Supplementary text

Process error for survival and catches

Survival process is defined by assuming a schooling behavior and schools of equal size but with no temporal variation. This situation corresponds with the concentration of mean survival probability (ψ) approaching infinity in case 3, Table 1 of Mäntyniemi et al. (2015), as follows:

$$N_{t+1}|N_t, p_t, \psi, sc \sim Betabin(N_t, p_t, \eta, (1-p_t)\eta),$$

with $\eta = \frac{(N_t - sc)}{(sc-1)}$, where sc is the group size.

For computation we used the beta approximation to the beta-binomial distribution from Appendix 3 in Mäntyniemi et al. (2015),

$$s_t \sim Beta(p_t\eta_t^*, (1-p_t)\eta_t^*),$$

with $\eta_t^* = \frac{N(1+\eta)}{(N+\eta)} - 1 = \frac{N_t}{sc} - 1$. To avoid potentially negative values in computation, we assume $\eta_t^* = \frac{N_t}{sc}$.

The catch process error is defined in the same way as the survival process, by replacing p_t by q_t and N_t by d_t, but not using the beta approximation to beta-binomial distribution.

JAGS code

```
gpdm<-
“
model{
#########################################################################
 population dynamics

  for(t in 1:Y){                    # T time steps
  # Change in total population size, scaled to initial population size ni

  n[t+1]<-s[t]*n[t]+R[t+1]+0.006
  oR[t]<-R[t]*ni # recruitment in original scale
  on[t]<-n[t]*ni                    # total abundance in original scale

  # survival process
  s[t]~dbeta(as[t],bs[t])T(0.01,0.99)      # surviving proportion

  as[t]<-p[t]*eta_star[t]+0.1
  bs[t]<-(1-p[t])*eta_star[t]+0.1
  eta_star[t]<-n[t]*ni*qEta

  p[t]<-sum(phiSs[t,1:LC])       # survival probability

   #surv[t]<-exp(-Z[t])
   Z[t]<-Mconstant+mueF


  # Dynamics of the size distribution

  phiG[t,1:LC]<-phi[t,1:LC]%*%G[1:LC,1:LC]       # phi: proportion of fish in size class
  phiS[t,1:LC]~ddirch(aphiS[t,1:LC]+0.1)

  for(k in 1:LC){
  aphiS[t,k]<-phiSs[t,k]*eta_star[t]
  }
                                # phiS lenght distribution after growth
  Rratio[t+1]<-R[t+1]/(n[t+1]+0.000001)
   for(k in 1:LC){

   phiSs[t,k]<-exp(-Z[t])*phiG[t,k]                  # Expected survival
  # lenght distribution after recruitment:
   phi[t+1,k]<-(1-Rratio[t+1])*phiS[t,k]+Rratio[t+1]*phiR[k]
                                 # initial state for the next time step
   deltas[t,k]<-(1-exp(-Z[t]))*phiG[t,k]
   delta[t,k]<-deltas[t,k]/(1-p[t])             # length distribution of dead fish
```

```
    gammas[t,k]<-delta[t,k]*mueF/Z[t]  #*Fsel[k]
    gamma[t,k]<-gammas[t,k]/q[t]              # length distribution in the catch
    }
   q[t]<-sum(gammas[t,1:LC])  # proportion of caught fish from all the dead fish
   }

qEta~dbeta(1,1)T(0.001,0.99) #cluster size as a proportion of the total population size


## Dirichlet prior for the size composition in the beginning of the first time step
phi[1,1:LC]~ddirch(aphi[1:LC]+0.2)#aphi[1:LC]#MUPHI
for(k in 1:LC){
aphi[k]<-mu_phi[k]*eta_phi
}

# Mortality rates and initial abundance: multivariate normal
#prior for a linear combination of transformed parameters
# -> less correlation between simulated parameters and block updating
#using M-H alg. -> better mixing
Mconstant<-exp(LM)  # Natural mortality rate
LM<-par[1]+muM
Fconstant<-exp(LF)  #Fishing mortality rate
LF<-par[2]+muF
n[1]<-1
ni<-exp(lni)  # Initial population size
lni<-par[3]+muN
R[1]<-0       # no recruits in the first year: they are already
#included in the initial size distribution
R[2]<-0.0000000000001
R[3]<-0.0000000000001
R[4]<-0.0000000000001
R[5]<-0.0000000000001
R[6]<-0.0000000000001
R[7]<-0.0000000000001
R[8]<-0.0000000000001
R[9]<-0.0000000000001
R[10]<-0.0000000000001


Ja[1]<-0.0000000000001
Jb[1]<-0.0000000000001
Jc[1]<-0.0000000000001
Ja[2]<-0.0000000000001
Jb[2]<-0.0000000000001
Jc[2]<-0.0000000000001
Ja[3]<-0.0000000000001
Jb[3]<-0.0000000000001
Jc[3]<-0.0000000000001
Ja[4]<-0.0000000000001
Ja[5]<-0.0000000000001
Ja[6]<-0.0000000000001
Ja[7]<-0.0000000000001


Jb[4]<-0.0000000000001
Jb[5]<-0.0000000000001
Jb[6]<-0.0000000000001
Jb[7]<-0.0000000000001
Jb[8]<-0.0000000000001
```

```
Jc[5]<-0.0000000000001#dlnorm(log(500), 5.48)T(0,);#5.48=1/log(1+100/500)
Jc[4]<-0.0000000000001#dlnorm(log(500), 5.48)T(0,);
Jc[6]<-0.0000000000001
Jc[7]<-0.0000000000001
Jc[8]<-0.0000000000001
Jc[9]<-0.0000000000001



Jamean[1]<-0.0000000000001
Jamean[2]<-0.0000000000001
Jamean[3]<-0.0000000000001
Jamean[4]<-0.0000000000001
Jamean[5]<-0.0000000000001
Jamean[6]<-0.0000000000001
Jamean[7]<-0.0000000000001


Jbmean[1]<-0.0000000000001
Jbmean[2]<-0.0000000000001
Jbmean[3]<-0.0000000000001
Jbmean[4]<-0.0000000000001
Jbmean[5]<-0.0000000000001
Jbmean[6]<-0.0000000000001
Jbmean[7]<-0.0000000000001
Jbmean[8]<-0.0000000000001


Jcmean[1]<-0.0000000000001
Jcmean[2]<-0.0000000000001
Jcmean[3]<-0.0000000000001
Jcmean[4]<-0.0000000000001
Jcmean[5]<-0.0000000000001
Jcmean[6]<-0.0000000000001
Jcmean[7]<-0.0000000000001
Jcmean[8]<-0.0000000000001
Jcmean[9]<-0.0000000000001

Rmean[1]<-0.0000000000001
Rmean[2]<-0.0000000000001
Rmean[3]<-0.0000000000001
Rmean[4]<-0.0000000000001
Rmean[5]<-0.0000000000001
Rmean[6]<-0.0000000000001
Rmean[7]<-0.0000000000001
Rmean[8]<-0.0000000000001
Rmean[9]<-0.0000000000001
Rmean[10]<-0.0000000000001

for (t in (Y+2):(Y+5)){
 n[t]~dnorm(0,1)
}

for (t in 5:222){
Cu[t]<-Eggs[t]*exp(-li*(W[t+2,4]+W[t+2,3]+W[t+2,2]+W[t+2,1]+W[t+1,4]+W[t+1,3]
+W[t+1,2]+W[t+1,1]))
TCF[t]<-step(T[t+1,1]-T[t,4]-0.25)*step(T[t+1,1]-16)
TCS[t]<-step(T[t+1,2]-T[t+1,1]-0.25)*step(T[t+1,2]-16);
TCT[t]<-step(T[t+1,3]-T[t+1,2]-0.25)*step(T[t+1,3]-16);
TCL[t]<-step(T[t+1,4]-T[t+1,3]-0.25)*step(T[t+1,4]-16);
```

```
#First juveniles
Jamean[t+3]<-TCF[t]*Cu[t]+TCS[t]*Cu[t]*exp(-li*(W[t+3,1]-W[t+1,1])+
TCT[t]*Cu[t]*exp(-li*(W[t+3,2]+W[t+3,1]-W[t+1,1]-W[t+1,2])-0.01)+TCL[t]*Cu[t]
*exp(-li*(W[t+3,3]+W[t+3,2]+W[t+3,1]-W[t+1,1]-W[t+1,2]-W[t+1,3])));


Jbmean[t+4]<- Ja[t+3]*r*dnorm((log(Disch[t+4])-4.6052),0,1);
#-179.23, Second juveniles, 179.23 is the mean of Discharges
Jcmean[t+5]<- Jb[t+4]*r*dnorm((log(Disch[t+5])-4.6052),0,1);
#]-179.23 Third juveniles
Rmean[t+6]<- Jc[t+5]*r*dnorm((log(Disch[t+6])-4.6052),0,1);

Ja[t+3]~dnorm(Jamean[t+3], 0.00000000003)T(0,);#Jamean[t+3]
Jb[t+4]~dnorm(Jbmean[t+4], 0.00000025)T(0,Ja[t+3]);#B(3,1) is 0
Jc[t+5]~dnorm(Jcmean[t+5], 0.00000025)T(0,Jb[t+4]);
preR[t+6]~dnorm(Rmean[t+6], 0.00000025)T(0,Jc[t+5]);

ER[t+6]<-log(preR[t+6])-0.5/TR+eR[t+6]
#1/TR # tauer: precision matrix for the recruitment deviations
R[t+6]<-exp(ER[t+6])+0.0000001
 eR[t+6]~dnorm(0,TR)    # recruitment residuals

}

CVR~dunif(0,3)
TR<-1/log(CVR*CVR+1)

par[1]~dnorm(mue[1],1/s2M)T(,0.823)
par[2]~dnorm(mue[2],1/s2F)T(-0.589,)
#exp(par[2]+muF(log(0.09)))>0.05, par[2]>log(0.555)=-0.589
par[3]~dnorm(mue[3],1/s2N)

mue[1]<-0
mue[2]<-0
mue[3]<-0

#########################################################################
# Von Bertalanffy Growth model: expected length from size class k


   Gk<-Gconstant                          # growth rate, fixed for now
   Lsigma<-sqrt(Linfsigma*Linfsigma*(1-exp(-Gk*2)))
   # Standard deviation of growths over a  time step
   # Linfsigma: standard deviation of lengths of old fish
   # In time, the length distribution of a cohort converges
   # to have mean Linf and SD Linfsigma

    for(k in 1:LC){
    Eg[k]<-(Linf-length[k])*(1-exp(-Gk))+length[k]
    }


## Scaled and shifted multivariate logit-normal priors for growth parameters:
#multivariate normal prior for a linear combination of transformed parameters
# -> less correlation between simulated parameters and block updating using
#M-H alg. -> better mixing
Linf<-minLinf+(maxLinf-minLinf)*pLinf
logit(pLinf)<-par3[1] #dnorm(20,0.1)T(15,25) #<-exp(par3[1])
minLinf<-18
maxLinf<-20
```

```
Linfsigma<-minsdLinf+(maxsdLinf-minsdLinf)*psdLinf
logit(psdLinf)<-par3[3]
minsdLinf<-0.1
maxsdLinf<-3
Gconstant<-mink+(maxk-mink)*pk

logit(pk)<-par3[2]
mink<-0.05 #0.1
maxk<-0.08 #0.3

par3[1:3]~dmnorm(mu3[1:3],tau3[1:3,1:3])  # multivariate normal

mu3[1]<-0
mu3[2]<-0
mu3[3]<-0

tau3<-inverse(cov3)
cov3[1,1]<-1
cov3[1,2]<-0
cov3[1,3]<-0

cov3[2,1]<-cov3[1,2]
cov3[2,2]<-1
cov3[2,3]<-0

cov3[3,1]<-cov3[1,3]
cov3[3,2]<-cov3[2,3]
cov3[3,3]<-1

#########################################################################
# Growth matrix: normally distibuted growths from each length class.
#Negative growths are allowed to counter balance
# the too fast growths from small sizes. As a result, the size distribution is kept
#plausible at the population level
# and the tendency to sink all fish into highest length class becomes avoided.
# Of course, this is not an appropriate model for individuals

# I[j]: lower bound of length class k
# l[k]: midpoint of length class k

   for( k in 1:LC){
     for( j in 1:LC){
      G[k,j]<-(phi((I[j+1]-Eg[k])/Lsigma)-
      phi((I[j]-Eg[k])/Lsigma))/(phi((I[LC+1]-Eg[k])/Lsigma)-phi((I[1]-Eg[k])/Lsigma))
     }
   }


#########################################################################
# weight-length relationship
# parameters may vary in time, fixed for now
 # for(t in 1:Y){
   # for( k in 1:LC){
     # log(w[t,k])<-logaw[t]+bw[t]*log(length[k])
     # }
   # logaw[t]<-logawConstant
    #bw[t]<-bwConstant
#  }
for( k in 1:LC){
     log(w[k])<-logawConstant+bwConstant*log(length[k])
```

```
    }


# Length weight parameters: fixed dummy values for now

logawConstant<--5.84304454 #ln(0.0029)
bwConstant<-3.3438
#########################################################################
# Eggs produced by a length class
 for(t in 1:Y){
SSB[t]<-sum(ssb[t,1:LC])/100       # Calculating SSB for output, not used in the model
BIOM[t]<-sum(biom[t,1:LC])
   for(k in 1:LC){
    eggs[t,k]<-mat[k]*fecConstant*sexr*w[k]
                                    ssb[t,k]<-mat[k]*w[k]*phiG[t,k]*on[t]
    # Spawning stock per length. Not used in the model
                                       biom[t,k]<-w[k]*on[t]*phiG[t,k]
   }
 }

#########################################################################
# total number of eggs in time t
 for(t in 1:Y){
  Eggs[t]<-inprod(eggs[t,1:LC],phi[t,1:LC])*n[t]
 }
#########################################################################
# Maturity of a length class

   for( k in 1:LC){
    mat[k]<-step(length[k]-matLength)
    }


#########################################################################
# fecundity of a length class
 #for(t in 1:Y){
  # for( k in 1:LC){
   #  fec[t,k]<-fecConstant
    # }
  #}
#########################################################################
# sex ratio (females) in a length class
  #for(t in 1:Y){
   #for( k  in 1:LC){
    # sex[t,k]<-0.5
   #}
  #}
sexr<-0.5
#########################################################################
# Natural mortality
 # for(t in 1:Y){
  # for( k in 1:LC){
   #  M[t,k]<-Mconstant  # this will be a function of length
   #}
  #}
#########################################################################
# Fishing mortality
 #for(t in 1:Y){
   #for( k in 1:LC){
    #F[t]<-mueF
   #}
```

```
#}
#   eF[t]~dgamma(aF,bF)
 #  reF[t]<-eF[t]/Fconstant
  #
#aF<-1/(CVF*CVF)
#bF<-1/(CVF*CVF*mueF)
mueF<-Fconstant
#CVF~dbeta(1,1)
#########################################################################
# gear selectivity
#   for(k in 1:LC){
 #  Fsel[k]<-SFsel[k]/SFsel[LC]
  #  probit(SFsel[k])<-(length[k]-mFsel)/sdFsel # selection curve
  # }
  # mFsel~dnorm(7,1)T(5.5,8.5)   # length at 50% selectivity
  # sdFsel~dunif(0.01,5)         # softness of the selection curve

#########################################################################
#recruitment process
#   for(t in 1:Y){
 #   R[t+1]<-exp(LR[t+1])
  #  LR[t+1]~dnorm(ER[t+1],TR)
   # ER[t+1]<-log(Eggs[t])+log(K)-log(K/alpha+Eggs[t])-0.5/TR
    #oR[t]<-R[t]*ni # recruitment in original scale
    #}

# Stock recruitment parameters: multivariate normal prior for a linear combination
#of transformed parameters
# -> less correlation between simulated parameters and block updating using M-H alg.
#-> better mixing
#oK<-ni*K      # oK : carrying capacity on original scale
#K<-exp(LK)    # carrying capacity relative to abundance in 1st year
#LK<-par2[1]+mu2[1]

#logit(alpha)<-Lalpha      # logit normal prior for alpha
#Lalpha<-par2[1]+mu2[1]-par2[2]-mu2[2]

#par2[1:2]~dmnorm(mue2[1:2],tau2[1:2,1:2])

#mu2[1]<-muK
#mu2[2]<-muK-mualpha

#mue2[1]<-0
#mue2[2]<-0

#tau2<-inverse(cov2)     # Covariance matrix to acoount for the transformations
#so that the original parameters
#cov2[1,1]<-s2K          # still have independent priors
#cov2[1,2]<-s2K
#cov2[2,1]<-cov2[1,2]
#cov2[2,2]<-s2K+s2alpha

#CVR~dunif(0,1) # CV of recruitment residuals
#TR<-1/log(CVR*CVR+1)

#########################################################################
# fishing process
  for(t in 1:Y){
   d1[t]<-round(n[t]*(1-s[t])*ni)      # number of dead fish
   d[t]<-step(d1[t]-c[t])*d1[t]+step(c[t]-d1[t])*c[t]
   # Beta-binomial model for catches
```

```
   ac[t]<-q[t]*eta_star[t]+0.1
   bc[t]<-(1-q[t])*eta_star[t]+0.1
 c[t]~dbetabin(ac[t],bc[t],d1[t])
  # LLC[t]<-loggam(d[t]+1)-loggam(c[t]+1)-loggam(d[t]-c[t]+1)+
  loggam(ac[t]+c[t])+loggam(d[t]+bc[t]-c[t])-loggam(ac[t]+bc[t]+d[t])+loggam(ac[t]+bc[t])
#-loggam(ac[t])-loggam(bc[t])
 #   pdummy[t]<-step(d1[t]-c[t])*exp(LLC[t])
  #  dummy[t]~dbern(pdummy[t])
   }

#########################################################################
# Observation model for length data

 # for( k in 1:LC){
  #bandw[k]<-(I[k+1]-I[k])/4   # smoothing bandwidth based on widths of length intervals
  #}
  #for( t in 1:Y){
   #for( j in 1:OLC){
    #for( k in 1:LC){
   #qlength[t,j,k]<-(phi((IO[j+1]-length[k])/bandw[k])-phi((IO[j]-length[k])/bandw[k]))
   #/(phi((IO[OLC+1]-length[k])/bandw[k])-phi((IO[1]-length[k])/bandw[k]))  #check
     # }
    #  plength[t,j]<-inprod(qlength[t,j,1:LC],gamma[t,1:LC])
   #}
 #x[t,1:OLC]~dmulti(plength[t,1:OLC],SSize[t])
  #}

 #for( t in 1:Y){
  # for( j in 1:OLC){
#x[t,1:OLC]~dmulti(plength[t,1:OLC],SSize[t])
#}}

for( j in 1:length(CPUEindex)){
CCPUE[j]~dnorm(Q*n[CPUEindex[j]]*ni,0.000001)
}

Q<-exp(logQ)
logQ~dnorm(muQ,1/s2Q)
li~dunif(0,2);
r~dnorm(0.1,10)T(0,);
M<-log(meanac1)-0.5*S2
Tac1<-1/S2
S2<-log(pow(1/invsigmac1,2)/pow(meanac1,2)+1)
meanac1<-on[71]+0.1
invsigmac1~dgamma(0.001,0.001)T(0.001,)

#AcousticJune2004<-exp(lnx2)
#lnx2~dnorm(M2,T2)
M2<-log(meanac2)-0.5*S2ac2
T2<-1/S2ac2
S2ac2<-log(pow(1/invsigmac1,2)/pow(meanac2,2)+1)
meanac2<-on[215]+0.1
AcousticJune1993~dlnorm(M,Tac1)T(0,);
AcousticJune2004~dlnorm(M2,T2)T(0,);
}
“
```