Supporting Information

**Appendix S2**

**Disentangling the effects of date, individual and territory quality on the seasonal decline in fitness**

Tomas Pärt, Jonas Knape, Matthew Low, Meit Öberg & Debora Arlt

Swedish University of Agricultural Sciences, Department of Ecology, Box 7044, SE-75007 Uppsala, Sweden .

Corresponding author: Tomas Pärt, [tomas.part@slu.se](mailto:tomas.part@slu.se)

**Contents**

BUGS-model for covariance components of laying data

and number of recruits produced p. 2-3

R-script for running the JAGS model of wheatear covariance components p. 4-5

######################################################################################

## BUGS-model for covariance components of laying data and number of recruits produced

######################################################################################

model {

for (i in 1:nBreed) {

mu[i, 1] <- mu0[1] + cAge[1] \* ageF[i] + cFLH[1]\*(FLH[i] - 1) + cRoof[1]\*nesttype[i] + sY[1]\*alphaY[yr[i], 1] + sS[1]\*alphaS[siteID[i], 1] + sF[1]\*alphaF[femaleID[i], 1]

mu[i, 2] <- mu0[2] + cAge[2] \* ageF[i] + cFLH[2]\*(FLH[i] - 1) + cRoof[2]\*nesttype[i] + sY[2]\*alphaY[yr[i], 2] + sS[2]\*alphaS[siteID[i], 2] + sF[2]\*alphaF[femaleID[i], 2]

# residual random effects

eggdat[i] ~ dnorm(mu[i, 1], pow(sE[1], -2))

logmu2[i] ~ dnorm(mu[i,2] + rhoE\*sE[2]/sE[1]\*(eggdat[i] - mu[i, 1]), pow(sE[2], -2)/(1-pow(rhoE, 2)))

# Poisson response for recruits

mu2[i] <- exp(logmu2[i])

recruits[i] ~ dpois(mu2[i])

}

# inidividual random effects

for (female in 1:nFemale) {

alphaF[female, 1:2] ~ dmnorm(zeroV, tauF)

}

# site effects

for (site in 1:nSite) {

alphaS[site, 1:2] ~ dmnorm(zeroV, tauS)

}

# year effects

for (year in 1:nYear) {

alphaY[year, 1:2] ~ dmnorm(zeroV, tauY)

}

#### Define precision matrices

## Females

sigmaF[1, 1] <- 1

sigmaF[2, 2] <- 1

sigmaF[1, 2] <- rhoF

sigmaF[2, 1] <- sigmaF[1, 2]

tauF <- inverse(sigmaF)

## Sites

sigmaS[1, 1] <- 1

sigmaS[2, 2] <- 1

sigmaS[1, 2] <- rhoS

sigmaS[2, 1] <- sigmaS[1, 2]

tauS <- inverse(sigmaS)

## Sites

sigmaY[1, 1] <- 1

sigmaY[2, 2] <- 1

sigmaY[1, 2] <- rhoY

sigmaY[2, 1] <- sigmaY[1, 2]

tauY <- inverse(sigmaY)

#### Priors

sE[1] ~ dunif(0, 1000)

sE[2] ~ dunif(0, 5)

rhoE ~ dunif(-1, 1)

sF[1] ~ dunif(0, 1000)

sF[2] ~ dunif(0, 5)

rhoF ~ dunif(-1, 1)

sS[1] ~ dunif(0, 1000)

sS[2] ~ dunif(0, 5)

rhoS ~ dunif(-1, 1)

sY[1] ~ dunif(0, 1000)

sY[2] ~ dunif(0, 5)

rhoY ~ dunif(-1, 1)

mu0[1] ~ dnorm(0, 1E-4)

mu0[2] ~ dnorm(0, 1E-2)

cAge[1] ~ dnorm(0, 1E-4)

cAge[2] ~ dnorm(0, 1E-2)

cFLH[1] ~ dnorm(0, 1E-4)

cFLH[2] ~ dnorm(0, 1E-2)

cRoof[1] ~ dnorm(0, 1E-4)

cRoof[2] ~ dnorm(0, 1E-2)

#### Utility variables

zeroV[1] <- 0

zeroV[2] <- 0

}

#########################################################################

## R-script for running the JAGS model of wheatear covariance components

#########################################################################

library(rjags)

## Number of MCMC iterations for each chain. For final analyses this script was run 8 times on separate cores.

nSamp = 3750

############################

## Load data

############################

data = read.csv("wheatearData.csv", row.names = 1)

jagsData = list(eggdat = data$eggdat,

siteID = data$siteID,

yr = data$yr,

FLH = data$FLH,

nesttype = data$nesttype,

nBreed = dim(data)[1],

nSite = max(data$siteID),

nYear = max(data$yr),

recruits = data$recruits,

femaleID = data$femaleID,

ageF = data$ageF,

nFemale = max(data$femaleID))

############################

## Set variables to monitor

############################

monitor = c("cFLH", "cAge","cRoof",

"sS", "alphaS", "rhoS",

"sY", "alphaY", "rhoY",

"sF", "alphaF", "rhoF",

"sE", "rhoE")

inits = list(mu0 = c(14,0.5), cAge = c(0,0), cFLH = c(0, 0), cRoof = c(0,0), sS = c(1, 0.1), sY = c(1, 0.1), rhoS = 0, rhoY = 0,

sF = c(1, 0.1), rhoF = 0, rhoE = 0, sE = c(2,.1))

############################

## Run model

############################

model = jags.model("covarianceModel.bugs",

data = jagsData,

inits = inits, n.adapt = 10000)

samp = coda.samples(model, variable.names = monitor, n.iter=nSamp\*500, thin = 500)