

SEGUIMIENTO DE LA DIVERSIDAD BIOLÓGICA

Modelos N-mixtos

José Jiménez García-Herrera (IREC-CSIC)

Universidad de Castilla-La Mancha

Los modelos N-mixtos (Royle and Nichols, 2003) son muy similares conceptualmente a los modelos de ocupación, pero en vez de describir presencia/ausencia de una especie en un territorio, describen la abundancia. Se basan en el uso de dos procesos anidados: abundancia en un sitio (Poisson), y caso de estar (condicional), se detecta o no en los muestreos (binomial). La estima se infiere a partir de réplicas espaciales (sitios de muestreo) y temporales (ocasiones de muestreo), entre las cuales las poblaciones deben ser cerradas (sin entradas o salidas geográficas o demográficas). Los modelos N-mixtos pueden ser abiertos, al añadirles una componente temporal.

Hay que tener en cuenta varias cuestiones 1) en los conteos no puede haber duplicidades 2) los conteos deben ser independientes (atención a las especies gregarias) y 3) En algunas ocasiones, los modelos con la binomial negativa, pese a ser seleccionados por el AIC, pueden estimar valores irreales. Recomendando los trabajos de Kéry (2018) a este respecto.

Repetid los análisis que se ejecutan aquí con R (R Core Team, 2020), cambiando cada uno de vosotros el valor de *rnd* que controla la aleatoriedad de la simulación.

Los scripts que se usan aquí están extraídos del libro de Kéry, M., & Schaub, M. (2012). Bayesian population analysis using WinBUGS. A hierarchical perspective. Bayesian Population Analysis using WinBUGS. Academic Press / Elsevier. <http://doi.org/10.1016/B978-0-12-387020-9.00014-6>.

1. Modelo sin covariables

Simulación de datos

En los procesos de modelado, emplear simulaciones nos va a permitir comparar datos “perfectos” (no afectados por errores o heterogeneidad) con los resultados de los modelos usando esos datos. Por otro lado las simulaciones también, en una segunda instancia, nos van a servir

para preparar el trabajo de campo y prever el tamaño de muestra a partir de un error máximo deseado. Por último, las simulaciones son muy adecuadas para el aprendizaje. Elegimos el tamaño de muestra y preparamos la matriz para contener los datos observados.

```
> set.seed(24)                # Para repetir siempre la misma creación de datos
> M <- 150                    # Número de sitios
> J <- 4                      # Número de réplicas por sitios
> C <- matrix(NA, nrow = M, ncol = J) # matriz de datos
```

Valores de los parámetros

```
> lambda <- 14                # Abundancia esperada
> p <- 0.3                    # Probabilidad de detección (por individuo)
```

Proceso ecológico

Lo que hay. Generamos datos de abundancia de la especie objetivo.

```
> # Generamos datos de abundancia local
> N <- rpois(n = M, lambda = lambda)
```

Proceso de observación

Lo que vemos. Generamos datos de detección/no detección.

```
> # Simulamos la observación "J" veces de esos datos
> for(j in 1:J){
+   C[,j] <- rbinom(n = M, size = N, prob = p)
+ }
```

Veamos los datos creados

```
> table(N)                    # Verdadera distribución de la abundancia
N
 6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25
 2  1  5  8  7 18 17 17 13 15 15 12  4  3  4  1  5  2  1

> sum(N)                      # Tamaño total de población en los M sitios
[1] 2086

> sum(N>0)                    # Número real de sitios ocupados
```

```
[1] 150
> mean(N)                                # Verdadera abundancia media (estima de lambda)
[1] 13.90667
> # ... y lo que observamos
> table(apply(C, 1, max))                # distribución de la abundancia observada (max conteos)
 2  3  4  5  6  7  8  9 10 12
1 14 22 29 26 28 16 12  1  1
> sum(apply(C, 1, max))                  # Tamaño de la población observada en los M sitios
[1] 887
> sum(apply(C, 1, max)>0)                 # Número de sitios observados como ocupados
[1] 150
> mean(apply(C, 1, max))                 # Media observada de la abundancia
[1] 5.913333
> head(cbind(N=N, count1=C[,1], count2=C[,2], count3=C[,3]))
      N count1 count2 count3
[1,] 11      5      0      5
[2,] 14      4      6      2
[3,] 19      9      6      8
[4,] 16      7      4      6
[5,] 11      1      5      5
[6,] 15      5      7      4
```

Modelo en una aproximación MLE (usando unmarked)

```
> library(unmarked)                    # Cargamos la librería
> umf <- unmarkedFramePCount(y = C)     # preparamos datos
> summary(umf)

unmarkedFrame Object

150 sites
```

Maximum number of observations per site: 4

Mean number of observations per site: 4

Sites with at least one detection: 150

Tabulation of y observations:

0	1	2	3	4	5	6	7	8	9	10	12
13	41	76	112	113	102	57	42	28	14	1	1

```
> (fm1 <- pcount(~1 ~1, data = umf, K=150)) # ejecutamos el modelo
```

Call:

```
pcount(formula = ~1 ~ 1, data = umf, K = 150)
```

Abundance:

Estimate	SE	z	P(> z)
2.74	0.146	18.7	2.1e-78

Detection:

Estimate	SE	z	P(> z)
-0.997	0.199	-5	5.7e-07

AIC: 2505.011

```
> backTransform(fm1, "state") # y lo obtenemos a escala natural
```

Backtransformed linear combination(s) of Abundance estimate(s)

Estimate	SE	LinComb	(Intercept)
15.4	2.26	2.74	1

Transformation: exp

```
> backTransform(fm1, "det")
```

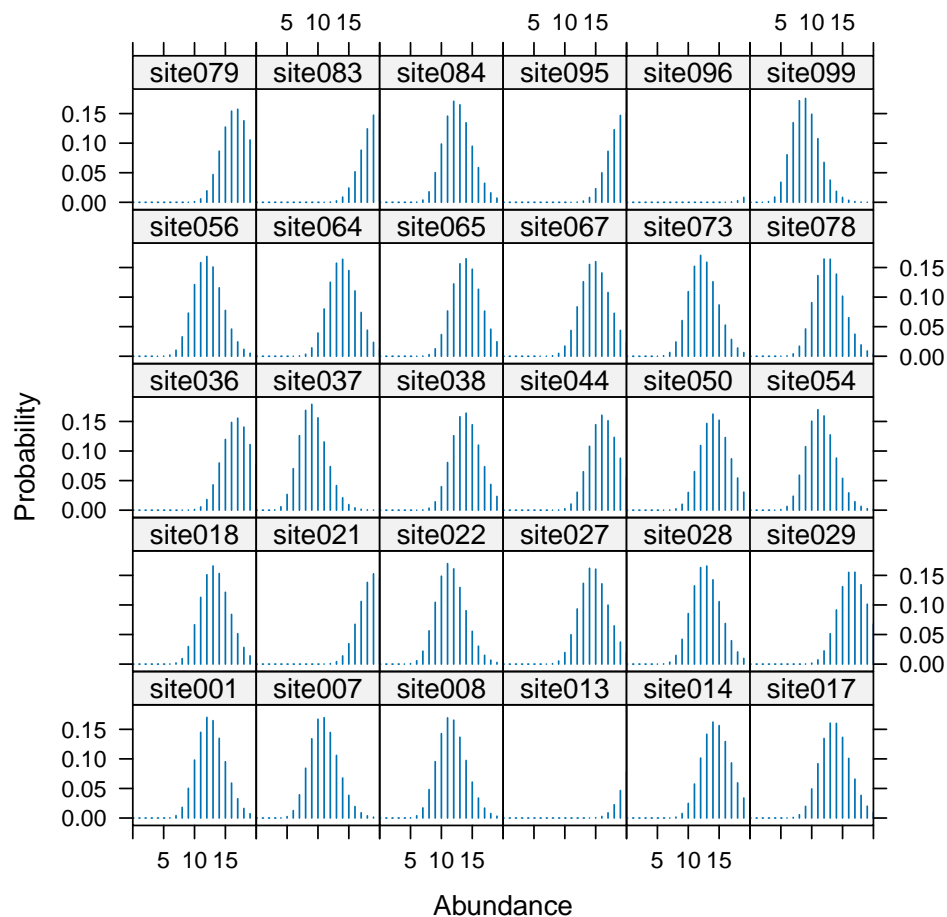
Backtransformed linear combination(s) of Detection estimate(s)

Estimate	SE	LinComb	(Intercept)
0.27	0.0392	-0.997	1

Transformation: logistic

... y podemos ver el resultado para cada sitio utilizando métodos bayesianos empíricos

```
> re<-ranef(fm1)
> plot(re, xlim = c(0,20))[sort(sample(1:100, 30))]
```



Obtenemos los intervalos del 95 % y contrastamos con los 10 primeros datos creados

```
> # Para obtener los intervalos de confianza usamos bootstrapping, ya que
> # ranef puede subestimar la varianza de la distribución posterior al no
> # tener en cuenta la incertidumbre de los hiperparámetros (lambda y psi).
> ppd <- posteriorSamples(re, nsims=10000)
> res<-data.frame(Media=round(apply(ppd$samples, 1, mean), 2),
```

```
+           SD=round(apply(ppd@samples,1,sd),2),
+           lower=apply(ppd@samples,1,quantile,0.025),
+           upper=apply(ppd@samples,1,quantile,0.975))
> # Contrastamos con los datos creados en la simulación (N)
> head(cbind(N,res))
```

	N	Media	SD	lower	upper
1	11	12.73	2.32	9	18
2	14	14.19	2.40	10	19
3	19	20.95	2.66	16	27
4	16	17.77	2.56	13	23
5	11	13.67	2.40	9	19
6	15	17.27	2.55	13	23

Modelo en una aproximación bayesiana con JAGS

Modelo

La estructura del modelo N-mixto es muy sencilla, y similar al modelo de ocupación, pero en vez de una estructura Binomial-Binomial, es Poisson-Binomial. El modelo lo ajustaremos con JAGS (Plummer, 2003).

```
> cat(file = "model.jags",
+ "
+ model {
+
+   # Información a priori
+   lambda ~ dgamma(0.005, 0.005) # Información previa para lambda
+   # lambda ~ dunif(0, 10)      # Otra posibilidad
+   p ~ dunif(0, 1)
+
+   # Probabilidad
+   # Modelo ecológico para la abundancia
+   for (i in 1:R) {
+     N[i] ~ dpois(lambda)
+   }
+   # Modelo de observación para los conteos replicados
+   for (j in 1:T) {
+     y[i,j] ~ dbin(p, N[i])
+   }
+ }
```

```
+     } # j
+   } # i
+ }
+
+ ")
```

Preparamos datos

```
> y<-C # Reutilizamos los mismos datos de antes
> win.data <- list(y = y, R = nrow(y), T = ncol(y))
```

Valores de inicio

```
> Nst <- apply(y, 1, max, na.rm=TRUE) + 1 # Importante!!
> inits <- function() list(N = Nst)
```

Parámetros a monitorizar

```
> params <- c("lambda", "p")
```

Configuración MCMC

```
> ni <- 25000
> nt <- 10
> nb <- 5000
> nc <- 3
```

Ejecución del modelo

Llamamos a JAGS desde R (R Core Team, 2020) utilizando jagsUI (Kellner, 2020).

```
> library(jagsUI)
> out <- jags(win.data, inits, params, "model.jags", n.chains = nc,
+           n.thin = nt, n.iter = ni, n.burnin = nb, parallel=TRUE)
```

Processing function input.....

Done.

Beginning parallel processing using 3 cores. Console output will be suppressed.

Parallel processing completed.

Calculating statistics.....

Done.

```
> #Resumimos resultados
> print(out, dig = 2)
```

JAGS output for model 'model.jags', generated by jagsUI.
Estimates based on 3 chains of 25000 iterations,
adaptation = 100 iterations (sufficient),
burn-in = 5000 iterations and thin rate = 10,
yielding 6000 total samples from the joint posterior.
MCMC ran in parallel for 0.433 minutes at time 2024-02-08 20:08:56.715644.

	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
lambda	15.80	2.37	12.10	15.49	21.54	FALSE	1	1.02	199
p	0.27	0.04	0.19	0.27	0.34	FALSE	1	1.01	262
deviance	2362.29	20.31	2323.48	2361.65	2403.99	FALSE	1	1.00	733

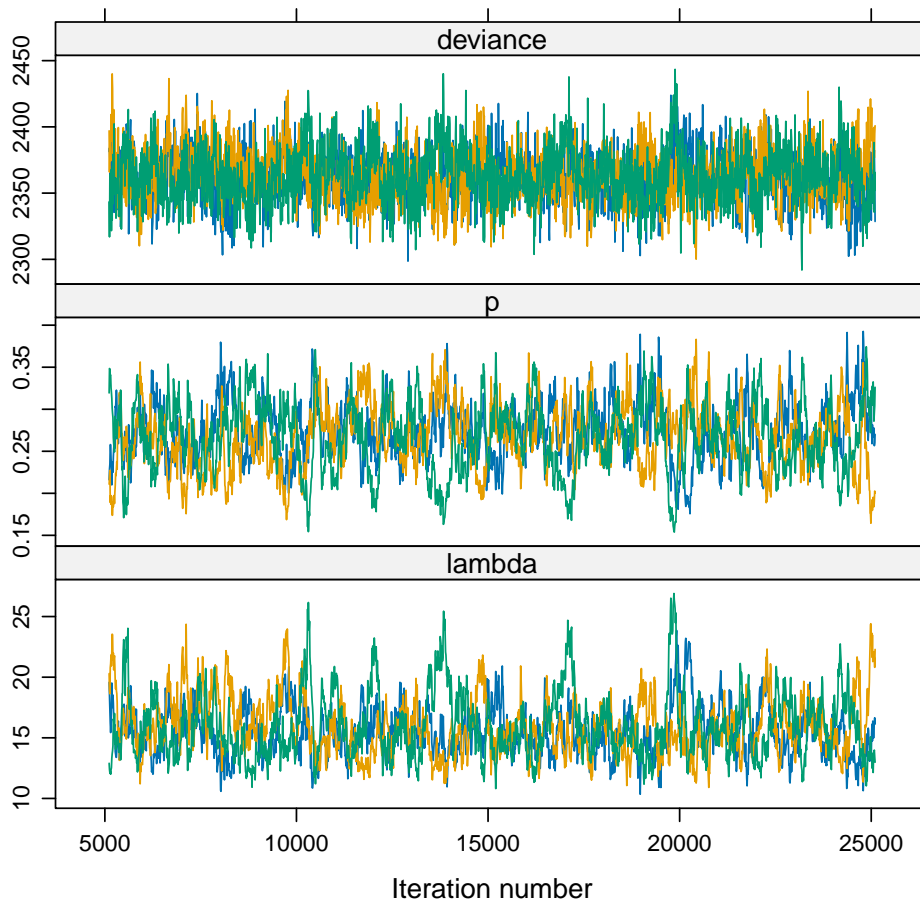
Successful convergence based on Rhat values (all < 1.1).
Rhat is the potential scale reduction factor (at convergence, Rhat=1).
For each parameter, n.eff is a crude measure of effective sample size.

overlap0 checks if 0 falls in the parameter's 95% credible interval.
f is the proportion of the posterior with the same sign as the mean;
i.e., our confidence that the parameter is positive or negative.

DIC info: (pD = var(deviance)/2)
pD = 205.8 and DIC = 2568.04
DIC is an estimate of expected predictive error (lower is better).

Vemos la convergencia de las cadenas de Markov

```
> library(lattice)
> xyplot(out$samples)
```

2. REFERENCIAS

- Fiske, I. J., & Chandler, R. B. (2011). **unmarked**: An R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, 43(10), 1–23. doi:10.1002/wics.10.
- Kellner, K. (2020). jagsUI: A Wrapper Around ‘rjags’ to Streamline ‘JAGS’ Analyses. R package version 1.5.1.9100. Retrieved from <https://github.com/kenkellner/jagsUI>.
- Kéry, M., & Schaub, M. (2012). *Bayesian population analysis using WinBUGS. A hierarchical perspective*. Academic Press / Elsevier. doi:10.1016/B978-0-12-387020-9.00014-6.
- Kéry, M. (2018). Identifiability in N-mixture models: a large-scale screening test with bird data. *Ecology*, 99(2), 281–288. doi:10.1002/ecy.2093
- Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. 3rd International Workshop on Distributed Statistical Computing (DSC 2003). Vienna, Austria. Retrieved from <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/>
- R Core Team. (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from <https://www.r-project.org/>.
- Royle, J. A., & Nichols, J. D. (2003). Estimating abundance from repeated presence-absence data or point counts. *Ecology*, 84, 777–790. doi:10.1890/0012-9658(2003)084[0777: EAFRPA]2.0.CO;2