Combining capture-recapture data and pedigree information to assess heritability of demographic parameters in the wild. by Julien Papaix, Sarah Cubaynes, Mathieu Buoro, Anne Charmantier, Philippe Perret and Olivier Gimenez.

Supporting Information 1: R script and BUGS code to implement the capture-recapture animal model

1 R script to run the capture-recapture animal model from OpenBUGS

Here, we provide the R script which calls OpenBUGS to fit the capture-recapture animal model to the blue tit data. The pedigree and capture-recapture data are available from http://www.cefe.cnrs.fr/biom/zips/CRAMcode.zip or from the last author’s webpage. Note that, due to confidentiality reasons, we provide only a random sub-sample of the whole data set here.

# R script to run the CRAM model (see CRAM.bug) using OpenBUGS
# J. Papaix, O. Gimenez
# April 2010

### DATA

# 1. a matrix "cr.ped" with
# col. 1 = individual id
# col. 2 = father id (NA if unknown)
# col. 3 = mother id (NA if unknown)
# col. 4-last = detections
# (NA if before first capture, 1 if individual is seen, 0 otherwise)

# 2. a vector "first" with occasion of first capture for each individual
############ DATA MANIPULATION

x <- as.matrix(cr.ped[,4:32])  # Blue tit data, 29 occ. => 32 col. in cr.ped
ntot <- dim(x)[1]  # total number of individuals
nyear <- dim(x)[2]  # number of capture occasions
nasc <- 327  # number of individuals without recorded parents
ndesc <- ntot-nasc  # number of individuals with recorded parents
ad <- seq((nasc+1),ntot)  # pointer to obs. number for an ind. with recorded parents
nd <- seq(1,ndesc)  # pointer to obs. number for an ind. without recorded parents
ped <- as.data.frame(cr.ped[,1:3])  # get the pedigree
dimnames(ped)[[2]] <- c("id","father","mother")  # name columns of the pedigree
SID <- ped[,2]  # sire id
DID <- ped[,3]  # dam id

# load GeneticsPed package to work out the pedigree
# (http://www.bioconductor.org/packages/2.0/bioc/html/GeneticsPed.html)
library(GeneticsPed)
ped <- as.Pedigree(ped)
sq_D <- sqrt(diag(mendelianSamplingD(ped)))

############ RUN OPENBUGS

# data
data <- list(x=x[(nasc+1):ntot,],first=first[(nasc+1):ntot],nyear=nyear,
sq_D=sq_D,nasc=nasc,ndesc=ndesc,ad=ad,nd=nd,SID=SID,DID=DID)

# initial values (2 chains)
z <- as.matrix((x[(nasc+1):ntot,] == 0) | (x[(nasc+1):ntot,] == 1))+0
init1 <- list(ptemp=0.5,beta=-0.8,z=z,sigma_gen=0.1,sigma_ngen=0.1,sigma_env=.4)
init2 <- list(ptemp=0.8,beta=0.4,z=z,sigma_gen=1,sigma_ngen=1,sigma_env=.2)
inits <- list(init1,init2)
# load R2WinBUGS package to run OpenBUGS from R
library(R2WinBUGS)

# parameters to be monitored
parameters <- c("ptemp","beta","sigma_env","sigma_gen","sigma_ngen","herit")

# run OpenBUGS
res.sim <- bugs(data,inits,parameters,"CRAM.bug",n.chains=2,n.iter=15000,
               n.burn=5000,n.thin=10,program="OpenBUGS")

# display results
round(res.sim$summary,3)

2 BUGS code implementing the capture-recapture animal model

# CRAM model in BUGS language
# J. Papaix, O. Gimenez
# April 2010

# Freely inspired from:

# Damgaard, L. H. 2007. Technical note: How to use WinBUGS to
# draw inferences in animal models.

# Royle, J. A. 2008. Modeling individual effects in the
# Cormack-Jolly-Seber model: a state-space formulation.
# Biometrics, 64:364-370.
#

### LIKELIHOOD

# state-space formulation

for(i in 1:ndesc){
  z[i,first[i]] ~ dbern(1) # initial state
  for(j in (first[i]+1):nyear){
    mu2[i,j]<-PHI[i,j-1]*z[i,j-1]
    z[i,j] ~ dbern(mu2[i,j]) # state equation
    mu1[i,j]<-P[i,j-1]*z[i,j]
    x[i,j] ~ dbern(mu1[i,j]) # observation equation
  }
}

for(i in 1:ndesc){ # individuals with recorded parents
  # genetic effect
  a[ad[i]] <- v[ad[i]] * sigma_gen
  v[ad[i]] <- gam[ad[i]] * sq_D[ad[i]] + (v[SID[ad[i]]] + v[DID[ad[i]]])/2
  gam[ad[i]] ~ dnorm(0,1)
  na[ad[i]] <- vna[ad[i]] * sigma_nngen
  vna[ad[i]] ~ dnorm(0,1)

  for(t in 1:(nyear-1)){
    logit(P[i,t])<- lp[t] # logit link for time-dep detection probabilities
    # mean survival + non-genetic + genetic + year effect
    muphi[i,t] <- beta + na[ad[i]] + a[ad[i]] + eta[t]
  }
}
PHI[i,t] <- phi(muphi[i,t]) # probit link on survival

for (i in 1:nasc){ # individuals without recorded parents
  v[nd[i]] <- gam[nd[i]]* sq_D[nd[i]]
  gam[nd[i]] ~ dnorm(0,1)
}

for(j in 1:(nyear-1)){
  lp[j]<-log(ptemp/(1-ptemp)) # make the detection prob. constant through time
  eta[j] ~ dnorm(0,tau_env) # temporal random effect on survival
}

######### PRIORS

# prior on the detection probability
ptemp ~ dunif(0,1)

# prior on the mean survival probability
beta ~ dnorm(0,1.0E-1)

# prior on the SD of the non-genetic effect
sigmangen ~ dunif(0,10)
var_nngen <- sigmangen * sigmangen

# prior on the SD of the additive genetic effect
sigmagen ~ dunif(0,10)
var_gen <- sigmagen * sigmagen
# prior on the SD of the temporal effect

```r
sigma_env ~ dunif(0, 10)
```

tau_env <- 1 / var_env

```r
var_env <- sigma_env * sigma_env
```

# monitor heritability

```r
herit <- var_gen / (var_gen + var_ngen + var_env + 1)
```

```r
}
```