Bayesian biomass dynamic model for fisheries stock assessment

Charles T T Edwards

NIWA Ltd., Wellington, NZ

charles.edwards@niwa.co.nz

June 2014

1 Brief introduction to surplus production models

We describe the general properties of different surplus production models that have been proposed, with specific reference to the MSY reference point:

$$\phi = \frac{b^{MSY}}{K}$$

and the intrinsic rate of population growth:

$$r = \lim_{b \to 0} \frac{1}{b} \frac{db}{dt}$$

as defined in ecological theory [1].

The logistic model of population growth is traditionally attributed to Schaefer [2, 3] who introduced the concept fisheries.

```
dbdt <- function(b) r * b * (1 - b/K)
```

The Schaefer model was genearlised by Pella and Tomlinson [4] to allow $\phi \neq 0.5K$ by introducing a shape parameter p. However, the parameter referred to as r no longer has a simple biological interpretation.

```
dbdt <- function(b) r/p * b * (1 - (b/K)^p)
```

The Pella-Tomlinson model at the limit of $p \to 0$ was given by Fox [5].

```
dbdt <- function(b) r * b * (1 - log(b)/log(K))
```

The Pella-Tomlinson model was re-paramaterised by Fletcher [6] to give it more biologically meaningful parameters. However, for $n \leq 1$ this model has the anomalous property of predicting near infinite rates of surplus production per capita as abundance decreases to low levels [7].

```r
dbdt <- function(b) g * m * b/K * (1 - (b/K)^(n-1))
```

A combined Fletcher-Schaefer hybrid model was proposed by McAllister [8] to allow $\phi < K/2$ whilst maintaining an ecologically consistent interpretation of r.

```r
dbdt <- function(b) {
  bf <- b[which(b>bmsy)]
  bs <- b[which(b<=bmsy)]
  c(r * bs * (1 - bs/h),g * m * bf/K * (1 - (bf/K)^(n-1)))
}
```

This allows us to specify a model with $\phi = 0.40$, which is the default assumption for stock assessments in New Zealand.

The Beverton-Holt model is parameterised by M and $\phi < 0.5$.

```r
dbdt <- function(b) alpha * b / (1 + beta * b) - M * b
```

Given these two inputs we estimate α numerically as the solution to the relationship

$$\phi = \frac{\sqrt{\frac{\alpha}{M}} - 1}{\frac{\alpha}{M} - 1}$$

The β parameter is then obtained from

$$\beta = \frac{1}{K}\left(\frac{\alpha}{M} - 1\right)$$

This model predicts a similar replacement yield to the Fletcher-Schaefer hybrid model at $\phi = 0.4$. However the productivity per unit biomass is a different shape, and higher at low biomass levels. This difference becomes even more exagerated at lower ϕ values.

It is informative to compare all potentially useful models at $\phi = 0.4$. The Schaefer model is included as a reference.

```r
r <- 0.1
K <- 100
b <- seq(0,K,1e-3)
rlim <- function(b) dbdt(b) * (1/b)

# Schaefer model
dbdt <- function(b) r * b * (1 - b/K)
dfr <- data.frame(biomass=b,
                  production=dbdt(b),
                  growth=rlim(b),
                  model='Schaefer')
# P-T model
dbdt <- function(b) r/p * b * (1 - (b/K)^p)
p <- 0.2
dfr <- rbind(dfr,data.frame(biomass=b,
                            production=dbdt(b),
                            growth=rlim(b),
```
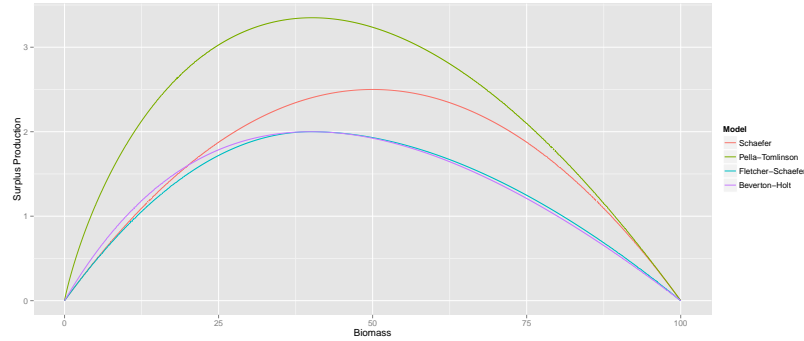
Figure 1: Production functions for different models assuming $\phi = 0.4$

```
                               model='Pella-Tomlinson'))
# Fletcher-Schaefer model
dbdt <- function(b) {
  bf <- b[which(b>bmsy)]
  bs <- b[which(b<=bmsy)]
  c(r * bs * (1 - bs/h),g * m * bf/K * (1 - (bf/K)^(n-1)))
}
n <- 1.1881
bmsy <- (1/n)^(1/(n-1)) * K
h <- 2*bmsy
m <- r*h/4
g <- (n^(n/(n-1)))/(n-1)
dfr <- rbind(dfr,data.frame(biomass=b,
                            production=dbdt(b),
                            growth=rlim(b),
                            model='Fletcher-Schaefer'))
# Beverton-Holt model
dbdt <- function(b) alpha * b / (1 + beta * b) - M * b
phi <- 0.4
M <- 0.1
objective <- function(alpha) phi - (sqrt(alpha/M)-1)/((alpha/M)-1)
alpha <- uniroot(objective,interval=c(0,10))$root
beta  <- 1/K * (alpha/M - 1)
dfr <- rbind(dfr,data.frame(biomass=b,
                            production=dbdt(b),
                            growth=rlim(b),
                            model='Beverton-Holt'))
```

2   The bdm package

We begin with an example application using hake from the Chatham rise in New Zealand.
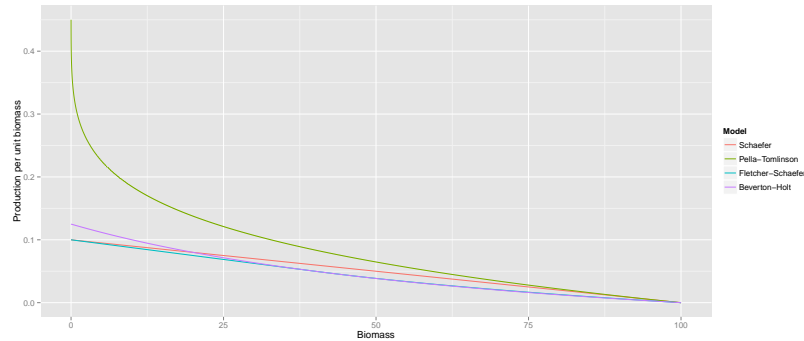
```
# load data
data(hakcr)
```

3

Figure 2: Growth rate per capita as a function of biomass

```
# create empirical data.frame
dat <- edat(harvest=hakcr$catch,index=cbind(hakcr$survey,hakcr$cpue),year=rownames(hakcr))

## Re-normalised indices.

# inspect r prior
#hist(r)

# get stan code and compile
mdl <- bdm()
mdl <- update_bdm(mdl,list(a=r@lognormal.par[['E[log(x)]']],b=r@lognormal.par[['SD[log(x)]']],par='r'))
mdl <- compile_bdm(mdl)

##
## TRANSLATING MODEL '.Object@model_code' FROM Stan CODE TO C++ CODE NOW.
## COMPILING THE C++ CODE FOR MODEL '.Object@model_code' NOW.
## cygwin warning:
##   MS-DOS style path detected: C:/PROGRA~1/R/R-31~1.0/etc/x64/Makeconf
##   Preferred POSIX equivalent is: /cygdrive/c/PROGRA~1/R/R-31~1.0/etc/x64/Makeconf
##   CYGWIN environment variable option "nodosfilewarning" turns off this warning.
##   Consult the user's guide for more details about POSIX paths:
##     http://cygwin.com/cygwin-ug-net/using.html#using-pathnames
## C:/Program Files/R/R-3.1.0/library/rstan/include//stansrc/stan/agrad/rev/var_stack.hpp:49:17: warning: 'void s
## C:/Program Files/R/R-3.1.0/library/rstan/include//stansrc/stan/agrad/rev/chainable.hpp:87:17: warning: 'void s

# inspect code
mdl

## S4 class stanmodel 'BDM' coded as follows:
##
##     data {
##        int T;
##        int I;
##        real index[T,I];
##        real harvest[T];
##        real n;
##        real sigmaO[I];
##        real sigmaP;
```

```
##      }
##      parameters {
##        real<lower=3,upper=30> logK;
##        real<lower=0,upper=2> r;
##        real<lower=0> xdev[T];
##      }
##      transformed parameters {
##
##        real x[T];
##        real q[I];
##        real H[T];
##
##        // variance terms
##        real sigmaOsq[I];
##        real sigmaPsq;
##
##        // fletcher-schaefer
##        // parameters
##        real dmsy;
##        real h;
##        real m;
##        real g;
##
##        dmsy <- pow((1/n),(1/(n-1)));
##        h <- 2*dmsy;
##        m <- r*h/4;
##        g <- pow(n,(n/(n-1)))/(n-1);
##
##        // variance terms
##        for(i in 1:I)
##          sigmaOsq[i] <- square(sigmaO[i]);
##        sigmaPsq <- square(sigmaP);
##
##        // compute biomass dynamics
##        x[1] <- 1.0 * xdev[1];
##        H[1] <- fmin(exp(log(harvest[1]) - logK),0.99);
##        for(t in 2:T){
##          if(x[t-1]<=dmsy) x[t] <- (x[t-1] + r * x[t-1] * (1 - x[t-1]/h) - H[t-1]) * xdev[t-1];
##          if(x[t-1]> dmsy) x[t] <- (x[t-1] + g * m * x[t-1] * (1 - pow(x[t-1],(n-1))) - H[t-1]) * xdev[t-1];
##         H[t] <- fmin(exp(log(harvest[t]) - logK),x[t]);
##        }
##
##        // compute catchability assuming
##        // constant sigmaO over time
##        // and uniform prior on ln(q)
##        {
##          real err;
##          real p;
##          for(i in 1:I){
##            err <- 0.0;
##            p <- 0.0;
##            for(t in 1:T){
##              if(index[t,i]>0.0 && x[t]>0.0) {
##                err <- err + log(index[t,i]/x[t]);
##                p <- p + 1.0;
##              }
##            }
```

```
##              if(p>2.0) { q[i] <- exp(err/p); // exp(sigma0sq[i] * (p-2)/(2*p) + err/p);
##            } else q[i] <- 0.0;
##          }
##        }
##      }
##      model {
##
##         // prior densities for
##         // estimated parameters
##         // *******************
##         logK ~ uniform(3.0,30.0);
##         r ~ lognormal(-0.913,0.405);
##
##         // random deviations
##         // *****************
##         xdev ~ lognormal(log(1.0)-sigmaPsq/2,sigmaP);
##
##         // observation equation
##         // *******************
##         for(i in 1:I){
##           for(t in 1:T){
##             if(index[t,i]>0.0 && x[t]>0.0 && q[i]>0.0)
##               index[t,i] ~ lognormal(log(q[i]*x[t])-sigma0sq[i]/2,sigma0[i]);
##           }
##         }
##
##         // apply penalty for H>0.95
##         // **************************
##         for(t in 1:T){
##         real H_; H_ <- H[t]/x[t];
##           if(H_>0.95) {
##             increment_log_prob(-log(H_/0.95) * (1/sigmaPsq));
##           }
##         }
##      }
##      generated quantities {
##
##         real biomass[T];
##         real depletion[T];
##         real harvest_rate[T];
##         real surplus_production[T];
##
##         real current_biomass;
##         real current_depletion;
##         real current_harvest_rate;
##
##         real biomass_at_msy;
##         real harvest_rate_at_msy;
##
##         real observed_index_biomass[T,I];
##         real observed_index_depletion[T,I];
##         real predicted_index[T,I];
##
##         for(t in 1:T) {
##           biomass[t] <- x[t] * exp(logK);
##           depletion[t] <- x[t];
##           harvest_rate[t] <- harvest[t]/exp(log(x[t]) + logK);
```

```
##          if(x[t]<=dmsy) surplus_production[t] <- r * x[t] * (1 - x[t]/h) * xdev[t];
##          if(x[t]> dmsy) surplus_production[t] <- g * m * x[t] * (1 - pow(x[t],(n-1))) * xdev[t];
##       }
##
##       current_biomass <- biomass[T];
##       current_depletion <- x[T];
##       current_harvest_rate <- harvest_rate[T];
##
##       biomass_at_msy <- dmsy * exp(logK);
##       harvest_rate_at_msy <- m / dmsy;
##
##       for(i in 1:I){
##         for(t in 1:T){
##           if(index[t,i]>0.0) {
##             observed_index_biomass[t,i]   <- (index[t,i]/q[i]) * exp(logK);
##             observed_index_depletion[t,i] <- index[t,i]/q[i];
##           }
##           predicted_index[t,i] <- q[i]*x[t];
##         }
##       }
##     }
##

# update depletion at MSY
shape(dat) <- 0.4
shape(dat,"n")

## [1] 1.188

shape(dat,"phi")

## [1] 0.4
```

References

[1] C. J. Krebs. *Ecology: The experimental analysis of distribution and abundance.* Harper and Row, New York, NY, USA, 3 edition, 1986.

[2] M. B. Schaefer. Some aspects of the dynamics of populations important to the management of commercial marine fisheries. *Bulletin of the Inter-American Tropical Tuna Commission Bulletin*, 1(2):26–56, 1954.

[3] M. B. Schaefer. A study of the dynamics of the fishery for yellowfin tuna in the eastern tropical pacific ocean. *Bulletin of the Inter-American Tropical Tuna Commission Bulletin*, 2:247–285, 1957.

[4] J. J. Pella and P. K. Tomlinson. A generalized stock production model. *Inter-American Tropical Tuna Commission Bulletin*, 13:421–496, 1969.

[5] W. J. Jr. Fox. An exponential surplus yield model for optimising exploited fish populations. *Transactions of the American Fisheries Society*, 99(1):80–88, 1970.

[6] R. I. Fletcher. On the restructuring of the pella-tomlinson system. *Fishery Bulletin*, 76(3):515–521, 1978.

[7] M. H. Prager. Preliminary application of a non-equilibrium generalized production model to north atlantic swordfish. Technical report, 1999.

[8] M. K. McAllister, E. A. Babcock, E. K. Pikitch, and M. H. Prager. Application of a non-equilibrium generalized production model to south and north atlantic swordfish: combining bayesian and demographic methods for parameter estimation. *Collected Volume of Scientific Papers ICCAT*, 51(5):1253–1550, 2000.