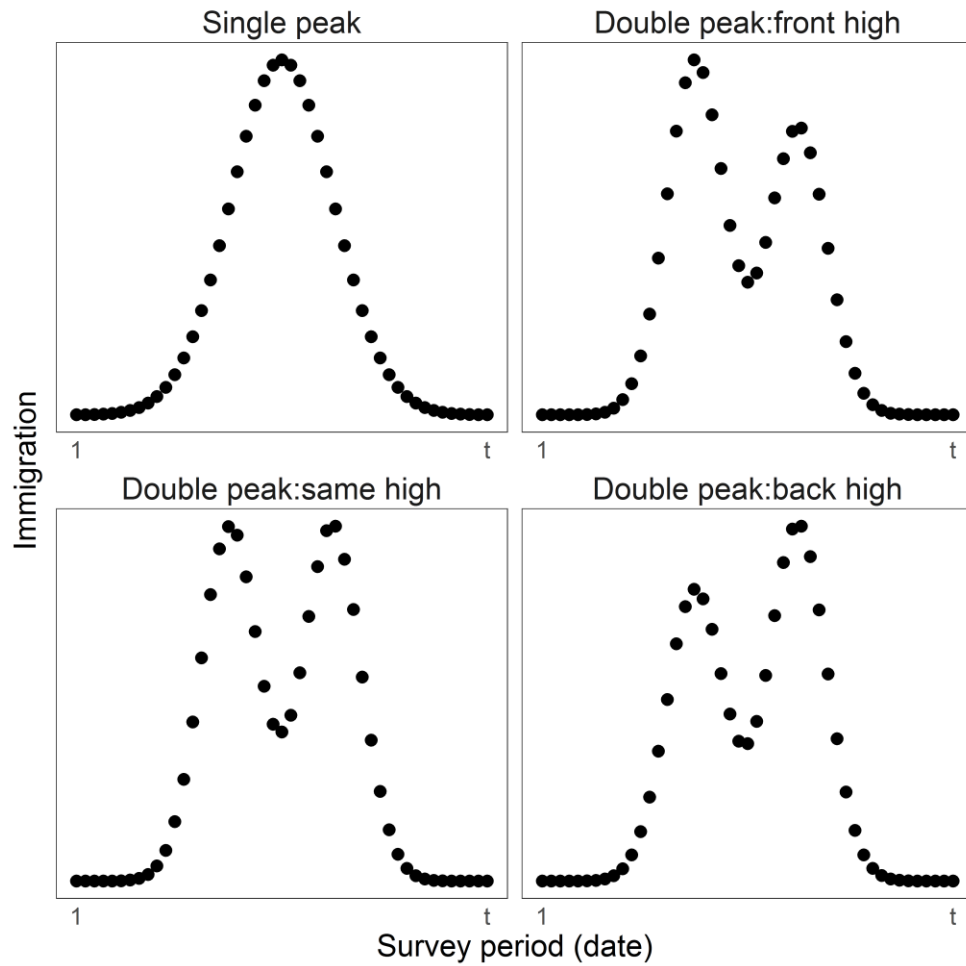
bipeak <- matrix(rep(x$density,6),nrow=6,ncol=47,byrow=TRUE)

```

For simulating with double peak immigration models, repeat the above R scripts for ‘single
peak immigration’ Scenario 1–6 after replacing ‘bell’ with ‘bipeak’.

S2. Hypothesized temporal structure of immigration tested for ruddy turnstone data

We created four dummy variables each representing one of the following hypothesized immigration structures. The total number of survey periods differed by year, and ' t ' was 44, 46, and 50 for 1997–1999 respectively.



S3. Simulation results with bimodal immigration models

Figure S3-1. Boxplots showing the distribution of parameter estimates from generalized N-mixture models with the probability that an individual survived and stayed on site between survey days (ϕ , ‘persistence’) being A) constant, B) decreasing with date, or C) fluctuating with a stochastic environmental factor. Each scenario was tested with the detection probability of 80% (‘High P’) or 30% (‘Low P’) and was simulated 500 times. Red lines indicate the true value of each parameter. Blank panels are when there was no covariate effect to estimate. The same model we used to generate the dataset were fit to the dataset. Immigration was set to show bimodal peaks, and detection probability covaried with two environmental variables. Environmental covariate for persistence represented ‘date’ (B) or a ‘stochastic environmental factor’ (C).

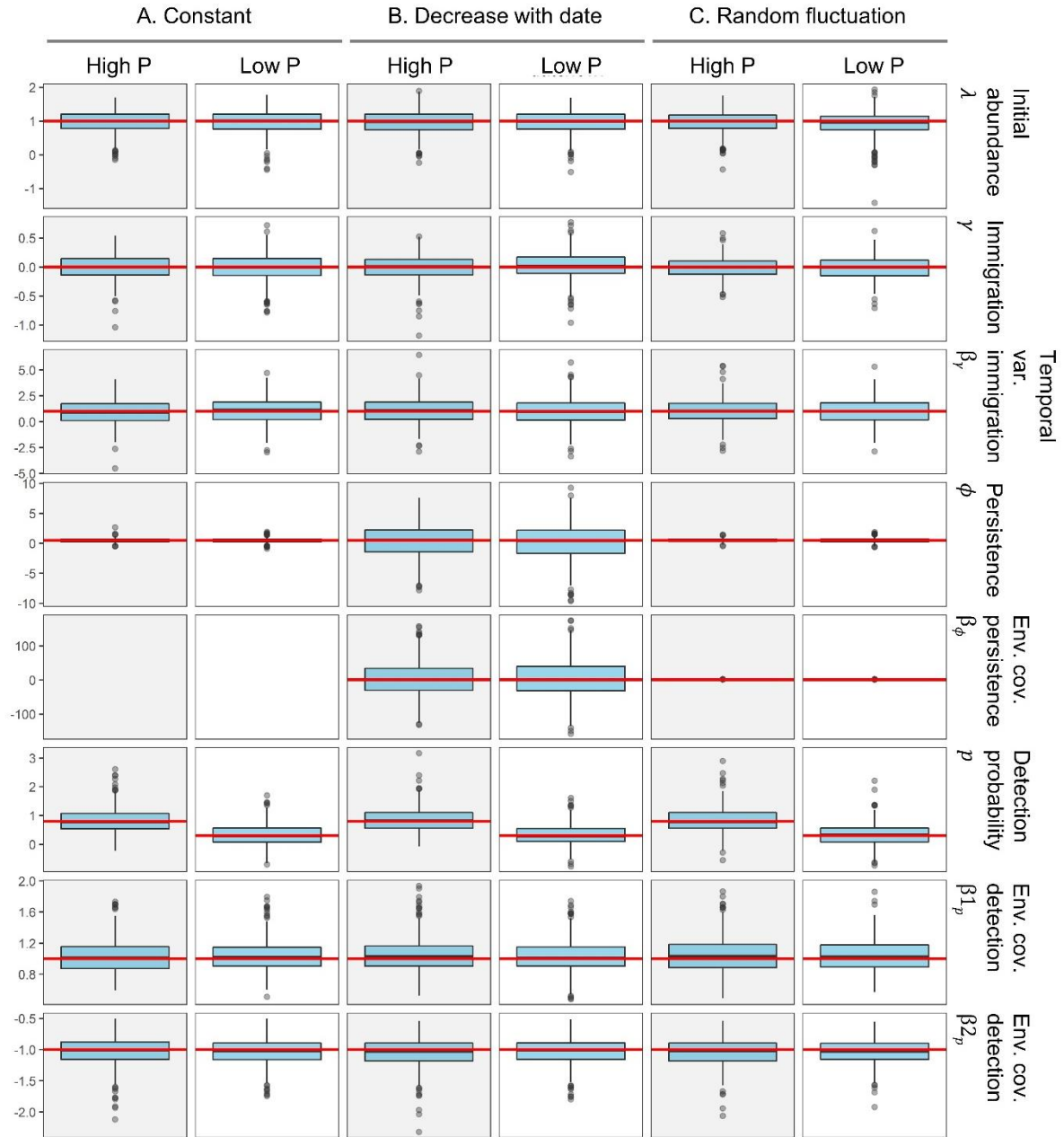


Figure S3-2. Comparison of true abundance (red) to predicted abundance (blue) from three scenarios of persistence. Models used for analyses and models used to generate dataset were matched. Mean abundances are shown with error bars.

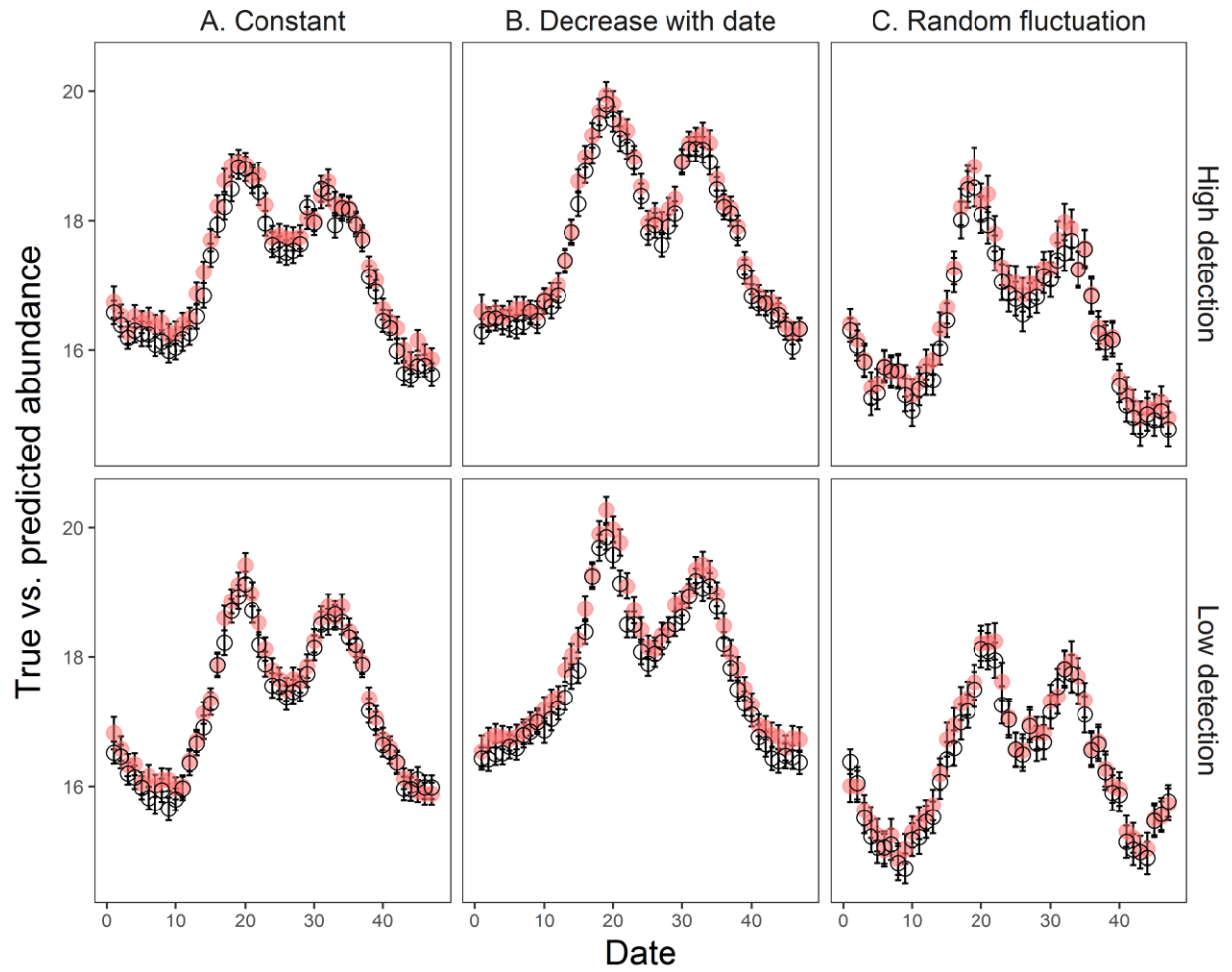


Figure S3-3. Boxplots showing the distribution of parameter estimates from generalized N-mixture model where A) persistence, B) immigration, or C) detection probability was mis-specified in the model as ‘constant’. Each scenario was tested with the detection probability of 80% (‘High P’) or 30% (‘Low P’) and was simulated 500 times. Dataset for each scenario was simulated so that persistence decreased over time, immigration showed bimodal peaks, and detection probability covaried with two environmental variables. Red lines indicate the true value of each parameter. Blank panels are when there was no covariate effect to estimate.

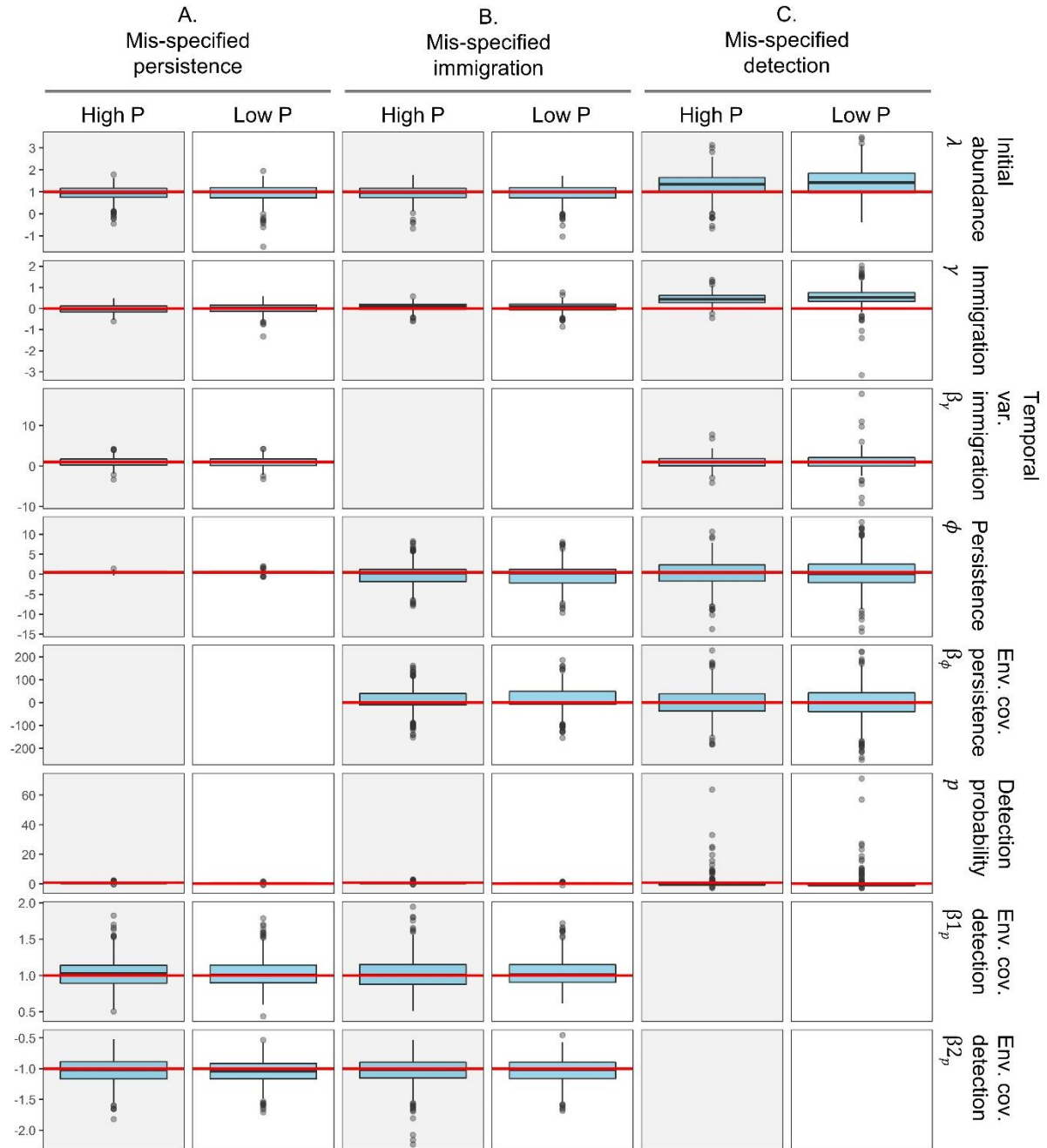
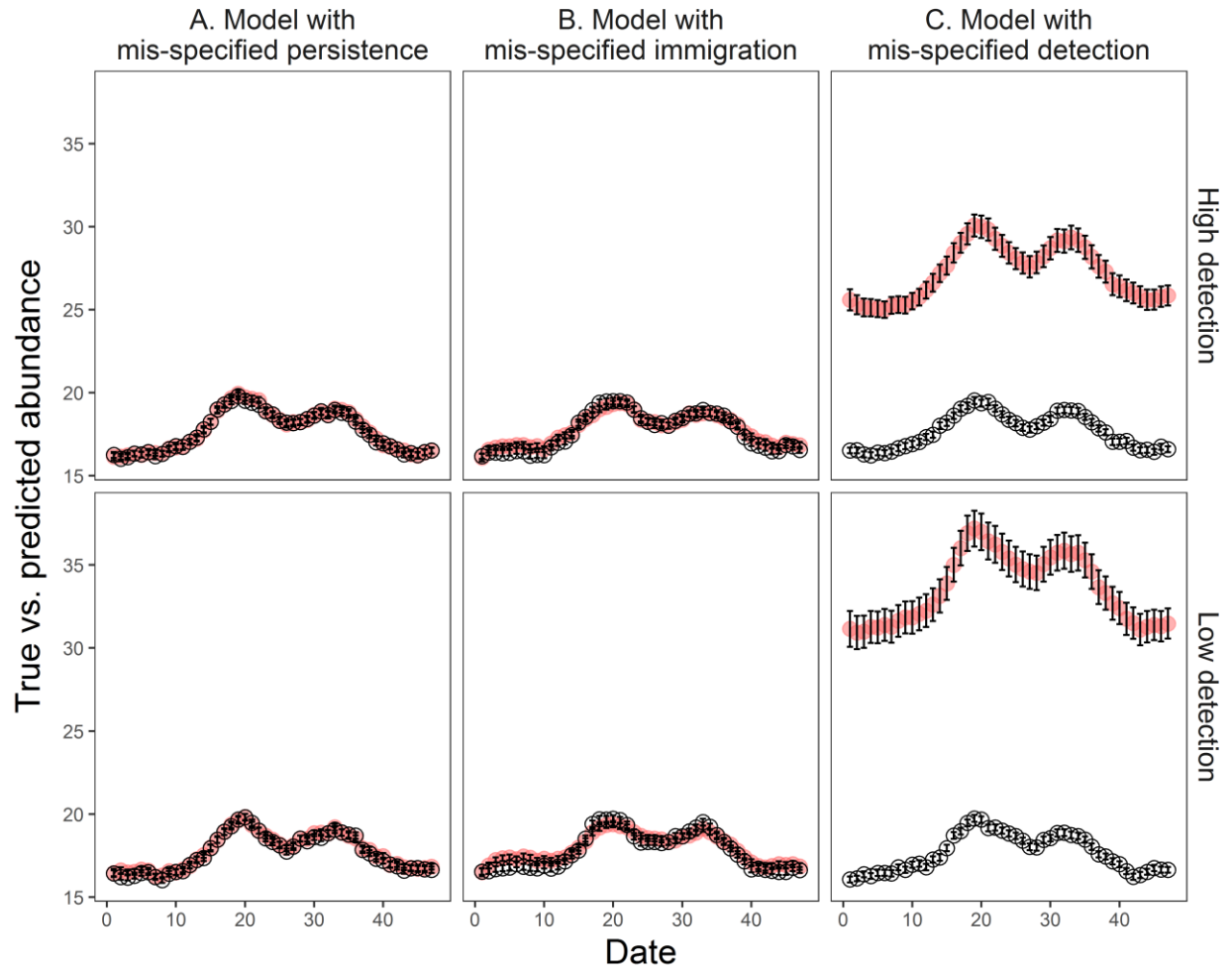


Figure S3-4. Comparison of true abundance (red) to predicted abundance (blue) from three scenarios where A) persistence, B) immigration, or C) detection probability was mis-specified in the model as ‘constant’. Models used for analyses and models used to generate dataset were not matched. Mean abundances are shown with error bars.



S4. Supplementary results

Table S4-1. Covariate coefficient estimates and 95% confidence intervals from the best model on estimating daily abundance of ruddy turnstones during spring migration of 1997–1999 on Westhampton Island, New York. ‘-’ indicates covariates not included in the best model.

Parameter	Covariate	1997				1998				1999			
		Estimate	SE	Lower	Upper	Estimate	SE	Lower	Upper	Estimate	SE	Lower	Upper
Detection probability	Time of day	-	-	-	-	-0.15	0.08	-0.32	0.01	-0.50	0.08	-0.65	-0.35
	Tide height	0.43	0.05	0.33	0.53	0.50	0.07	0.38	0.63	0.44	0.06	0.32	0.56
	Precipitation	0.18	0.03	0.12	0.24	1.25	0.42	0.43	2.08	-0.24	0.07	-0.37	-0.11
	Wind speed	-	-	-	-	-0.16	0.10	-0.35	0.03	-0.13	0.05	-0.23	-0.03
	Wind direction	-	-	-	-	0.45	0.09	0.29	0.62	0.24	0.07	0.11	0.37
	Time×Tide	-	-	-	-	0.04	0.08	-0.11	0.19	-0.31	0.08	-0.46	-0.16
Immigration rate	Subsite*	-4.52	0.71	-5.92	-3.13	-4.86	0.71	-6.25	-3.47	-4.89	0.58	-6.03	-3.75
	Dates	-13.49	2.60	-18.59	-8.39	-2.83	0.49	-3.79	-1.87	-6.30	1.29	-8.82	-3.78
	Dates ²	-8.31	1.48	-11.20	-5.42	-3.45	0.44	-4.32	-2.59	-4.44	0.75	-5.90	-2.98

* Westhampton Dune was used as a reference site.

Figure S4-1. Daily observed counts (red) and predicted abundance of ruddy turnstones that is expected to have been detectable given the estimated detection probability (black) during spring migration of 1997–1999 on Westhampton Island, New York.

