

Recolonisation of rabbit warrens following coordinated ripping programs in Victoria, south-eastern Australia

D. S. L. Ramsey^{A,B,G}, S. R. McPhee^{C,D}, D. M. Forsyth^A, I. G. Stuart^E, M. P. Scroggie^A, M. Lindeman^A and J. Matthews^F

^AArthur Rylah Institute for Environmental Research, Department of Environment and Primary Industries, 123 Brown Street, Heidelberg, Vic. 3084, Australia.

^BSchool of Earth and Environmental Sciences, University of Adelaide, Adelaide, SA 5005, Australia.

^CDepartment of Environment and Primary Industries, 600 Sneydes Road, Werribee, Vic. 3030, Australia.

^DAgricultural Technical Services P/L, 48 Warooka Road, Yorketown, SA 5576, Australia.

^EKingfisher Research P/L, 177 Progress Road, Eltham, Vic. 3095, Australia.

^FInvasive Plants and Animals Operations Branch, Biosecurity Victoria, Department of Environment and Primary Industries, Hamilton Centre, Mount Napier Road, Hamilton, Vic. 3300, Australia.

^GCorresponding author. Email: david.ramsey@depi.vic.gov.au

Appendix S1. JAGS code used for the discrete-time Weibull survival analysis with Gibbs variable selection (GVS).

```

# data step to populate the full warren x time event history matrix y[,]
# from event summaries. The time[] variable contains the last time of
# event or censoring for each warren and d[] is the censoring indicator
# (1 - event; 0 - censoring)
data {
  for(i in 1:N) {
    y[i,time[i]] <- d[i]
    for (j in 1:time[i]-1) {y[i,j] <- 0}
  }
}

Model {
# Warren level model. Discrete-time Weibull
  for (i in 1:N) {
    for (j in 1:time[i]) {y[i,j] ~ dbern(q[i,j])}
    cloglog(q[i,j]) <- a0[sites[i]] + Kappa * log(j) +
      beta[1]*NW[i] +
      beta[2]*AE[i] +
      beta[3]*IE[i] +
      beta[4]*K[i] +
      beta[5]*TH[i] +
      beta[6]*U[i]
  }
}

# Site level model
  for(j in 1:Nsite) {
    a0[j] ~ dnorm(eta[j], tau.0)
    eta[j]<- beta0 + beta[7]*RA[j] + beta[8]*MT[j] + beta[9]*MC[j]
  }

# Gibbs variable selection. gam[] is the indicator for each covariate

  for(j in 1:p) {
    gam[j] ~ dbern(0.5)
    betaG[j] ~ dnorm(0, tau.b)
    beta[j]<- gam[j]*betaG[j]
  }

  beta0 ~ dnorm(0, 0.001)
  tau.0<- 1/(sig.0*sig.0)
  sig.0 ~ dunif(0,100)
  tau.b<- 1/(sig.b*sig.b)
  sig.b ~ dunif(0,20)
  Kappa ~ dnorm(0, 0.01)

# Model indices
  for (j in 1:p){mindex[j] <- pow(2,j)}
  modl <- 1 + inprod(gam, mindex)
}

```