Martin, T.G., Wintle, B.A., Rhodes, J.R., Kuhnert, P.M., Low Choy, S., Field, S.A., Tyre, A.J., & Possingham, H.P. (2005) Zero tolerance ecology: improving ecological inference by modelling the source of zero observations. *Ecology Letters*, 8, 1235-1246.

Appendix 1.

WinBUGS code for the zero-inflated Poisson mixture model. Note grazeL is a binary variable indicating a low level of grazing, grazeM indicates a moderate level of grazing and grazeH indicates a high level of grazing.

Note WinBUGS is a freely available piece of software available for download at http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml

Zero-inflated Poisson Mixture Model for Bird/Grazing data

```
MODEL
{
# Pr(Y=0|x,z)
        for(i in 1:Nz){
                zeros[i] <- 0
                zeros[i] ~ dpois(mu[i])
                mu[i] <- -log(1-p[i] + p[i]*exp(-lambda[i]))
                logit(p[i]) <- alpha
                log(lambda[i]) <- beta1.low*grazeL[i] + beta1.mod*grazeM[i] +
                beta1.high*grazeH[i]
        }

# Pr(Y>0|x,z)
        for(i in (Nz+1):N){
                zeros[i] <- 0
                zeros[i] ~ dpois(mu[i])
                mu[i] <- -(log(p[i])-lambda[i] + y[i-Nz]*log(lambda[i])-logfact(y[i-Nz]))
                logit(p[i]) <- alpha
                log(lambda[i]) <- beta1.low*grazeL[i] + beta1.mod*grazeM[i] +
                beta1.high*grazeH[i]
        }

# Vague Priors for model coefficients
                alpha ~ dnorm(0.0,0.0001)
                beta1.low ~ dnorm(0.0,0.0001)
                beta1.mod ~ dnorm(0.0,0.0001)
                beta1.high ~ dnorm(0.0,0.0001)
```

Appendix 2.
WinBUGS code for the generalised zero-inflated binomial model used when accounting for excess zeros generated by false zeros.

Zero-inflated binomial model for Mt Lofty Bird data. Code presented is that for the full model including all covariates influencing the probability of site occupancy (p[i])

```
MODEL
{
for (i in 1:n) {  #  this first loop is transforming the independent variables and calculating their mean and standard deviation so that they may be standardized later
        logConnectivity[i] <- log(Connectivity[i])
        }
        meanconnectivity <- mean(logConnectivity[1:n])
        sdconnectivity <- sd(logConnectivity[1:n])

for(i in 1:155 ) { # this is the modelling loop, looping over i sites.
        stdarea[i] <- (log(area[i]) - 2.5495)/0.7678 #standardizing 'area'
        stdconnectivity[i] <- (log(Connectivity[i]) - meanconnectivity)/sdconnectivity
        #standardizing 'connectivity'
        obs[i] ~ dbin(q[i],3) # obs[i] is the observed presences after 3 visits (the data)
        Y[i] ~ dbern(p[i]) # Y[i] = 1 if species present and 0 if not
        q[i] <- Y[i] * alpha[1] # q[i] is detection probability (=0 if species not present)
        logit(p[i]) <- alpha[2] + beta[1]*stdarea[i] + beta[2]*stdconnectivity[i]+
        beta[3]*Hab[i] # the logistic regression model describing landscape and
        habitat influences on the probability of occupancy
        }

# Vague Priors for model coefficients
        alpha[1] ~ dbeta( 2,3 ) # detection probability (Pr(observed|presence))
        alpha[2] ~ dnorm( 0.0,0.0001)
        beta[1] ~ dnorm( 0.0,0.0001)
        beta[2] ~ dnorm( 0.0,0.0001)
        beta[3] ~ dnorm( 0.0,0.0001)
}
```