Estimates of population status of humpback chub (*Gila cypha*) in the Grand Canyon based on a length-based mark-recapture assessment model.

Steven J. D. Martell

UBC Fisheries Centre,

2202 Main Mall,

Vancouver, BC

V6T 1Z4,

CANADA

2011-09-27

Abstract

A length-structured population model that incorporates mark-recapture data was developed to estimate abundance of humpback chub (*Gila cypha*) in the Grand Canyon between 1989 and 2011. The model was fit to observations on catch-at-length and capture-recapture data from sampling programs that employed a variety of gear types including electrofishing, tramel netting, and hoop nets. Previous assessment models were age-based and have been shown to produce biased estimates of recent recruitment estimates. The age-structured model was also limited to data on fish that were greater than 150mm total length; the minimum size required for tagging. The length-structured model developed here makes no assumptions about age of individual fish and is also fit to catch-at-length data for fish that are too small to tag. Model outputs are based the number of fish greater than a specified size, or in a fixed size interval, there are no age-based results.

Contents

1 Introduction

In this report, I develop a length-structured mark-recapture model (hereafter, LSMR) where the accounting system for population numbers is based purely on length. The model is a statistical catch-at-length model, where the initial length distribution and recruitment each year are treated as latent variables to be estimated by fitting the model to a series of catch-at-length observations take over a period of time. A separable function is developed to estimate the year and size effect in observed catch-at-length data. The statistical nature of the model is very similar to that of Fournier and Archibald (1982), but is based on catch-at-length rather than catch-age data. The model is also fit to length-based capture-recapture data, where the growth and survival of tagged animals is updated at each time step and the predicted ratio of marked and unmarked fish-at-length is used to estimate the length-based recapture rates.

2 Methods

There are two major methodological components in this length-based model: (1) the development of an individual based model (IBM) for simulating a capture recapture program, and (2) a statistical catch-at-length mark-recapture model to estimate the number of individual fish in each length-class in each sampling year. A detailed description of the IBM simulation model is provided in the appendix; in short, this simulation model generates a matrix of the number of fish captured-at-length, a matrix of the number of newly marked fish released-at-length, and a matrix of marked fish recaptured-at-length. The remained of this section is a detailed analytical description of the statistical catch-at-length model used to estimate the abundance-at-length of humpback chub in the Grand Canyon.

  The following is a description of the analytical model for the length-structured mark-recapture model (hereafter, LSMR) used in this assessment. I present the analytical model in the form of a table where the order in which model equations are presented also represent the order in which the calculations proceed in the computer code. Equations presented in each table are referenced, for example, as (T1.1), where the T1 refers to Table 1, and the .1

refers to the first equation in that table. The LSMR model was implemented in AD Model Builder (Fournier et al., 2011), and the template code is available in the appendix of this document as well as a Git code repository (`https://code.google.com/p/lsmr-project/`). The description of the Length-Structured Mark-Recapture (LSMR) model is broken down into: input data, estimated parameters, dynamics of numbers-at-length, capture probability, and negative log-likelihoods and prior densities.

The following notation is used to define the dimensions of various variables. Vector quantities are designated with an an arrow ($\vec{x}$) or with a single subscript, and matrix is denoted by boldface uppercase letters ($\mathbf{X}$) and where two subscripts are shown denotes the element specific calculation. Higher dimensional arrays are indicated by normal upper case letters with 3 or more subscripts.

2.1 Input data

The model dimensions consists of time intervals (year indexed by i) and length intervals (index by j, T1.1). Capture-recapture data for the humpback chub have been collected on an annual basis since May 1, 1989, and the latest capture record in this analysis is February 27, 2012. The principle input data for LSMR consists of model dimensions (e.g., years, length intervals), catch-at-length $\mathbf{C}_{ij}$ for each year, the number of new marks released at length $\mathbf{M}_{ij}$, and the number of recaptured marks at length $\mathbf{R}_{ij}$.

2.2 Estimated parameters & parametric functions

An array of estimated parameters is denoted by Θ and consists of $8 + (I - 1) + J$ unknowns where I is the total number of years and J is the total number of length intervals. The average initial numbers-at-length is defined as $\dot{N}$, and the average number of new recruits at each time step is denoted by $\bar{R}$. To initialize the number of individuals in each length interval Λ in the initial year $i = 1$, the average initial number is multiplied by a length specific lognormal deviate η_j (T1.11) with the added constraint $\sum_j \eta_j = 0$ to ensure that $\dot{N}$ is identifiable. Similarly, the average number of newly recruiting fish in each time step is based on deviates at each time-step from the mean recruitment (T1.12), where again the constraint $\sum_i \nu_i = 0$ ensures that $\bar{R}$ is identifiable.

Natural mortality is a function of length (T1.6), where the natural mortality at the asymptotic length M_∞ is estimated from the data. Selectivity is also assumed to be a parametric function of length (T1.7) where l_x and g_x represent length-at-50% vulnerability and the standard deviation of the logistic function, respectively. Note that in (T1.6) the estimated natural mortality rate is confounded with asymptotic length l_∞ which is also an estimated parameter along with the von Bertalanffy growth coefficient k. The growth parameters are used to calculate a vector of growth increment $\vec{\Delta}$ assuming von Bertalanffy growth (T1.8). An additional parameter β is used to characterize the variability in annual growth increments for individual fish.

2.3 Growth transition

The asymptotic length l_∞ is defined as the average asymptotic length for a population of fish. It is assumed in (T1.8) that individuals greater than l_∞ continue to grow at a much reduced rate k; this is accomplished by exponentiating the growth increment equation $((l_\infty - x_j)(1 - exp(-k\tau)))$, adding 1.0, and taking the natural logarithm ensuring that (T1.8) remains positive for all positive values of l_∞, k, and Λ. The probability of transitioning from one length interval x_j to length intervals greater than x_j is based on a gamma density function (T1.9), where the mean is denoted by Δ_j and a variance equal to $\Delta_j \beta$. Each row of $\mathbf{P}_{j,j'}$ is normalized to sum to 1.0, and $\mathbf{P}_{j,j'} = 1.0$ when $j = j' = n$, where n is the number of length intervals (i.e., individuals in the last length interval represent a plus group).

2.4 Dynamics of numbers-at-length

2.5 Length-based capture probability

2.6 Negative log-likelihoods and prior densities
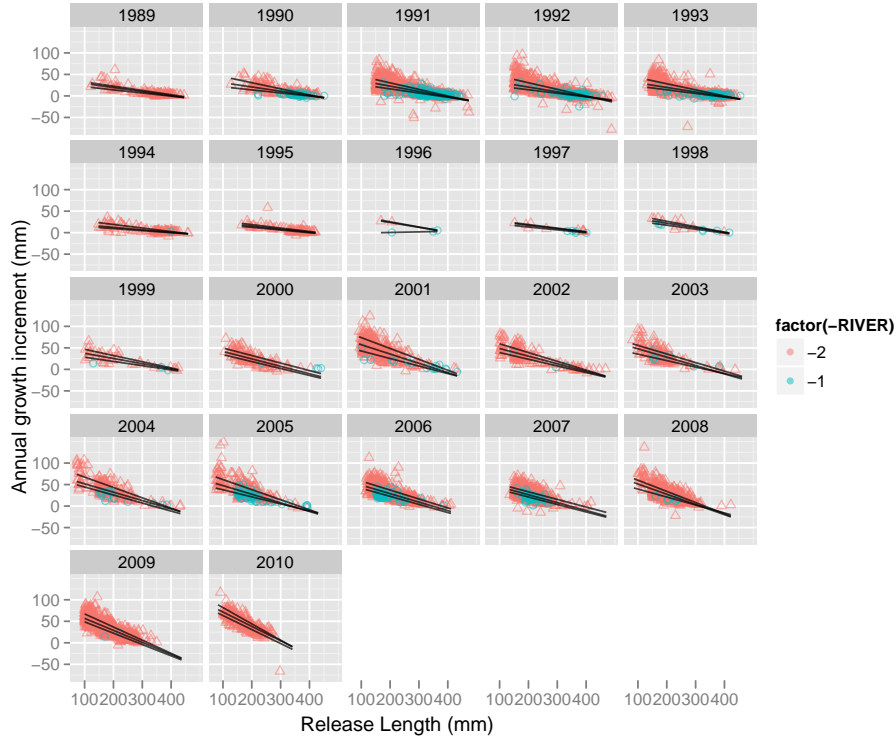
2.7 Length-structured mark-recapture model (LSMR)

Figure 1: caption

Table 1: Data, parameters, and analytical procedures for the length-based mark-recapture model.

INDEXES, DATA & CONSTANTS

index for time, index for length interval	i, j	(T1.1)
time step	τ	(T1.2)
set of midpoints of length intervals	$\Lambda = \{x_1, \ldots, x_J\}$	(T1.3)
catch, new marks, recaptures	$\mathbf{C}_{i,j}, \mathbf{M}_{i,j}, \mathbf{R}_{i,j}$	(T1.4)

PARAMETERS & DERIVED VARIABLES

estimated parameters
$$\Theta = \{\dot{N}, \bar{R}, M_\infty, l_\infty, k, \beta, l_x, g_x, \vec{\eta}, \vec{\nu}\} \quad \text{(T1.5)}$$

mortality-at-length
$$\vec{m} = \frac{M_\infty l_\infty}{\Lambda} \quad \text{(T1.6)}$$

selectivity-at-length
$$\vec{s} = \frac{1}{1 + \exp(-(\Lambda - l_x)/g_x)} \quad \text{(T1.7)}$$

growth increment
$$\vec{\Delta} = \ln(\exp[(l_\infty - \Lambda)(1 - \exp(-k\tau))] + 1) \quad \text{(T1.8)}$$

length transition probability
$$\mathbf{P}_{j,j'} = \int_{x_j - x^*}^{x_j + x^*} \frac{x_{j'}^{(\Delta_j/\beta - 1)} e^{x_{j'}/\beta}}{\beta^{\Delta_j/\beta} \Gamma(\Delta_j/\beta)} dx,$$

$$\sum_{j'=1}^{J} P_{j,j'} = 1 \quad \text{(T1.9)}$$

total numbers, marked numbers
$$\mathbf{N}, \mathbf{T} \quad \text{(T1.10)}$$

INITIAL STATES ($i = 1$, $j = 1$)

initial numbers-at-length
$$\mathbf{N}_{i,j} = \dot{N} \exp(\eta_j), \quad \text{where} \sum_j \eta_j = 0 \quad \text{(T1.11)}$$

new recruits
$$\mathbf{N}_{i,j} = \bar{R} \exp(\nu_i), \quad \text{where} \sum_i \nu_i = 0 \quad \text{(T1.12)}$$

DYNAMIC STATES ($i > 1$)

capture probability	f_i	(T1.13)

References

Fournier, D. and Archibald, C. (1982). A general theory for analyzing catch at age data. *Canadian Journal of Fisheries and Aquatic Sciences*, 39(8):1195–1207.

Fournier, D., Skaug, H., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M., Nielsen, A., and Sibert, J. (2011). Ad model builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models.

R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Zhang, Z., Lessard, J., and Campbell, A. (2009). Use of bayesian hierarchical models to estimate northern abalone, haliotis kamtschatkana, growth parameters from tag-recapture data. *Fisheries Research*, 95(2-3):289–295.

A   Work plan

The following outlines a work plan for the assessment of abundance for humpback chub. The objective is to develop a much more flexible length-based model that can be used to explore alternative hypotheses about natural mortality rates, cumulative effects of release mortality for intensive sampling periods, and to potentially integrate other sources of environmental variations such as the effects of turbidity on capture probability and recruitment variation.

Analytical approach

I will develop a statistical catch-at-length model in the AD Model Builder software and additions R-scripts for manipulating data and summarizing model results. Input data for the model will consist of a matrix of the total catch-at-length in 5 to 10 mm size intervals for all years, a matrix of the number of marks released by size and year, a matrix of the number of marks recaptured by size and year, and a three dimensional array of the number of marks recaptures by size for each tag-cohort released (optional).

   Estimated parameter will include: natural mortality rate, growth parameters, a vector of the initial numbers in each length interval, and a vector of age-0 recruits each year. Propagation of the numbers-at-length to the next time step will be based on a size-transition matrix, which is a function of the growth parameters and variation in growth. Observation models will include a probability of capturing an animal of a given length each year, the probabilities of capturing a marked and unmarked animal of a given length, and optionally the probability of recapturing a specific tag-cohort of a given length. These predicted observations will be compared with the empirical data using a negative binomial likelihood function. The negative binomial model is more suitable here because it can account for over-dispersed data and accommodate zero observations in cases where there is sparse information.

Detailed work plan

Major components of the project

The following is list of milestones for this project. Each of these items will be expanded upon in the section on project details.

1. Data acquisition and processing.

2. Development of an operating model for simulating data with known parameter values.

3. Development of a length-based assessment model to be fitted to data on capture and recapture information by length interval.

4. Simulation testing; exploring the precision and bias of the assessment model in jointly estimating recruitment, size-specific capture probability, and growth and natural mortality using simulated data sets.

5. Application of the length-based model to the HBC data.

6. Quantifying uncertainty in model parameters and estimates of recruits using Markov Chain Monte Carlo methods to integrate the joint posterior distribution.

7. Report & presentation to the Technical Working Group.

Project details

Data acuisition & processing The necessary data required to conduct the analysis will require information from the following fields in the GCMRC database: FISHNO, TRIP_ID, DATES, RM, RIV, TL, TAGNO. The following SQL statement was used in a previous study to extract the necessary information. Note that the following code has been modified to obtain all fish lengths, not just those greater than 150 mm.

```
--****
--All
--****

CREATE OR REPLACE VIEW FISH.V_ASMR_2009_ALL
(
    FISHNO,
    TRIP_ID,
    DATES,
    RM,
    RIV,
    TL,
    TAGNO
)
AS
SELECT "CAPTURE_ID" fishno, "TRIP_ID" trip_id, "START_DATE" dates,"START_RM" rm,"RI
       FROM FISH.CAPTURE_HISTORY_20091211_0832
     WHERE SPECIES_CODE = 'HBC'
     AND (
       (RIVER_CODE = 'COR' AND START_RM >= 57 AND START_RM <= 68.5)
       OR RIVER_CODE = 'LCR'
       )
     AND START_DATE >= TO_DATE('04/01/1989', 'MM/DD/YYYY')
     AND TOTAL_LENGTH >= 00
```

An R-script will be developed for post processing of the data to assign the length capture and recapture information into discrete length intervals for assembling input data into the assessment model.

Operating model An individual based model will be developed to generate simulated data sets with known natural mortality rates, recruitment vectors, growth rates and capture probabilities. The pseudo code for the operating model is as follows:

1. Specify a vector of absolute recruitment from 1950-2011.

2. For each individual recruit in each year apply the following procedure:

    (a) boolean trail for survival, if the animal survives then go to (b) else, individual died and restart at step 2.

    (b) boolean trail for capture:

        i. Captured: obtain length of fish, if greater than 150mm then tag and release fish, go.

        ii. Recaptured: obtain length of fish, goto step (a).

        iii. Not captured: goto step (a).

3. Store information about individual capture history and length into simulated database.

The above algorithm is intended to generate the exact data that is currently collected in the HBC monitoring program. Specific details about factors that affect capture probability and survival would be incorporated into the boolean probabilities defined in 2a and 2b.

Length-based assessment model A statistical length-based assessment model is similar in nature to a statistical catch-age model in that numbers-at-age, or in this case, numbers-at-length are propagated forward in time. The previous Age-Structured Mark-Recapture Model (ASMR) was dependent on catch-age information; age data for HBC were inferred from an analytical age-length key (i.e., based on inferences about growth, not empirical age-length data). Estimates of uncertainty were overly precise due to the pre-processing of the data to be used with ASMR. The length-based model makes no such inferences about the age of individual fish and is based strictly on the length observation data.

In a length based model, individuals in a given length interval are propagated in time by redistributing these individuals into new length bins based on a length transition probability (Fig 2). The length transition probability is a function of growth and the time interval between sampling events. The graphical example in Fig 2 does not account for mortality over time, its only meant to show the transition of individuals from one length bin to subsequent length bins.

Simulation testing The purpose of simulation testing is to (i) demonstrate that the model is able to estimate the model parameters given perfect information and satisfying all of the model assumptions, (ii) examine precision and bias in parameter estimates when faced with observation, process, and structural errors, and lastly (iii) to examine the
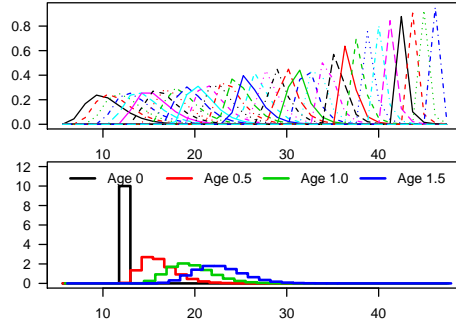
Figure 2: A graphical example of a length-based transition. Starting with 10 age-0 individuals, these 10 would then be distributed to the age-0.5 distribution distribution. The age-0.5 distribution then transitions to the age-1.0 distribution and so on.

estimability, bias, and precision of model parameters when underlying structural assumptions are not met. Simulation testing will be conducted to examine the ability to jointly estimate recruitment, capture probability, growth and natural mortality in a reliable and unbiased manner.

Application to the HBC data The length-based assessment model will then be applied to the length-based capture and recapture data for the humpback chub monitoring program. Model outputs will include estimates of recruits (and associated uncertainty), estimated model parameters (and uncertainty). It is anticipated that more reliable estimates of recruits will be available with the length based model because the model is not limited to length information that is greater than 150mm, as is the case in the ASMR model.

Quantifying uncertainty Estimates of uncertainty will jointly consider uncertainty in the mark-recapture data, growth and mortality. To do so the joint posterior distribution of the data will be constructed numerically using a Markov Chain Monte Carlo procedure (using the Metropolis Hastings algorithm implemented in AD Model Builder). Uncertainty in model parameters as well as outputs will be cast in the form of marginal posterior densities.

Report and presentation This is original research and the methods outlined for a length-based model that incorporates growth increment data from a mark recapture program has not been previously published to my knowledge. Ideally these results will be disseminated in the primary literature, but will also be presented to the Technical Working Group and be available as a technical report (e.g., USGS Open File Report). Also source code, scripts, and documentation will be hosted on an open source repository with version control. The intention here is to create a repository for continued development of the software and to document the historical changes over time.

B Individual based model for simulating the dynamics and sampling of humpback chub in the Grand Canyon

B.1 Introduction

The following is a detailed description of the simulation model that was used for simulating the population dynamics and data collection programs for humpback chub in the Grand canyon. We first describe in detail the life-history trajectory of an individual fish: the survival probability, growth, and capture history and provide the documented code to implement this process. I then describe the data structures that resemble a databased of individual capture histories for both tagged and untagged fish. The individual based model (IBM) is then used to estimate the fate and capture histories of a known number of recruits starting life at a 40mm length interval.

The simulation model was constructed using R (R Development Core Team, 2009). The algorithm populates three matrixes that contain the total number of fish captured in year t at length interval x, the total number of newly marked fish, and a matrix of the total number of recaptured individuals. At each time step the fish is alive, information on length and age is stored along with information on capture history, the tag number if the fish was large enough to tag.

B.2 R-code

```
## ------------------------------------------------------------ ##
## Simulation model for humpback chub                           ##
## Author: Steven Martell                                       ##
## DATE: September 26, 2011                                     ##
## Contact: martell.steve@gmail.com                             ##
##                                                              ##
## FUNCTIONS:                                                   ##
##   − .ibm ⇒ given N recruits returns list with the capture    ##
##            history of each recruit. It also fills the        ##
##            global variables CM and R for the assessment      ##
##            model.                                            ##
##   − .runIBM Loops over syr:nyr and applies the .ibm function##
##            to each of Rt recruits.                           ##
##                                                              ##
##                                                              ##
## ------------------------------------------------------------ ##


## -- Required libraries ------------------------------------- ##
```

```
require( PBSmodelling )
require( Hmisc )
require( ggplot2 )


## -- Simulation controls -------------------------------- ##

#  Model Dimensions
syr       <- 1950      #initial year
nyr       <- 2011      #final year
dt        <- 3/12      #time step
tmax      <- 50/dt     #max longevity index (50 yrs)
dyr       <- seq(syr, nyr, by=dt)
xbin      <- seq(3, 55, by=1)

#  Growth (units in cm)
lmin       <- 4
lmin.sig   <- 0.4
m.linf     <- 0.08
linf       <- 40    #Asymptotic length (cm)
k          <- 0.18 #Growth coefficient

#  Selectivity
lh         <- 10    #length @ 50% selectivity
gamma      <- 1.5   #std in selectivity

sample.yrs  <- seq(1989, 2011, by=dt)
fishing     <- dyr %in% sample.yrs
min.size    <- 15.0 #minimum size for tagging (15 cm)



#  Parameter vector for the IBM model
THETA <- list( lmin=lmin,
               lmin.sig=lmin.sig,
               m.linf=m.linf,
               lh=lh,
               gamma=gamma )

set.seed(991)
iclr <- rev(topo.colors(length(syr:nyr),0.5))
Rt<-floor(rlnorm(length(syr:nyr), log(300), 0.9))
```

13

```
h1 <- 0.04
h2 <- 0.20
t  <- min(sample.yrs)
T  <- max(sample.yrs)
pr <- 1
Et <- h1+(h2-h1)*cos(0.5+(sample.yrs-t)/(T-t)*pr*pi)^2
#Et<-runif(length(sample.yrs), 0.05, 0.2)  #Effort

#  Global objects for storing data
fish.id      <- 0    #Unique id for individual fish
tag.id       <- 0    #Unique tag no. for individual
#  Total catch, marked, recap, unmarked by year at length interval
C  <- matrix(0,nrow=length(sample.yrs), ncol=length(xbin))
M  <- matrix(0,nrow=length(sample.yrs), ncol=length(xbin))
R  <- matrix(0,nrow=length(sample.yrs), ncol=length(xbin))

## ------------------------------------------------------------ ##




## ------------------------------------------------------------ ##
## PROCEDURES                                                   ##
## ------------------------------------------------------------ ##


.ibm <- function( Nt, jyr, THETA )
{
    #ARGUMENTS:
        #Nt    = Number of recruits from cohort
        #jyr   = brood year (year the cohort was age-0)
        #THETA = a list of simulation parameters for the ibm.

    #RETURNS:
        #df    = a list w records of capture date, len,  tag.no

    with(THETA, {
        #jyr <- syr+(ii-1)
        ## indexs i = individual; j=time

        ## Calculate length & mortality for time j.
```

```
lj <- linf*(1.-exp(-k*dt*1:tmax))
mj <- m.linf*linf/lj

## Draw initial recruit size (gamma).
scale   <- (lmin.sig)^2/lmin
shape   <- lmin/scale
li <- rgamma(Nt, shape, 1/scale)


## Growth trajectory
growth <- function ( li )
{
    linf.i <- rnorm(1, linf, 0.1*linf)
    li+(linf.i-li)*(1-exp(-k*1:tmax*dt))
}
L <- sapply(li, growth)

## Survival trajectory
## S is a list of lengths of individuals at
## the time they were alive.
survival <- function ( lj )
{
    sj <- exp(-m.linf*linf/lj*dt)
    pj <- rbinom(tmax, 1, sj)
    nj <- which.min(pj)
    return(lj[1:nj])
}
S <- apply(L, 2, survival)


## Capture history
pcap <- function ( lj )
{
    #Arg: lj is the length at times j
    #Algorithm:
    #-1. determine index for sampling time periods (fyr)
    #-2. determine capture probability at sampling times
    #-3. if captured fish > 150 mm then assign a tag no.
    if(!is.null(lj))
    {
        # 1. Index for sampling time periods (fyr)
        n    <- length(lj)
```

```r
d1  <- which(dyr==jyr)  #min index of dyr for jyr
d2  <- min(d1+n-1, length(dyr))
jj  <- d1:d2
iyr <- dyr[jj]              #years alive
fyr <- fishing[jj]          #years fished fyr==1


## Did not survive into sampling years
if(length(iyr[fyr])==0) return(NULL)

# -2. Selectivity at time of sampling
len <- (lj[1:length(jj)])[fyr]
iyr <- iyr[fyr]
# TODO Add annual effort to sj capture probability
eyr <- findInterval(iyr, sample.yrs)
#print(cbind(sample.yrs[eyr],Et[eyr]))
sj  <- Et[eyr]*plogis(len, lh, gamma)

cj  <- rbinom(length(sj), 1, sj)
rid <- which(cj==1)  #row index for capture
if(length(rid)==0) return(NULL) #never captured

tmp <- c(iyr[rid], len[rid],
    tag.no=rep(NA,length=length(rid)))
tmp <- matrix(tmp, ncol=3)


# -3. Assign tag.id if greater than 150 mm
if(max(tmp[,2])>=min.size)
{
    tag.id <<- tag.id + 1
}
tmp[len[rid]>=min.size, 3] <- tag.id

# -4. Populate global variables with catch history
tt <- findInterval(tmp[,1],sample.yrs)
xx <- findInterval(tmp[,2],xbin)
for(i in 1:length(tt))
    C[tt[i], xx[i]] <<- C[tt[i], xx[i]] + 1

#print(cbind(tmp,sample.yrs[tt],xbin[xx],tmp[,3]))
```

```r
                    # - Newly  marked  fish  into  matrix M
                    im    <- !is.na(tmp[,3])
                    if(sum(im)>0) #fish  was  large  enough  to  tag
                    {
                         it <- min(which(im))
                         iy <- findInterval(tmp[it,1],sample.yrs)
                         ix <- findInterval(tmp[it,2],xbin)

                         #print(cbind(tmp,  sample.yrs[iy],xbin[ix]))
                         M[iy, ix] <<- M[iy, ix] + 1

                         ir <- which(im)[-1]
                         ry <- findInterval(tmp[ir,1],sample.yrs)
                         rx <- findInterval(tmp[ir,2],xbin)
                         for(i in 1:length(ry))
                             R[ry[i], rx[i]] <<- R[ry[i], rx[i]] + 1
                    }

                    return(tmp)
                }
            }
        P <- sapply(S, pcap)
        return(P)
    })
}

.runIBM <- function()
{
    j <- 1
    A <- NULL
    for(i in syr:nyr)
    {
        A <- c(A, .ibm(Rt[j], i, THETA))
        j <- j+1
    }
    return(A)
}

.writeLSMRdata <- function()
{
        # This function writes the data file for LSMR model
        fn <- "simLSMR.dat"
```

```r
        C  <- cbind(floor(sample.yrs),sample.yrs%%1*4+1,C)
        M  <- cbind(floor(sample.yrs),sample.yrs%%1*4+1,M)
        R  <- cbind(floor(sample.yrs),sample.yrs%%1*4+1,R)

        write("#Simulated_data_from_HBCsim.R", fn)
        write("#Model_Dimensions", fn, append=T)
        write("#Years,_timestep", fn, append=T)
        write(c(range(sample.yrs), dt), fn, append=T)
        write("#Array_dimensions(C,_M,_R)", fn, append=T)
        write(dim(C), fn, append=T)
        write("#Number_of_length_intervals", fn, append=T)
        nbin=length(xbin)+1
        write(nbin, fn, append=T)
        write("#Length_intervals_(cm)", fn, append=T)
        write(c(xbin,xbin[nbin-1]+diff(xbin[(nbin-2):(nbin-1)])), fn, append=

        write("#Catch_at_length_by_period", fn, append=T)
        write("#Year,_Period,_Count_...", fn, append=T)
        write.table(C, fn, append=T, col.names=F, row.names=F)
        write("#Number_Marked_at_length_by_period", fn, append=T)
        write("#Year,_Period,_Count_...", fn, append=T)
        write.table(M, fn, append=T, col.names=F, row.names=F)
        write("#Number_Recaptured_at_length_by_period", fn, append=T)
        write("#Year,_Period,_Count_...", fn, append=T)
        write.table(R, fn, append=T, col.names=F, row.names=F)

        write("#End_of_file", fn, append=T)
        write(999, fn, append=T)

        file.copy(fn, "../ADMB/srcLSMR/simLSMR.dat", overwrite=TRUE)
}

## ------------------------------------------------------------ ##
## MAIN                                                         ##
## ------------------------------------------------------------ ##
#A  <- .ibm(1000, 1980, THETA)
A <- .runIBM()
.writeLSMRdata()

## ------------------------------------------------------------ ##

par(mfcol=c(6, 4), las=1, mar=c(2, 2, 1, 1), oma=c(3, 3, 1, 1))
```

```r
for(i in seq(1,length(sample.yrs)-1/dt,by=1/dt))
{

        ut <- colSums(C[i:(i+3),]-M[i:(i+3),]-R[i:(i+3),])
        mt <- colSums(M[i:(i+3),])
        rt <- colSums(R[i:(i+3),])
        tmp <- rbind(ut, mt, rt)

    barplot(tmp,names.arg=xbin,xlab=""
    ,ylab="", main=sample.yrs[i], col=1:3)
}
mtext(c("Length (cm)","Frequency"),
    c(1, 2), outer=T, las=0, line=1)
```