

Evaluating the relationship between capture probability and uncertainty in estimates of humpback chub abundance using ASMR

Steven Martell
University of British Columbia
Fisheries Centre
2202 Main Mall
Vancouver, BC, Canada

November 5, 2010

Executive summary

Since the listing of humpback chub, *Gila cypha* (HBC) on the federal list of endangered species in 1967, mark-recapture data have been used to monitor the status and abundance of HBC in the Little Colorado River (LCR) since 1989. Since 1989, annual capture probabilities for HBC have ranged from less than 5% per year to greater 35% and has averaged nearly 21% since 2005. The current monitoring program for HBC in the LCR region consists of spring and fall sampling trips. Estimates of abundance based on an Age-Structured Mark-Recapture (ASMR) model are highly precise given the high mark-rate that exists in the LCR populations. Furthermore, estimates of uncertainty are likely too precise, and possibly biased, due to the conversion of fish-length into fish-age using a fixed age-length key that is based on a bioenergetics growth model, rather than traditional empirical age-length data.

The overarching objective of this study is to examine how estimates of abundance and uncertainty for HBC would change if monitoring efforts were reduced such that capture probabilities in future years were reduced by as much as 50%. To carryout this objective, seven alternative data scenarios were explored that included (1) all available data to represent the status quo, (2) no fall sampling, (3) no September sampling, (4) no October sampling, (5) no sampling of the lower 1200m of the LCR reach, (6) no April samples or first spring trips, and (7) no US Fish and Wildlife samples from the lower 5km of the LCR (excluding the lower 1200m spring trips). These seven alternative data sets were then used in the ASMR-3 model to estimate adult abundance (age 4+) and age-2 recruits along with the associated levels of uncertainty. To account for variation in individual growth and age miss-specification an aging error classification matrix was used to approximate the Monte Carlo procedure used in previous ASMR assessments for HCB in the LCR.

The assessments conducted herein were based on the ADModel Builder code supplied by Lewis G. Coggins (US FWS). This code was modified to accommodate the necessary scenarios and Monte Carlo procedures and was compared against the original code that was used in the previous assessment by Coggins (2008). Assessment results and retrospective analyses based on the modified code were exactly the same as that produced by the original code. I did not attempt to reproduce the results of Coggins (2008); those results are based on a Monte Carlo procedure for including the effects of aging error that could not be repeated due to unknown random number seeds.

Based on subsamples of the historical data, estimates of abundance are sensitive to which data is included in the analysis; it appears the basic assumptions of equal capture probability for marked and unmarked individuals is violated. Spatial or temporal reductions in sampling effort are likely to result in minor changes in age-specific capture probabilities that have the potential to bias future estimates of recruits and adult abundance due to this non-random sampling. Continued fall sampling and the spring samples from the lower 5km are very important to maintain consistency with historical data. Absolute uncertainty in the estimates of abundance are largely linked to assumptions about aging error and model structure. Decreases in sampling effort that lead to average capture probabilities

of less than 0.1 results in minor increases in uncertainty in age-2 recruits, but capture probabilities of less than 0.05 result in substantial increases in uncertainty. Estimates of variance decrease exponentially with increases in capture probabilities. Alternative mixed error models should be explored to better reflect the true uncertainty and to avoid the assumption of known ages in ASMR.

Contents

Executive summary	i
1 Introduction	1
2 Methods	2
2.1 Description of ASMR-3	2
2.2 Data Scenarios	4
2.3 Simulation model	6
3 Results	8
3.1 Reproducing results based on code from Coggins and Walters	8
3.2 Maximum likelihood estimates of abundance	8
3.3 Recent estimates of age-specific capture probability	11
3.4 Estimates of future age-2 recruits using ASMR	12
4 Discussion	15
5 Acknowledgements	17
A Setting up ASMR input files	22
B ADMB code & and example data file for this study	26

1 Introduction

Mark-recapture methods have been used to monitor the status and abundance of humpback chub (HBC) in the Little Colorado River (LCR) for more than two decades. Since 1989, capture probabilities for HBC have ranged from less than 5% in the mid to late 1990s to greater than 15% in more recent years (Coggins, 2008). The current monitoring program for HBC in the LCR consists of spring and fall sampling trips. The overarching objective of this work is to examine how estimates of uncertainty for HBC would change if monitoring efforts were reduced such that capture probabilities in future years were reduced by as much as 50% (ca., 7.5% per year). There are two stages to this project: 1) construct alternative datasets by removing temporal or spatial samples to determine the location and seasonal effects on capture probabilities and abundance estimates, 2) simulate future data collection from 2009 forwards and examine the relationship between capture probabilities and uncertainty in population estimates.

The Age Structured Mark Recapture (ASMR) assessment model that is currently used to assess the abundance and status of humpback chub in the Colorado and Little Colorado rivers was first published by Coggins et al. (2006). To use this method, the required data consist of a matrix of newly marked individuals at age in each year, and a matrix of recaptured individuals at age in each year. The method then attempts to reconstruct the numbers of unmarked individuals using Virtual Population Analysis (where unmarked numbers at age are reconstructed backwards in time by adding in the observed newly marked individuals), and this assumes there are no aging errors in the number of newly marked animals. At the same time, the method uses a synthetic approach (or forward propagation of numbers at age) to keep track of the number of tagged animals at age in each year. Estimated parameters in the model consist of a vector of terminal numbers-at-age and are estimated by using maximum likelihood methods assuming the mark and recapture data follow a Poisson distribution. Additional assumptions used in the Coggins (2008) assessment include: natural mortality is known and decays as a function of length (asymptotic $M=0.13$) following the description of Lorenzen (2000), growth, or length-at-age, is known and time invariant, there are no aging errors in the conversion of lengths to age. Unfortunately, it is not possible to age humpback chub with high certainty without sacrificing the animal for aging; therefore the catch-age data collected is inferred from catch-length data. Coggins (2008) have incorporated some aging error into the most recent assessment using a modified bootstrapping technique to create 1000 sample data sets then integrate assessment results over these replicate datasets. In general, the version of ASMR used herein, and in previous assessments, is an observation error-only model and thus estimates of uncertainty are very likely to be biased downward (i.e., overly precise).

In this report, I provide a general description of the version of ASMR used in this assessment (based on ASMR-3 in Coggins et al., 2006), and verify that the modified AD Model Builder (Otter Research, 2008) code is capable of reproducing the results of the original code supplied by Lew Coggins. I then describe the alternative data scenarios

that were constructed to examine the issue of changes in capture probabilities associated with changes in spatial and temporal sampling. Next, I describe an individual based model (IBM) that was used for simulating future data collection, and an algorithm used for estimating the probability of changes in recruitment.

2 Methods

2.1 Description of ASMR-3

The following is a brief summary of the Age Structured Mark Recapture model originally described by Coggins et al. (2006) that is used herein to estimate age-2 recruits, adult abundance (ages 4+) and capture probabilities based on the conditional maximum likelihood estimates of the age-specific capture probabilities. The observed data consists of the number of new marks released in year t at age a ($m_{t,a}$), and the number of recaptured marks in year t at age a ($r_{t,a}$). The actual age of fish sampled in the field are not known and were obtained based on length based on the following polynomial:

$$a = (-1.05952 \times 10^{-8}) \cdot l^4 + (1.25697 \times 10^{-5}) \cdot l^3 - 0.005058694 \cdot l^2 + 0.878738367 \cdot l - 53.55077549 \quad (1)$$

which is based on the growth function described in Coggins (2008), and was provided to the author by Lewis Coggins via email on December 15, 2009. We assume the youngest age-class tagged is 2 years and the oldest age-class is 50 years.

The equations ASMR-3 model are laid out in Table 1. The observations, new marks and recaptured marks, are defined in (T1.1) and the estimated model parameters (the log of the terminal numbers at age) are given by (T1.2). Survival is assumed to be age-specific and is based on an allometric relationship defined by Lorenzen (2000), that used the von-Bertalanffy growth coefficient ($k = 0.122$) and an asymptotic natural mortality rate of $M = 0.13$. This age-specific survival function is given by (T1.3), where a is the integer age. The number of unmarked fish at large in the terminal year (2009 for this assessment) is initialized based on the estimated parameter vector (Θ) for ages 2 to $A_t = 13$ years (T1.4). For ages greater than A_T years we assume a stable age distribution based on the age-specific survival rate (T1.5). The number of marked fish at large in the initial year is set equal to 0 (T1.6). To update the numbers-at-age for unmarked and marked individuals, we use a back calculation (akin to Virtual Population Analysis) to fill in the matrix of unmarked animals by adding back in the new marks (T1.7), and a forward calculation to survive marked individuals and add in new marks (T1.8). The conditional maximum likelihood estimate of the age-specific capture probability is given by (T1.9). The predicted number of new marks released and recaptured each year are given by (T1.10) and (T1.11), respectively.

The standardized residuals for the Poisson model are given by (T2.1) and (T2.2) for the number of marks at large and recaptures, respectively. The joint log-likelihood for the

Table 1: Age-structured mark-recapture model (ASRM-3).

Observations	
$m_{t,a}, r_{t,a}$	(T1.1)
Estimated parameters	
$\theta = (\{X_{T,a}\}_{a=1}^{a=A_T})$	(T1.2)
Age-schedule variables	
$s_a = \left(\frac{k(a+1) - 1}{ka - 1} \right)^{(-M/k)}$	(T1.3)
Initial states	
$U_{t,a} = \exp(X_{T,a}) \quad \{t = T, \quad 2 \leq a \leq A_T\}$	(T1.4)
$U_{t,a} = U_{t,a-1}s_{a-1} \quad \{t = T, \quad A_T < a \leq A\}$	(T1.5)
$M_{t,a} = 0 \quad \{t = 1\}$	(T1.6)
State dynamics	
$U_{t,a} = (U_{t+1,a+1}/s_a) + m_{t,a}$	(T1.7)
$M_{t+1,a+1} = s_a(M_{t,a} + m_{t,a})$	(T1.8)
$p_{t,a} = \frac{m_{t,a} + r_{t,a}}{U_{t,a} + M_{t,a}}$	(T1.9)
Predicted observations	
$\hat{m}_{t,a} = U_{t,a}p_{t,a}$	(T1.10)
$\hat{r}_{t,a} = M_{t,a}p_{t,a}$	(T1.11)

new marks and recaptures is given by (T2.3).

One of the key assumptions in the ASMR approach to reconstructing abundance is that fish age is known without error. Age-composition data was reconstructed from length based on the results of fitting a temperature dependent growth model to growth increment data which resulted in the polynomial (1) that was used to deterministically convert length to age. To account for variation in growth and mis-specification of age, the previous assessments used a Monte Carlo procedure to generate 1000+ data sets to incorporate aging errors into the assessment (Coggins, 2008). In this assessment, I did not repeat the aging error efforts of Coggins (2008) and note herein that estimates of uncertainty are biased in the direction of being too precise.

Table 2: Residuals and negative log-likelihoods for Poisson model.

Residuals for Poisson model	
$\epsilon_{t,a} = \frac{m_{t,a} - \hat{m}_{t,a}}{\sqrt{\hat{m}_{t,a}}}$	(T2.1)
$\delta_{t,a} = \frac{r_{t,a} - \hat{r}_{t,a}}{\sqrt{\hat{r}_{t,a}}}$	(T2.2)
Poisson log-likelihood	
$\ell(m, r \theta) = \sum_{t=1}^T \sum_{a=1}^A [\hat{m}_{t,a} - m_{t,a} \ln(\hat{m}_{t,a})] + \sum_{t=2}^T \sum_{a=2}^A [\hat{r}_{t,a} - r_{t,a} \ln(\hat{r}_{t,a})]$	(T2.3)

2.2 Data Scenarios

In total, seven alternative data sets were used to explore historical estimates of abundance and uncertainty using the ASMR model described by Coggins et al. (2006). These data sets were constructed based on queries from the database and the total number of records by month and year are summarized in Table 4. All queries and data were provided by Paul Alley (Database Administrator, Grand Canyon Monitoring and Research Center, U.S. Geological Survey, 2255 N. Gemini Dr., Flagstaff, AZ 86001). The seven alternative data sets are summarized as follows:

Scenario 1 represents the base line scenario where all available records between 1989 and 2009 are used to construct the ASMR input file.

Scenario 2 all September and October USFWS records have been removed (i.e., no fall sampling).

Scenario 3 all September USFWS records have been removed.

Scenario 4 all October USFWS records have been removed.

Scenario 5 all lower 1200 records have been removed.

Scenario 6 all April or first sampling trips of the spring have been removed, second trip or May trips have not been excluded.

Scenario 7 all USFWS lower 5 km samples have been removed from the spring sampling periods (lower 1200 spring trips have not been excluded).

The full dataset (Scenario 1) contains over 26,000 marked individuals since 1989 (Table 3). For all of the scenarios, the number of new marks released was relatively low in the

first two years (< 800 individuals per year) an increased substantially in the following three years (> 2,300 individuals per year, Table 3). During the mid 1990s there was relatively low effort and the number of new marks released ranged from 880 in 1994 to 60 in 1997. The number of recaptures during this same time period ranged from 1,620 individuals in 1994 to 135 in 1997. From 2000 to 2009, the number of new marks released averaged over 1000 individuals per year, and the number of recaptures averaged over 700 individuals per year. For the full data set (Scenario 1), a total of 26,671 tags have been released and 13,710 tags have been recaptured at least once per year between 1989 and 2009 (Table 3).

Table 3: Number of marks released and recaptured by year for each data scenario.

Marks released							
Scenario	1	2	3	4	5	6	7
1989	772	772	772	772	375	772	772
1990	716	716	716	716	461	716	716
1991	3,752	3,752	3,752	3,752	3,548	3,752	3,752
1992	2,358	2,358	2,358	2,358	2,330	2,358	2,358
1993	2,837	2,837	2,837	2,837	2,760	2,837	2,837
1994	880	880	880	880	834	880	880
1995	340	340	340	340	317	340	340
1996	60	60	60	60	20	60	60
1997	60	60	60	60	38	60	60
1998	213	213	213	213	213	213	213
1999	217	217	217	217	217	217	217
2000	696	481	553	644	696	696	434
2001	1,131	726	940	1,013	1,131	1,064	105
2002	1,105	748	886	1,075	1,100	1,013	109
2003	1,013	665	935	782	985	934	184
2004	836	539	784	626	811	777	247
2005	1,393	898	1,032	1,310	1,313	1,343	597
2006	1,439	1,250	1,366	1,339	1,418	1,364	722
2007	2,372	1,756	2,005	2,192	2,334	2,203	592
2008	2,055	1,493	1,873	1,738	1,974	1,951	519
2009	2,426	1,818	2,215	2,065	2,293	2,229	536
Total	26,671	22,579	24,794	24,989	25,168	25,779	16,250

Marks recaptured							
Scenario	1	2	3	4	5	6	7
1990	18	18	18	18	13	18	18
1991	225	225	225	225	222	225	225
1992	1,566	1,566	1,566	1,566	1,566	1,566	1,566

Continued...

Scenario	1	2	3	4	5	6	7
1993	1,753	1,753	1,753	1,753	1,753	1,753	1,753
1994	1,620	1,620	1,620	1,620	1,616	1,620	1,620
1995	906	906	906	906	905	906	906
1996	109	109	109	109	107	109	109
1997	135	135	135	135	133	135	135
1998	102	102	102	102	102	102	102
1999	116	116	116	116	116	116	116
2000	135	135	135	135	135	135	135
2001	475	461	472	475	475	475	414
2002	652	639	649	651	652	651	435
2003	565	536	565	564	565	561	327
2004	502	486	502	497	500	500	286
2005	511	480	508	508	510	510	346
2006	648	640	646	648	648	641	417
2007	888	875	883	888	887	875	625
2008	1,268	1,221	1,259	1,260	1,266	1,262	716
2009	1,516	1,434	1,503	1,498	1,512	1,503	712
Total	13,710	13,457	13,672	13,674	13,683	13,663	10,963

In Scenario 2, all the records from the USFWS trips between September and October have been removed. The filtering of the data reduces the number of unique marks released to 22,579 individuals and the number of recaptures to 13,457 (Table 3). Of the 26,671 tags that have been released, 5,377 and 6,299 were released in the months of September and October, respectively (Table 4). On average, the month of October has the highest percentage of newly marked animals (58.5%, Table 5) followed by the month of March at 57.4%. For Scenarios 3-6 the number of new marks released and recaptured are on the order of 25,000 and 13,670, respectively. Scenario 7, which excludes all samples collected by the USFWS in the lower 5km during the spring sampling periods, results in the largest reduction in the total number of new marks and recaptures. A total of 16,250 new marks were released and 10,963 recaptures were retained in the scenario 7 data set. The largest number of records in the database occur during the spring sampling trips, with a total of 10,850 records in April and 12,753 records in May (Table 4).

2.3 Simulation model

An individual based model was used to simulate the sampling process of capturing fish, tagging and recapturing individuals. Input variables for the simulation model included a time series of age-2 recruits and annual capture probabilities. The simulation model is summarized in Table 6. The simulation model was conditioned on the full historical data set (Scenario 1) between 1989 and 2009. This was done by first running ASMR in the estimation mode and obtaining maximum likelihood estimates of model parameters. Next,

Table 4: Total number of records in the database (new marks and recaptures) by month and year. Note that the numbers here include multiple recaptures of the same individual in the same year.

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
1989	0	0	0	0	876	0	0	0	0	0	0	0	876
1990	0	0	0	401	121	0	0	0	0	43	42	0	607
1991	79	3	135	9	285	394	1623	1050	536	291	200	175	4780
1992	168	340	780	1293	491	1179	369	145	144	290	266	0	5465
1993	117	152	1130	753	1325	815	946	1080	269	226	249	239	7301
1994	153	201	296	655	705	410	198	152	152	129	128	55	3234
1995	226	231	383	914	476	83	0	0	2	0	0	0	2315
1996	0	2	20	96	47	6	0	0	27	0	0	0	198
1997	0	0	7	42	114	18	0	0	32	0	0	0	213
1998	0	0	1	230	42	36	39	64	5	18	0	0	435
1999	8	0	0	181	43	18	0	0	41	35	29	0	355
2000	20	0	0	418	18	205	0	37	19	233	111	13	1074
2001	0	0	1	29	390	680	9	147	67	787	615	0	2725
2002	0	4	0	725	944	0	19	0	782	490	0	0	2964
2003	16	16	47	597	433	0	55	13	267	553	0	0	1997
2004	18	25	51	536	326	22	27	15	165	484	32	0	1701
2005	23	12	43	400	156	274	93	19	796	333	0	0	2149
2006	24	35	160	1008	671	492	158	55	269	317	0	0	3189
2007	0	0	2	876	1423	1086	0	0	805	569	0	0	4761
2008	0	5	3	846	1646	324	452	0	527	765	0	0	4568
2009	0	0	5	841	2221	1308	0	0	472	736	0	0	5583
Total	852	1026	3064	10850	12753	7350	3988	2777	5377	6299	1672	482	56490

project forward 10 years with known age-2 recruits (R_t) and for each individual at each time step determine if the individual is captured with probability $p_t v_a$. If the individual is not tagged, then it is tagged and released. Here we assume there is an equal probability of capturing an untagged or previously tagged fish and there are no tag related mortality effects. We then determine if the individual survives to the next year using a binomial probability based on the adult natural mortality rate M and the length of the individual relative to the mean asymptotic length (see Lorenzen, 2000).

In addition to new recruits, the simulation model also propagates forward previously marked and unmarked fish that existed in 2009. For the unmarked individuals, each cohort undergoes the same two random binomial draws to determine capture or recapture at each time step and survival to the next time step. For previously marked individuals in 2009, each cohort undergoes the binomial draw for recapture and survival. The true numbers of age-2 recruits and age-4+ (adult) abundance is also saved for later comparison with the estimated abundance using ASMR.

Source code and data for the AD Model Builder programs used in this analysis is available upon request. Contact Steve Martell at s.martell@fisheries.ubc.ca. For the sake

of documenting the code, the template file for the ADMB program and the necessary C++ library files are also provided in Appendix B. In addition, an example of the control file and input data file for Scenario 1 are also included in the appendix.

3 Results

3.1 Reproducing results based on code from Coggins and Walters

I obtained the ADModel builder code used in the previous assessment (Coggins, 2008); this code was modified to accommodate the objectives of this report. Before proceeding, I first compared the assessment results based on the original ASMR code (provided by Lew Coggins) and the modified code applied to the data based on Scenario 1 (full data set including 2009 data). There are no differences between the estimates of age-4+ adults, age-2 recruits and age-specific capture probabilities based on the original code and the modified code for this project (Fig. 1).

I also conducted a retrospective analysis using the Coggins (2008) code and the modified code using data from the year 2000 to 2009 (Fig. 2). Identical retrospective results were obtained for both estimates of adult and age-2 recruits. I was not able to reproduce the results that were published in Coggins (2008) because estimates of abundance are based on the inclusion of the effects of aging error in the ASMR assessment. Aging error was incorporated into the original assessment using a Monte Carlo procedure where 1,000 datasets were created by selecting age from length using a multinomial probability density function (Coggins, 2008).

3.2 Maximum likelihood estimates of abundance

For all scenarios that use only a portion of the total available data, estimates of adult humpback chub are slightly less in comparison to the full data set (Fig. 3). Scenarios 2 and 7 (i.e., no fall sampling and no spring USFWS data from the lower 5 km, respectively) result in the lowest estimates of adult abundance beyond the year 2000. Furthermore, recent trends in abundance are markedly different for scenario 7, where estimates of adult abundance is roughly 3,400 individuals since 2004. Maximum likelihood estimates of abundance and the corresponding 95% confidence interval (Table 7) suggests no real significant difference in abundance between scenarios 3,4,5, and 6; scenario 5 is the closest estimate of recent abundance in comparison to the full data set.

Similar trends in age-2 recruits were also observed for each of the seven data scenarios (Fig. 4). Again, scenario 5 best approximates the full data set in terms of trends and absolute number of age-2 recruits. Scenarios 2 and 7 have markedly lower estimates of age-2 recruits in recent years. Estimates of uncertainty for age-2 recruits are incredibly precise in 1989 with roughly 60 individuals spanning the 95% confidence interval (Table 8). This level of precision owes to the backwards reconstruction of the numbers-at-age

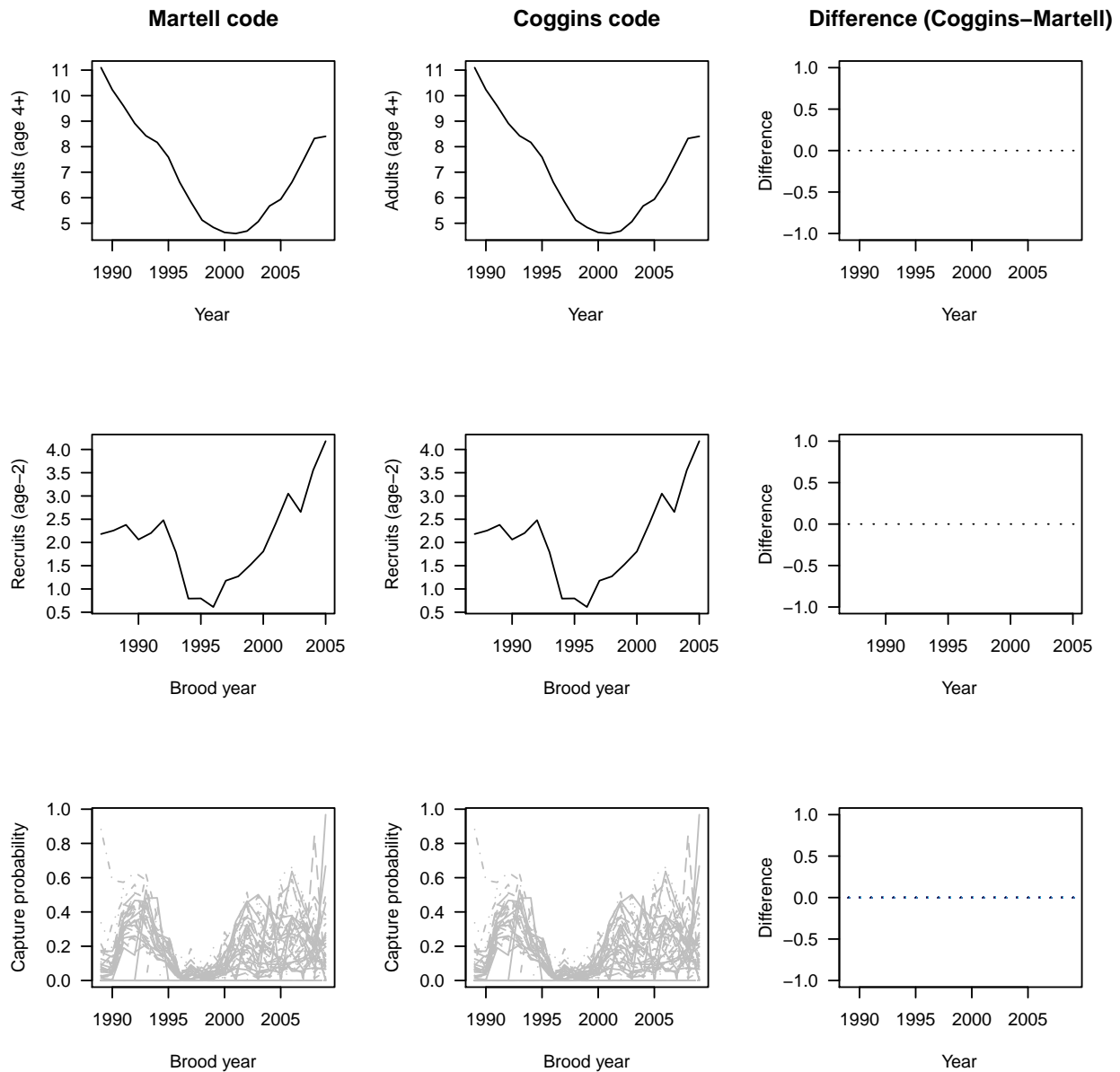


Figure 1: Comparing estimates of adult abundance (age-4+), recruits (age-2) versus brood year, and age-specific capture probabilities using the original code used in Coggins (2008) and the modified code in this assessment. There is absolutely no difference in the retrospective estimates of age-2 recruits or adult abundance between the two assessment models.

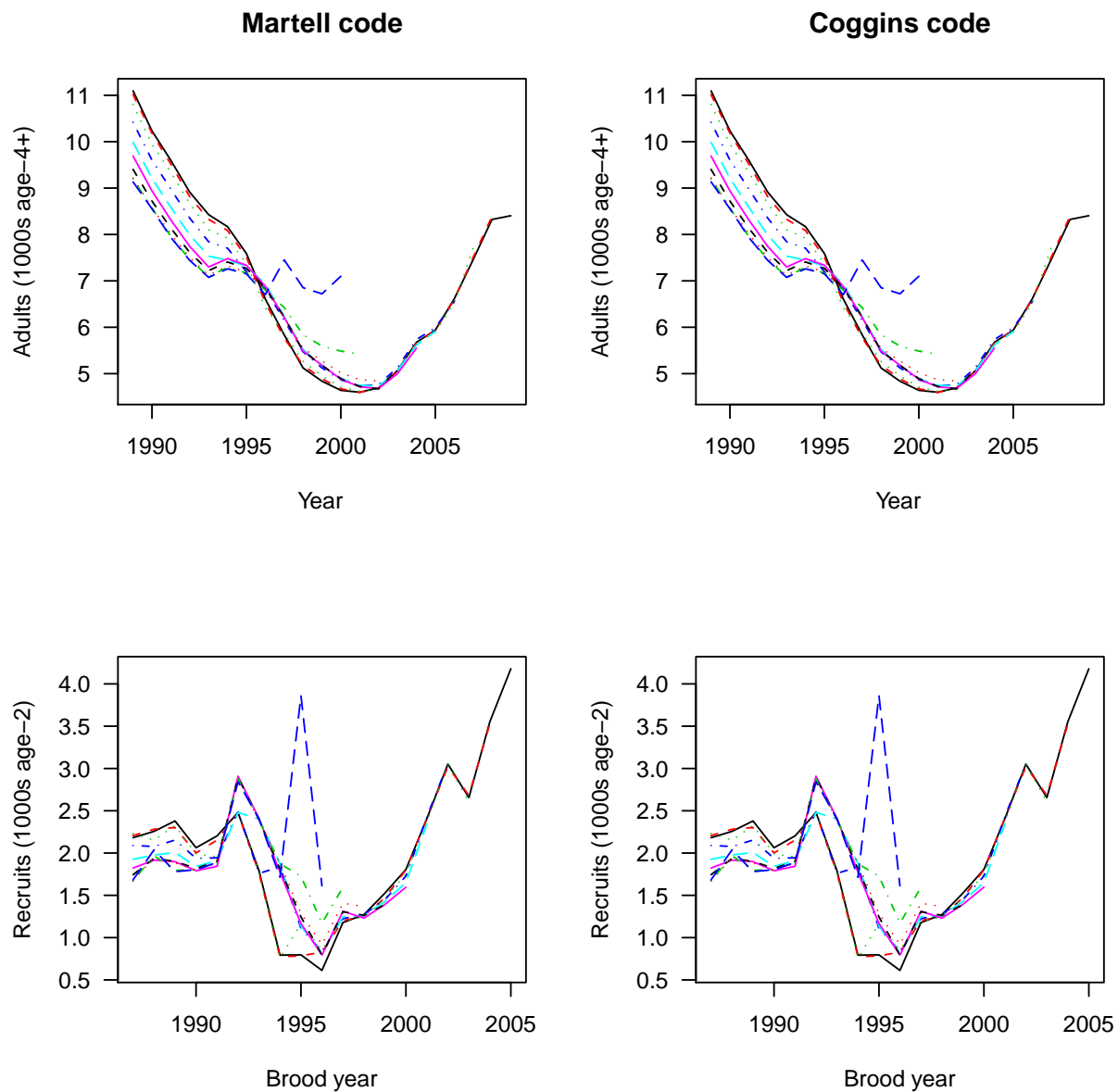


Figure 2: Comparing retrospective estimates of adult abundance (age-4+) and recruits versus brood year (age-2) using the original code used in Coggins (2008) and the modified code in this assessment.

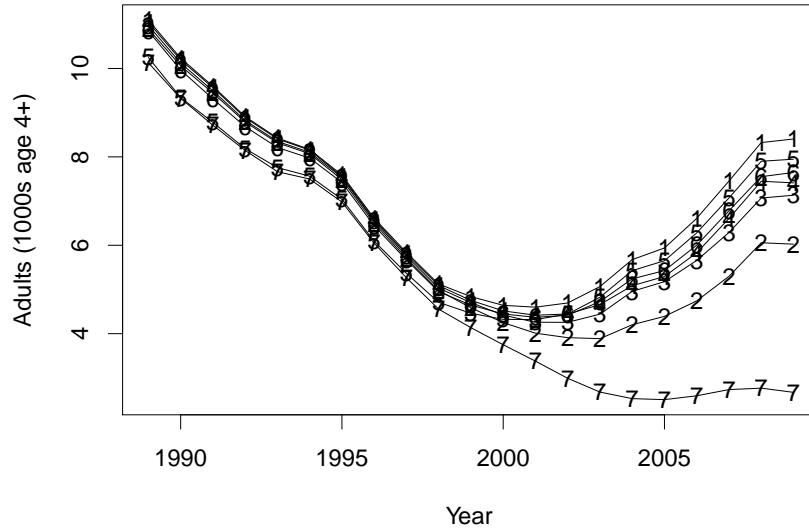


Figure 3: Maximum likelihood estimates of adult humpback chub (age-4+) using the 7 alternative data sets.

by adding in the previously marked fish and dividing by an assumed fixed natural mortality rate all while assuming no aging error in the conversion of lengths to age. Estimates of age-2 recruits for the near terminal year (2008) are much less precise because these cohorts have just recently recruited to the sampling gear and it takes several years of observing the same cohort (at least 4) before the strength of the year class can be determined reliably.

In general, it is somewhat expected that the alternative data scenarios would always lead to lower population estimates because the alternative scenarios have fewer individuals that have been both tagged and recaptured (see sample sizes in Table 3). These subsamples of all the available data indicate there is some non-random sampling occurring, in that there are spatial and temporal differences in capture probabilities. Ideally, under a purely random sample, the expected trends should be nearly identical but have less precision owing to the smaller sample sizes.

3.3 Recent estimates of age-specific capture probability

For each of the seven data scenarios, estimates of the relative age-specific selectivity were calculated based on the average age-specific selectivity between 2000 and 2009. For the simulation model, age-specific selectivity coefficients were approximated by using a Lowess smoother with a smoother span of 1/4 (R Development Core Team, 2009). The

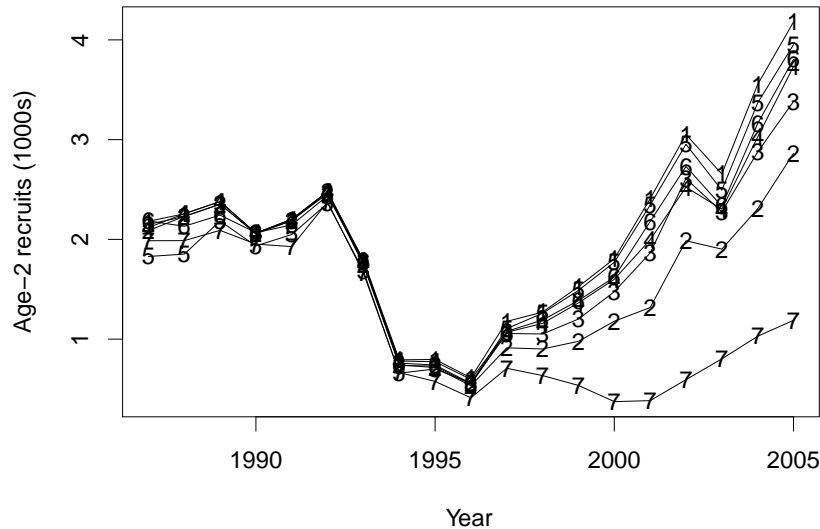


Figure 4: Maximum likelihood estimates of humpback chub age-2 recruits using the 7 alternative data sets.

resulting estimates of age-specific capture probabilities are shown in Fig. 5. Relative to the full data set in Scenario 1, Scenario 2 has a much lower probability of capturing larger, older fish, and Scenario 7 has a higher probability of capturing smaller fish. All other scenarios have similar age-specific capture probabilities to the full data set in Scenario 1 (Fig. 5).

The relative age-specific capture probabilities shown in Fig. 5 were used in simulating future data under each of the alternative sampling regimes. For example, if the maximum capture probability is 0.25 for future years, then age-2 fish would have a one in four chance of being sampled in any given year and an age-7 fish would have a one in eight chance of being sampled. An age-50 fish would have a very low chance of being sampled.

3.4 Estimates of future age-2 recruits using ASMR

Precision and bias in the estimates of future age-2 recruitment is a function of the capture probability and trends in future recruitment (Fig. 6). Increases in capture probability result in reduced uncertainty and bias in the estimates of age-2 recruits; however, there are diminishing returns in terms of the inner-quartile range in the estimate of age-2 recruits as capture probabilities increase from 0.15 to 0.25 (Fig. 6). It should also be noted, that the range of age-2 recruits shown in Fig. 6 is very conservative in that ageing errors were not considered in this assessment. We would still see reductions in the overall estimates

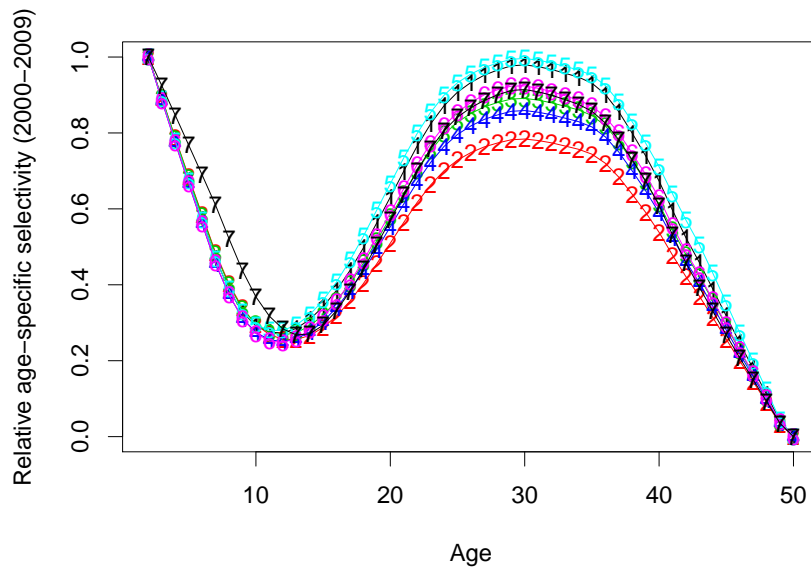


Figure 5: Smoothed estimates of relative age-specific selectivities for humpback chub based on mark-recapture data from 2000-2009 and each of the 7 data scenarios.

of uncertainty in age-2 recruits if ageing errors were considered in this analysis.

In either case where recruitment increases or decreases, the ability to detect changes in recruitment is markedly improved with higher capture probability (Fig. 6). In other words, there is less bias in the estimates of age-2 recruits with increased sample sizes. Also, there is some bias associated with rapid changes in recruitment that is primarily associated with aging errors and the minimum size limit of 150mm for tagging humpback chub.

Comparison across the different scenarios involving the slight changes in capture probabilities highlighted in Fig. 5 demonstrate that there is little difference in the estimates of age-2 recruits (Fig. 7). From 2009 to 2015, the inner quartile range of age-2 recruits across all seven scenarios are virtually identical. In fact, the minor differences in the estimated age-specific capture probabilities for each scenario have little effect on estimates of age-2 recruits because the highest capture probability for each scenario is actually age-2. Similar results were also obtained with lower capture probabilities, however the inner quartile ranges were much larger owing to the smaller sample sizes and greater uncertainty in estimates of age-2 recruits.

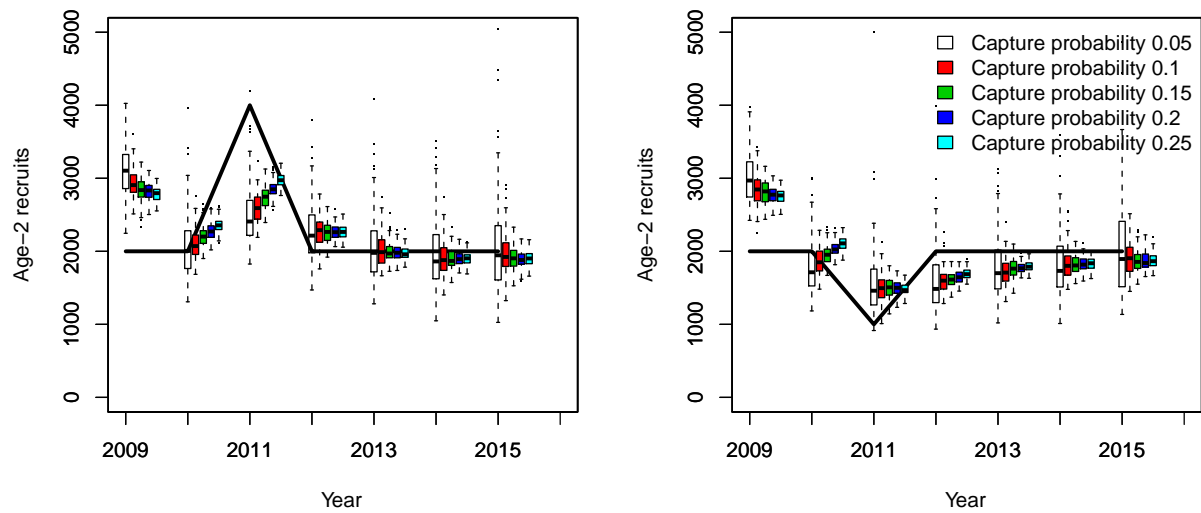


Figure 6: Distribution of age-2 recruits from 100 Monte Carlo trials using the ASMR model with age-specific capture probabilities for Scenario 1 (status quo) versus capture probabilities. True age-2 recruits used to simulate the data are shown by the solid line with a doubling of recruitment in 2011 (left panel) and a 50% reduction in recruitment in 2011 (right panel).

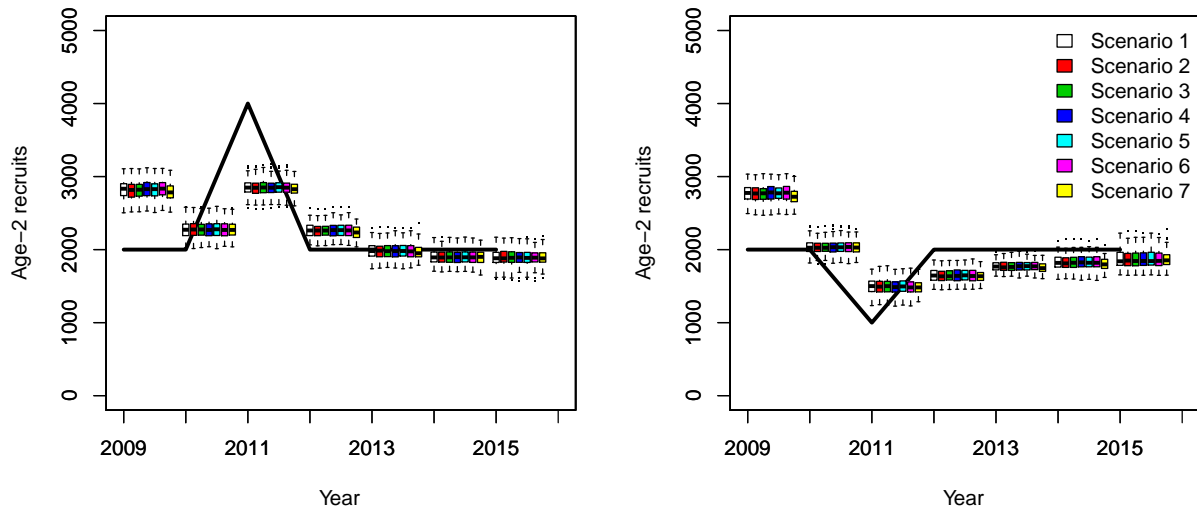


Figure 7: Distribution of age-2 recruits from 100 Monte Carlo trials using the ASMR model with age-specific capture probabilities for all 7 scenarios with the capture probability equal to 0.2. True age-2 recruits used to simulate the data is shown by the solid line with a doubling of recruitment in 2011 (left panel) and a 50% reduction in recruitment in 2011 (right panel).

4 Discussion

The overarching objective of this study was to examine how estimates of abundance and uncertainty in the estimates of humpback chub (HBC) would change if monitoring efforts were reduced. To examine the effect of a reduction in sampling, the historical data for HBC was filtered to remove spatial and or temporal records and then applied to the assessment model (ASMR) that is currently in use to assess the status and abundance of HBC. A total of seven different data scenarios were constructed, where the first scenario represents the status quo and utilizes all the available tagging information for HBC. A simulation model was also developed to generate future data over a range of average capture probabilities with age-specific selectivities that approximate the historical data. There are several limitations to the ASMR model as well as the simulation model used in this study. Overall, aging errors and the age-length keys used to compile the data are critical in establishing estimates of uncertainty as well as estimates of age-specific capture probabilities. Subsets of the historical data indicate non-random sampling is occurring. For example, omitting seasonal components or spatial components results in different estimates of the current status of the stock. Omitting all fall sampling and all spring sampling in the lower 5 km (Scenarios 2 and 7, respectively) results in the largest changes in the estimates of age-2 recruits and adult abundance. Scenario 5 (removal of the lower 1200

records only) gives rise to the least biased estimate of abundance in comparison to the full data set.

One of the major findings in this analysis and previous work by Coggins (2008) is that estimates of uncertainty are extremely sensitive to aging errors. I did not repeat the bootstrapping efforts of Coggins (2008) to better approximate the aging errors, and note here that estimated trends and maximum likelihood estimates are unaffected by the aging errors. Aging errors are also likely contributing to poor estimates of age-2 recruitment, where lags are introduced because of the asymmetric rounding down of small fish due to the minimum size limit of 150 mm. Aging errors are also largely the source of retrospective bias reported in Coggins (2008). The current version of ASMR is also an observation error only model and the major assumption in this model is that the catch-at-age is known without error. The observation error only model will also tend to underestimate the total uncertainty in the population estimates.

Perhaps a more honest representation of total error would be to use a mixed error model where annual age-2 recruits and the initial age-structure are treated as unknowns and estimated given the data. In contrast, ASMR estimates the terminal numbers-at-age and back calculates the number of unmarked animals by dividing the current estimate by an age-specific survival rate and adding the catch of newly marked animals (see Eqn. T1.7). The problem with this approach is that errors are inflated as they propagate backwards in time because of the repeated division of a number less than one. In contrast to a synthesis type model (forward propagation), aging errors become smaller as they propagate forward in time. Furthermore, it is not necessary to convert the mark-recapture data into age-specific mark-recapture information. Instead, the predicted numbers-at-age could be converted into predicted numbers-at-length using an integrated age-length key; where it would also be possible to jointly estimate the growth parameters (e.g., Kristensen et al., 2006). Such an integrated model would admit much more of the uncertainty associated with age-2 recruits and growth, and may be less biased because it would not assume that the catch-at-age is known without error.

Another issue with this assessment is that the simulation model used for future data collection uses fixed age-specific selectivities. In reality, there was a fair amount of inter-annual variability in these estimates that would also introduce additional uncertainty. The relative age-specific selectivities were based only on the recent estimates of age-specific capture probabilities (2000-2009 data). It is my understanding that the sampling gear used has changed very little during this time period (Coggins and Walters, 2009) and this was the justification for selecting this time period. There were only minor differences in the relative age-specific selectivities associated with the seven different data scenarios. It appears that there is a higher capture probability for smaller/younger HBC in the LRC region; when data from the lower 5km is omitted, there is a slightly higher capture probability for fish < age-10. Also, omitting all fall samples (Scenario 2) is likely to result in capturing fewer older individuals.

Another concern about changing the sampling protocol is the apparent non-random sampling. In theory, if tagged and untagged individuals all have the same probability

of capture, then reducing the number of samples in the data set should not result in a substantial change the estimates of abundance and stock status; there should only be an increase in the variance of the abundance estimate associated with the reduction in sample size. Fall sampling appears to be important time of year for putting out new marks, especially young animals. Removing the September data only (Scenario 3) results in slightly more biased estimates than October data (Scenario 4). Therefore, if one of the fall sampling trips had to be removed, September would be the preferred option based on the data collected thus far. The non-random sampling also begs the question, how would the estimated differ if additional sampling (spatial or temporal) was conducted?

Omitting spring samples from the lower 5 km resulted in the largest change in the estimates of abundance and depletion levels relative to that in 1989. In this case, there is a minor increase in the capture probability for smaller fish and this is interpreted by ASMR as fewer age-2 recruits after 1995 and a continued decline in age-4+ (adult fish) to a relatively stable abundance level near 3,400 individuals from 2004 to present. Implementing this scenario for future sampling programs will likely result in a downward bias in estimates of HBC.

Lastly, Coggins and Walters (2009) reported a large retrospective bias in the estimates of recruits and adult abundance using ASMR. Similar retrospective biases were not observed in this study. After much consternation, it was pointed out that the estimates in Coggins and Walters (2009) were based on the bootstrap replicates and this was the source of the retrospective bias (CJ Walters, pers. comm.).

In summary, spatial or temporal reductions in sampling effort are likely to result in minor changes in age-specific capture probabilities that have the potential to bias future estimates of recruits and adult abundance due to non-random sampling. Continued fall sampling (in at least 1 month –October –preferably both Sept. and Oct.) and the spring samples from the lower 5km are very important to maintain consistency with historical data. Absolute uncertainty in the estimates of abundance are largely linked to assumptions about aging error and model structure. Decreases in sampling effort that lead to average capture probabilities of less than 0.1 results in minor increases in uncertainty in age-2 recruits, but capture probabilities of less than 0.05 result in substantial increases in uncertainty. Estimates of variance decrease exponentially with increases in capture probabilities. The use of an integrated mixed error model should be considered to better reflect the true uncertainty and avoid the assumption of known ages in age-structured mark-recapture analysis.

5 Acknowledgements

I appreciate all the assistance and discussions with Lew Coggins, Carl Walters, Bill Pine and Robert Ahrens. The original prototype code for the IBM simulation model was written by Carl Walters in Visual Basic, and was adapted into C++ for this application. Thanks Carl. All data was provided by Paul Alley, thank you very much for your assistance in

assembling these data sets.

Table 5: Percentage of number of records that are new marks released by year and month. The mark rate column is the average mark rate calculated as proportion of the total number of fish caught that were previously marked.

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Mark rate
1989					88.1								11.9
1990				91.3	67.8					97.7	90.5		13.2
1991	91.1	66.7	84.4	100.0	86.0	83.8	74.6	57.9	53.9	38.1	46.5	35.4	31.8
1992	57.1	62.4	62.4	46.4	46.4	41.6	37.9	33.8	41.7	59.3	42.5		51.7
1993	25.6	34.2	37.4	34.4	50.0	52.0	50.1	45.5	34.2	39.8	36.9	25.9	61.1
1994	37.9	25.9	27.0	30.1	25.8	31.0	25.3	30.3	15.8	26.4	33.6	23.6	72.3
1995	20.8	15.2	21.4	21.7	28.4	3.6			50.0				77.0
1996		0.0	15.0	31.2	34.0	16.7			48.1				75.8
1997			0.0	52.4	23.7	16.7			21.9				77.1
1998			100.0	36.1	40.5	44.4	94.9	57.8	60.0	83.3			35.4
1999	62.5			55.8	30.2	33.3			82.9	94.3	79.3		37.4
2000	35.0			75.8	61.1	41.0		32.4	31.6	82.0	64.0	23.1	50.4
2001			100.0	27.6	40.8	54.7	11.1	28.6	44.8	46.9	24.4		57.9
2002		0.0		28.8	38.3		42.1		55.9	8.8			71.0
2003	50.0	75.0	55.3	51.4	40.2		29.1	69.2	61.0	56.4			45.8
2004	27.8	28.0	39.2	37.5	32.5	31.8	48.1	40.0	55.8	69.8	71.9		56.1
2005	65.2	91.7	55.8	49.8	44.2	43.8	51.6	84.2	75.0	56.5			38.2
2006	79.2	57.1	73.1	38.8	53.4	38.8	38.0	85.5	59.9	72.2			40.4
2007			100.0	57.2	55.4	17.4			72.5	52.0			40.9
2008		20.0	66.7	45.0	40.2	6.5	51.5		56.2	61.3			56.6
2009			80.0	45.9	49.2	25.9			52.3	49.0			49.6
Average	50.2	39.7	57.4	47.9	46.5	34.3	46.2	51.4	51.2	58.5	54.4	27.0	50.1

Table 6: Simulation model used for generating mark-recapture data.

<u>Input parameters</u>	<u>Description</u>
R_t	age-2 recruits
p_t	capture probability
l_a	mean length-at-age
$CV_l = 0.1$	coefficient of variation in length-at-age
$\sigma_m = 3.0$	standard deviation in measurement error
<u>Individuals</u>	
$\bar{l}_j \sim N(l_a, l_a CV_l)$	true length-at-age
$\hat{l}_j \sim N(\bar{l}_j, \sigma_m)$	measured length-at-age
$p_j \sim Binom(p_t v_a)$	age-based capture probability
$s_j \sim Binom(exp(-ML_\infty/l_a))$	survival probability

Table 7: Estimates of adult abundance (ages 4 and older) and the 95% confidence interval based on the inverse Hessian for 1989 and 2009, and the corresponding depletion level in 2009 relative to 1989, for the seven alternative data scenarios.

Scenario	Adult Abundance (age 4+)						Depletion	2.50%	97.50%
	1989	2.50%	97.50%	2009	2.50%	97.50%			
1	11,245	11,167	11,324	8,744	8,410	9,091	77.76%	75.31%	80.28%
2	11,032	10,957	11,108	6,290	6,052	6,537	57.01%	55.23%	58.85%
3	11,120	11,044	11,197	7,432	7,151	7,723	66.83%	64.76%	68.97%
4	11,189	11,112	11,267	7,682	7,410	7,964	68.65%	66.68%	70.68%
5	10,368	10,304	10,433	8,244	7,938	8,562	79.51%	77.03%	82.07%
6	11,010	10,932	11,090	7,955	7,653	8,268	72.25%	70.00%	74.56%
7	10,299	10,219	10,379	3,396	2,707	4,261	32.98%	26.49%	41.05%

Table 8: Estimates of age-2 recruits and the 95% confidence interval based on the inverse Hessian for 1989 and 2008.

Scenario	Age-2 recruits					
	1989	2.50%	97.50%	2008	2.50%	97.50%
1	2,436	2,407	2,466	2,733	0	3,203
2	2,406	2,378	2,435	1,165	0	1,401
3	2,408	2,379	2,437	2,320	0	2,729
4	2,435	2,406	2,465	1,577	0	1,863
5	2,229	2,205	2,254	2,624	0	3,073
6	2,306	2,276	2,336	2,660	0	3,117
7	2,155	2,124	2,185	702	0	1,273

References

- Coggins, J. L. and Walters, C. (2009). Abundance Trends and Status of the Little Colorado River Population of Humpback Chub: An Update Considering Data From 1989-2008. *US Geological Survey Open-File Report*, page 1075.
- Coggins, L. G. J. (2008). Abundance trends and status of the little colorado river population of humpbackchub; an update considering 1989-2006 data. *U.S. Geological Survey Open File Report*, 2007(1402):53 p.
- Coggins, L. G. J., Pine, W. E. I., Walters, C. J., and Martell, S. (2006). Age-structured mark-recapture analysis: A virtual-populatoin-analysis-based model for analyzing age-structured capture-recapture data. *North American Journal of Fisheries Management*, 26:201–205.
- Kristensen, K., Lewy, P., and Beyer, J. (2006). How to validate a length-based model of single-species fish stock dynamics. *Canadian Journal of Fisheries and Aquatic Sciences*, 63(11):2531–2542.
- Lorenzen, K. (2000). Allometry of natural mortality as a basis for assessing optimal release size in fish-stocking programmes. *Canadian Journal of Fisheries and Aquatic Sciences*, 57(12):2374–2381.
- Otter Research (2008). *An introduction to AD Model Builder for use in nonlinear modeling and statistics*. Otter Research Ltd., Nanaimo, B.C.
- R Development Core Team (2009). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

A Setting up ASMR input files

The mark-recapture data was provided by GCMRC staff (Paul Alley) as a flat file with 6 fields spanning 1989 to 2009 (Table 9). The data was provided on December 14, 2009. An R-script was used to manipulate the data into 3 separate tables representing the number of new marks at age in each year, the number of recaptures-at-age in each year, and the number of recaptures-at-age by tag year.

The following is the R-code that was developed to construct the input data file for ASMR given the database query as shown in Table 9.

```
setwd("~/Documents/CONSULTING/HBC ASMR Performance/HBC Data")

if(!exists("HBCdata"))
```

Table 9: First 6 rows of the default data provided by GCMRC staff.

	FISHNO		DATES		RM RIV	TL	TAGNO
1	148259	5/7/1989	6:45:00.000	PM	[NULL]	LCR 384	7040
2	147290	5/3/1989	9:30:00.000	AM	[NULL]	LCR 367	13306
3	147481	5/5/1989	12:00:00.000	AM	[NULL]	LCR 405	13307
4	301373	4/17/1994	10:15:00.000	AM	[NULL]	LCR 426	13308
5	147621	5/5/1989	10:30:00.000	AM	[NULL]	LCR 433	13308
6	171292	6/10/1991	4:50:00.000	PM	[NULL]	LCR 440	13308

```
{
HBCdata = read.csv("2009HBCdata.csv", header=T)
df=HBCdata
print(paste("Total number of records =",dim(df)[1]))

#Fix the Date column to proper m/d/Y format
x=as.character(df$DATES)
x=as.POSIXct(strptime(x,"%m/%d/%Y"))
df$DATES=x
}

nyr=length(1989:2009)
mta=matrix(0, nrow=nyr, ncol=49)
rta=matrix(0, nrow=nyr, ncol=49)
rcta=array(0, c(nyr, nyr, 49))

fn<-function(tagno)
{
#Subset the data frame for tagno
sdf=cbind(subset(df, df$TAGNO == tagno),
RECAP=F, YR=factor(NA, levels=1989:2009), AGE=NA)
sdf=sdf[order(sdf$TAGNO, sdf$DATE), ]

#Get year for each record
sdf$YR=getyr(sdf$DATE, sdf)

#Get age at time of tagging
sdf$AGE[1] = getage(sdf$TL[1])
iage=sdf$YR-sdf$YR[1]
sdf$AGE=sdf$AGE[1]+iage
}
```

```

#Recaptures & remove duplicates in same recapture YR
sdf$RECAP[-1]=T
sdf=sdf[!duplicated(sdf$YR), ]

#New marks released
mdf=sdf[sdf$RECAP==F, ]
mdf$AGE=factor(mdf$AGE, levels=2:50)
mdf$YR=factor(mdf$YR, levels=1989:2009)
mta<-mta+t(as.matrix(table(mdf[, 9:8])))

#Recaptured marks
rdf=sdf[sdf$RECAP==T, ]
rdf$AGE=factor(rdf$AGE, levels=2:50)
rdf$YR=factor(rdf$YR, levels=1989:2009)
r1 = t(as.matrix(table(rdf[, 9:8])))
rta<-rta+r1

#Recaptures by tag year
iyr=sdf$YR[1]-1989+1
rcta[iyr, , ]<-rcta[iyr, , ]+r1
}

getage=function(len)
{
  alast=30
  ## this function does new growth curve (LEW COGGINS pers comm.)
  ifelse(round(-1.05952E-08*len^4+1.25697E-05*len^3-0.005058694*len^2
+0.878738367*len^1-53.55077549)>alast,alast,
round(-1.05952E-08*len^4+1.25697E-05*len^3-0.005058694*len^2
+0.878738367*len^1-53.55077549))

  ifelse(round(-1.05952E-08*len^4+1.25697E-05*len^3-0.005058694*len^2
+0.878738367*len^1-53.55077549)==1,2,round(-1.05952E-08*len^4
+1.25697E-05*len^3-0.005058694*len^2
+0.878738367*len^1-53.55077549))
}

getyr=function(datee, df){
  ifelse(strptime(datee,"%Y")$year+1900==1992 |
strptime(datee,"%Y")$year+1900==1996 |

```

```

        strptime(datee,"%Y")$year+1900==2000 |
strptime(datee,"%Y")$year+1900==2004 |
strptime(datee,"%Y")$year+1900==2008,
        strptime(df$DATE-7862400,"%Y")$year+1900,
        strptime(df$DATE-7776000,"%Y")$year+1900)
}

#Write temporary output data file
write.data.file=function()
{
fn="HBCdata.output"
write("#mta",file=fn)
write.table(mta, file=fn, row.names=F, col.names=F, append=T)
write("#rta", file=fn, append=T)
write.table(rta, file=fn, row.names=F, col.names=F, append=T)
yrs=1989:2009; j=0
for(i in yrs)
{
j=j+1
write(paste("#rcta", i), file=fn, append=T)
write.table(rcta[j, , ], file=fn, row.names=F,
col.names=F, append=T)
}
write("#eof", file=fn, append=T)
write(999, file=fn, append=T)
}

## *****
##          MAIN
## *****
tag.ids=unique(df$TAGNO)

t1=Sys.time()
for(i in tag.ids) fn(i)
print(Sys.time()-t1)

write.data.file()
## ***END OF CODE***

```

B ADMB code & and example data file for this study

The following is the AD Model builder template file (ASMR.tpl) used in this project for simulating data as well as estimating the data from the seven alternative scenarios. In addition to this code, ASMR also requires a c++ library ASMR.cxx which is also included following this template code.

There are two required input data files for the ASMR program. For example, the S1.dat, and Scenario1.dat files represent the control file and input data for Scenario 1.

```

/*****
// Programmer: Steve Martell
// Project Name:
// Date:
// Version:
// Comments: For the sturgeon case, need to integrate the SRA with ASMR
// TO DO LIST FOR HBC PROJECT WORKSHOP:
// 1) Add aging error to simulation model (**watch array bounds in agetag**)
// 2) Extend simulation for 10 year projections with input Rt values (DONE)
// 3) Write code to produce simulated data file. (DONE)
// 4) Need to compare with ASMR3t (lews code) and repeat results
// 5)
// 6)
// **The plus group code in here has not been tested yet. (Ignored for this project)
/*****/
DATA_SECTION
//SM Code for adding a command line option to the program.
int sim; //flag for running the simulator
int rseed; //random number seed for generating fake data
int retyr; //number of retrospective years
number ptcap; //capture probability for simulating future data
int plus; //flag for using +group calculations (termage)
int scenario; //which data scenario to use
LOC_CALCS
int on,opt;
sim=0;
plus=0;
retyr=0;
rseed = 123; //default random number seed.
ptcap=0;
if((on = option_match(ad_comm::argc,ad_comm::argv,"-sim",opt))>-1)

    sim=1;
    rseed=atoi(ad_comm::argv[on+1]);
    scenario=atoi(ad_comm::argv[on+2]);
    ptcap=atof(ad_comm::argv[on+3]);
    //retyr=-6;
    ofstream ofs("seed.txt");
    ofs<<rseed<<endl;

if((on = option_match(ad_comm::argc,ad_comm::argv,"-retro",opt))>-1)

    retyr=atoi(ad_comm::argv[on+1]);
    sim=0; //turn off simulation model if doing retrospective analysis

if((on = option_match(ad_comm::argc,ad_comm::argv,"-scenario",opt))>-1)

    scenario=atoi(ad_comm::argv[on+1]); //scenario # for future data
    ptcap=atof(ad_comm::argv[on+2]);

END_CALCS

init_adstring data_file; //file name of data
init_adstring scenario_file; //file for alternative data scenario
init_int bycohort; //switch to allow by tagcohort consideration of recaptures
init_int model; //switch for model type ASMR1, ASMR2, ASMR3
init_int likelihood; //switch for poisson or negative binomial likelihood.
!! cout<<"-----"<<endl;
!! cout<<" -Model type = "<<model<<endl;
!! cout<<" -By cohort = "<<bycohort<<endl;
!! cout<<" -Likelihood: ";
!! if(likelihood==1) cout<<"Poisson"<<endl; else cout<<"Negative binomial"<<endl;
!! cout<<"-----"<<endl;

//-----
!! ad_comm::change_datafile_name(data_file);

```

```

//-----
//-----
!! cout<<"Data file name: "<<data_file<<endl;
init_int syr; //start year of data
init_int nnyr; //end year of data
int nyr; //end of simulation year (for retrospective analysis)
!! nyr = nnyr-retyr;
!! if(retyr) cout<<"Retrospective year = "<<nyr<<endl;
init_int sage; //youngest age
init_int nage; //oldest age
vector age(sage,nage);
!! age.fill_seqadd(sage,1);
!! cout<<"Age vector: "<<age(sage,sage+3)<<" ..."<<age(nage)<<endl;
ivector byr(syr,nyr-2); //brood years
!! byr.fill_seqadd(syr-2,1);

init_number vonbk; //von b k parameter
init_number linf; //Linfinity parameter
init_vector lenage(sage,nage); //length at age
init_int termage; //last age to estimate abundance in terminal year (ASMR 2 and 3)
!! cout<<"Terminal age = "<<termage<<endl;

init_int nva; //number of vulnerability blocks
init_ivector epics(1,nva); //ending year of each vulnerability epoch

init_int m_phz; //solution phaze for m-deviations
!! cout<<"Phase for estimating m deviations = "<<m_phz<<endl;

!! cout<<scenario_file<<endl;
!! if(!scenario_file == "NULL") ad_comm::change_datafile_name(scenario_file);

init_matrix i_mta(syr,nnyr,sage,nage); //Data, number of untagged fish captured by age and year
init_matrix i_rta(syr,nnyr,sage,nage); //Data, number of recaptured fish by age and year
//Data, number of tagged fish recaptured by age and year and tag cohort
init_3darray i_rcta(syr,nnyr,syr,nnyr,sage,nage);
!! cout<<i_rcta(nnyr-1)(nnyr)<<endl;

matrix mta(syr,nyr,sage,nage);
matrix rta(syr,nyr,sage,nage);
3darray rcta(syr,nyr,syr,nyr,sage,nage);
LOC_CALCS
int i,n_yr;
mta.initialize();
cout<<syr<<" "<<nnyr<<endl;
//some accounting here to deal with retrospective or projection.
if(retyr<=0)n_yr = nnyr; else n_yr=nyr;
for(i=syr;i<=n_yr;i++)

    mta(i)=i_mta(i);
    rta(i)=i_rta(i);
    for(int j=syr;j<=n_yr;j++)
        rcta(i)(j)=i_rcta(i)(j);

    epics(nva)=nyr;
    cout<<"Ok here?"<<endl<<endl;
END_CALCS

matrix epsilon(syr,nyr,sage,nage);
matrix delta(syr,nyr,sage,nage);
3darray Cdelta(syr,nyr,syr,nyr,sage,nage);
init_int eof;
!! if(eof!=999)cout<<"Error reading data file"<<endl; exit(1);
!! cout<<endl<<"_---*** END OF DATA FILE ***_---"<<endl<<eof<<endl;

int tau_phz;
int UT_phz;
int pterm_phz;
int ahat_phz;
int nf;

//-----
!! ad_comm::change_datafile_name("Simdata.dat");
//-----
//-----
//input data for simulation model.
init_vector sim_pt(syr,nnyr);
init_vector sim_rt(syr-10,nnyr);

//-----
!! ad_comm::change_datafile_name("Selectivity.dat");
//-----
//-----
//input selectivities for simulation model.
init_matrix sel_coeffs(sage,nage,1,7); //selectivity coefficients from 7 data scenarios
int nsim;
!! nsim=10; //number of simulation years into the future

```

```

matrix o_mta(syr,nyr+nsim,sage,nage);
matrix o_rta(syr,nyr+nsim,sage,nage);
3darray o_rcta(syr,nyr+nsim,syr,nyr+nsim,sage,nage);

//Classification matrix for aging errors
matrix Q(sage,nage,sage,nage);
LOC_CALCS
dvector ia(sage,nage);
dvector std(sage,nage);
//std = 0.1*age;
std = 0.2+(15-0.2)*plogis(age,15.,.3.);
ia.fill_seqadd(0,1);
//cout<<ia<<endl;
double z1,z2;
for(int i = sage;i<=nage;i++)
  for(int j=sage; j<=nage;j++)

    z1=(ia(i)+0.5 - ia(j))/std(i);
    z2=(ia(i)-0.5 - ia(j))/std(i);
    Q(i,j) = cumd_norm(z1)-cumd_norm(z2);

  Q(i)=Q(i)/sum(Q(i));

  //Q=inv(Q);
  Q=identity_matrix(sage,nage);
  //cout<<setprecision(2)<<Q<<endl;
  //exit(1);

END_CALCS

PARAMETER_SECTION
init_bounded_number m(0,1.0,-2);
!! if(model==1) pterm_phz=1; else pterm_phz=-1;
init_bounded_number pterm(0,1,pterm_phz);
!! if(likelihood==2) tau_phz=2; else tau_phz=-1;
init_bounded_number tau(0,500,tau_phz);
!! if(model==3) ahat_phz=-1; else ahat_phz=1;
init_bounded_vector ahat(1,nva,0,nage,ahat_phz);
init_bounded_vector tau_ahat(1,nva,0,50,ahat_phz);
init_bounded_vector g2(1,nva,0,1,-2);

!! if(model==2||model==3) UT_phz=1; else UT_phz=-1;
//init_bounded_vector log_UT(sage+1,termage,-10,10,UT_phz);
init_vector log_UT(sage+1,termage,UT_phz);
init_bounded_dev_vector m_dev(syr,nyr,-3.0,3.0,m_phz);

!! m= 0.13;
!! tau=25.0;
!! ahat = 3.0;
!! tau_ahat=2.0;
!! log_UT=5.0;
!! g2=0.01;

objective_function_value f;
sdreport_number sd_nt4_syr;
sdreport_number sd_nt4_nyr;
sdreport_number sd_rt_syr;
sdreport_number sd_rt_ppyr;

vector sa(sage,nage);
vector pt(syr,nyr); //capture probabilities

matrix va(syr,nyr,sage,nage);
matrix U(syr,nyr,sage,nage); //unmarked fish at large
matrix M(syr,nyr,sage,nage); //marked fish at large
matrix Mta(syr,nyr,sage,nage); //new marks deployed
matrix Rta(syr,nyr,sage,nage); //recaptures
matrix pta(syr,nyr,sage,nage); //age-specific capture probabilities

3darray Mc(syr,nyr,syr,nyr,sage,nage); //Unmarked cohorts vulnerable to capture
3darray Rcta(syr,nyr,syr,nyr,sage,nage); //Predicted recaptures by cohort.

PRELIMINARY_CALCS_SECTION
nf=0;
//call the data simulator from here.
//Note, you cannot use tabs in this section
if(sim)
  cout<<"-----" <<endl<<endl;
  cout<<" **Implementing Simulation--Estimation trial** " <<endl;
  cout<<"-----" <<endl;
  cout<<" Random Seed No.: " << rseed<<endl;
  cout<<"-----" <<endl<<endl;

  //Call the simulator (This appears to be working now.)
  //Still need to add aging error to simulation model

```



```

        //simulation_model();
        //exit(1);

PROCEDURE_SECTION
// *****

//          MAIN ROUTINES

// *****

get_survival();
get_vulnerabilities();
unmarked_and_marked_fish();
get_capture_probabilities();
calc_negloglike();
dvar_vector nt4=rowsum(trans(trans(U).sub(4,nage))+rowsum(trans(trans(M).sub(4,nage))));
dvar_vector rt=get_drt();
sd_nt4_syr = log(nt4(syr));
sd_nt4_nyr = log(nt4(nyr));
sd_rt_syr = log(rt(syr));
sd_rt_ppyr = log(rt(max(byr)));

if(mceval_phase()) write_mcmc_report();
// *****

// *****

FUNCTION get_survival
//to do: add mdevs
//survival rate ma=m*lnf/lenage
dvector t1 = lnf/lenage;
sa = mfexp(-m*t1);
//cout<<"sa"<<endl<<sa<<endl;
//exit(1);
// May 2, 2010. SM compared this routine with Coggins ASMR3t.tpl
// it gives the exact same results.

FUNCTION get_vulnerabilities
int i,j;
//vulnerabilities
j=1;
for(i=syr;i<=nyr;i++)

    if(i>epics(j)) j++;
    va(i)=plogis(age,ahat(j),tau_ahat(j));
    if(active(g2)) va(i)=eplogis(age,ahat(j),tau_ahat(j),g2(j));

FUNCTION unmarked_and_marked_fish
int i,j,k;
double tiny = 1.e-30;

//terminal abundances and VPA reconstruction
U.initialize();
U=tiny;
U(nyr,sage)=999.;

//Add catches to terminal age column
dvector d1 = column(mta,nage);
for(i=syr; i<=nyr;i++)U(i,nage)+=d1(i);

if(model==1) U(nyr) = elem_div((mta(nyr)+tiny),(pterm*va(nyr)));

if(model==2 || model==3) //ASMR 2
    U(nyr)(sage+1,termage)+=exp(log_UT);
    for(j=termage+1; j<=nage; j++)

        U(nyr,j) = U(nyr,j-1)*sa(j-1);

for(i=nyr-1; i>=syr; i--)

    for(j=nage-1; j>=sage; j--)

        U(i,j) = (U(i+1,j+1)/sa(j)) + mta(i,j);

//marked fish

```

```

if(!bycohort)
    //marked fish
    M.initialize();
    for(i=syr+1;i<=nyr;i++)

        M(i)(sage+1,nage) +=elem_prod(M(i-1)(sage,nage-1)+mta(i-1)(sage,nage-1),sa(sage,nage-1));

if(bycohort)

    Mc.initialize();
    for(k=syr;k<=nyr;k++) //loop over cohorts
        Mc(k)=tiny;
        for(i=k+1;i<=nyr;i++) //loop over cohort years
            for(j=sage+1;j<=nage;j++) //loop over ages

                if(i==k+1)Mc(k,i,j)= mta(i-1,j-1)*sa(j-1);
                else Mc(k,i,j)=Mc(k,i-1,j-1)*sa(j-1);

    M.initialize();
    M=tiny;
    for(k=syr;k<=nyr;k++)
        for(i=syr;i<=nyr;i++)
            for(j=sage;j<=nage;j++)
                M(i,j) += Mc(k,i,j);

    //cout<<M<<endl;
    //exit(1);

FUNCTION get_capture_probabilities
int i,j;
dvariable t1,t2;
Rta.initialize();
Mta.initialize();
Rcta.initialize();
dmatrix iQ = trans(trans(Q.sub(sage+1,nage)).sub(sage+1,nage));
if(model != 3)

    for(i=syr; i<=nyr;i++)

        t1=sum(mta(i)+rta(i));
        t2=sum(elem_prod(va(i),U(i)+M(i)));
        pt(i)=t1/t2;
        Mta(i) = elem_prod(U(i),va(i)*pt(i));
        Rta(i) = elem_prod(M(i),va(i)*pt(i));

        if(bycohort)

            for(j=syr;j<=nyr;j++)

                Rcta(i)(j)=elem_prod(Mc(i)(j),va(i)*pt(i));

if(model==3) //ASMR3

    dvar_vector tmp(sage,nage);
    for(i=syr; i<=nyr; i++)

        tmp = elem_div(mta(i)+rta(i),U(i)+M(i)); //MLE of capture probability
        //cout<<mta(i)+rta(i)<<endl;
        Mta(i) = elem_prod(U(i),tmp);
        Rta(i) = elem_prod(M(i),tmp);

        t1=sum(mta(i)+rta(i));
        t2=sum(U(i)+M(i));
        pt(i)=t1/t2;
        pta(i)=tmp;
        if(bycohort)

            for(j=syr;j<=nyr;j++)

                tmp = elem_div(mta(j)+rta(j),U(j)+M(j)); //MLE of capture probability
                Rcta(i)(j)=elem_prod(Mc(i)(j),tmp+1.e-30);

    //cout<<setprecision(2)<<endl<<Rcta(syr)<<endl;
    //exit(1);

```

```

//cout<<endl<<Mta<<endl<<endl;
FUNCTION calc_negloglike
/*
  There are two options here, 1: poisson, and 2: negative binomial
*/
int i,j;
dvar_vector lvec(1,4);
lvec.initialize();
epsilon.initialize();
delta.initialize();
Cdelta.initialize();

//Likelihood of the data given a poisson distribution
if(likelihood==1)

  for(i=syr; i<=nyr;i++)

    lvec(1)+=dpois_residual(mta(i)(sage,nage),Mta(i)(sage,nage),epsilon(i));

    if(!bycohort)
      lvec(2)+=dpois_residual(rta(i)(sage+1,nage),Rta(i)(sage+1,nage),delta(i));

    if(bycohort)

      for(j=syr;j<=nyr;j++)

        lvec(2)+=dpois_residual(rcta(i)(j)(sage+1,nage),Rcta(i)(j)(sage+1,nage),Cdelta(i)(j));

//Likelihood of the data given a negative binomial distribution.
if(likelihood==2)

  for(i=syr; i<=nyr;i++)

    lvec(1)+=dnbinom(mta(i)(sage,nage),Mta(i)(sage,nage),tau,epsilon(i));

    if(!bycohort)
      lvec(2)+=dnbinom(rta(i)(sage+1,nage),Rta(i)(sage+1,nage),tau,delta(i));

    if(bycohort)

      for(j=syr;j<=nyr;j++)

        lvec(2)+=dnbinom(rcta(i)(j)(sage+1,nage),Rcta(i)(j)(sage+1,nage),tau,Cdelta(i)(j));

//if(active(m)) lvec(3) = dlnorm(m,log(0.13-0.0005),0.1); //prior for adult m.
//lvec(4) = 0.1*sqrt(log_UT(sage)-log(mta(nyr,sage)/pt(nyr))); //use termial capture prob for age2 in terminal year
//cout<<"Likelihood " <<lvec<<endl;

f=sum(lvec);
nf++;
//exit(1);

TOP_OF_MAIN_SECTION
time(&start);
arrmblsize = 50000000;
gradient_structure::set_GRADSTACK_BUFFER_SIZE(1.e7);
gradient_structure::set_CMPDIF_BUFFER_SIZE(1.e7);
gradient_structure::set_MAX_NVAR_OFFSET(5000);
gradient_structure::set_NUM_DEPENDENT_VARIABLES(5000);

GLOBALS_SECTION
#include <admodel.h>
#include <time.h>
#include <ASMR.cxx>
time_t start,finish;
long hour,minute,second;
double elapsed_time;
#undef REPORT
#define REPORT(object) report << #object << endl << object << endl;

FINAL_SECTION
time(&finish);
elapsed_time=difftime(finish,start);
hour=long(elapsed_time)/3600;
minute=long(elapsed_time)%3600/60;

```

```

second=(long(elapsed_time)/3600)%60;
cout<<endl<<endl<<"*****"<<endl;
cout<<"--Start time: "<<ctime(&start)<<endl;
cout<<"--Finish time: "<<ctime(&finish)<<endl;
cout<<"--Runtime: ";
cout<<hour<<" hours, "<<minute<<" minutes, "<<second<<" seconds"<<endl;
cout<<"*****"<<endl;

if(!sim) write_trial();
if(sim)simulate_future_data();

FUNCTION write_trial
//write seed, pt, nyr, rt(2009,2015),
int seed;
ifstream ifs("seed.txt");
ifs>>seed;
ofstream ofs("Trials.txt",ios::app);
adstring tt=" ";
dvector rt=get_rt();
ofs<<scenario<<tt<<seed<<tt<<ptcap<<tt<<rt(2009,2015)<<endl;

REPORT_SECTION
cout<<"Report Section: Nyr is = "<<nyr<<endl;
report<<"M"<<endl<<m<<endl;
report<<"age"<<endl<<age<<endl;
dvector yr(syr,nyr);
yr.fill_seqadd(syr,1);
report<<"yr"<<endl<<yr<<endl;

report<<"va"<<endl<<va<<endl;
report<<"sa"<<endl<<sa<<endl;
report<<"pt"<<endl<<pt<<endl;
report<<"pta"<<endl<<pta<<endl;
report<<"epsilon"<<endl<<epsilon<<endl;
report<<"delta"<<endl<<delta<<endl;
report<<"mta"<<endl<<mta<<endl;
report<<"rta"<<endl<<rta<<endl;
report<<"nt2"<<endl<<rowsum(trans(trans(U).sub(2,nage)))+rowsum(trans(trans(M).sub(2,nage)))<<endl;
report<<"nt3"<<endl<<rowsum(trans(trans(U).sub(3,nage)))+rowsum(trans(trans(M).sub(3,nage)))<<endl;
report<<"nt4"<<endl<<rowsum(trans(trans(U).sub(4,nage)))+rowsum(trans(trans(M).sub(4,nage)))<<endl;
REPORT(byr);
report<<"rt"<<endl<<get_rt()<<endl<<endl;
report<<"Q"<<endl<<Q<<endl;

if(model==3) report<<"cdelta"<<endl<<Cdelta<<endl;

FUNCTION dvar_vector get_drt()
//back-calculated-recruitment
int pyr = min(byr);
int ppyr = max(byr);
dvar_vector rt(pyr,ppyr); rt.initialize();
rt(syr,ppyr)=(column(U,sage)(syr,ppyr));
dvar_vector lx(sage,nage); lx(sage)=1;
for(int i=sage;i<sage+2;i++)

    lx(i+1) = lx(i)*(sa(i));
    rt(syr-i+(sage-1)) = (U(syr,i+1))/lx(i+1);

return(rt);

FUNCTION dvector get_rt()
//back-calculated-recruitment
int pyr = min(byr);
int ppyr = max(byr);
dvector rt(pyr,ppyr); rt.initialize();
rt(syr,ppyr)=value(column(U,sage)(syr,ppyr));
dvector lx(sage,nage); lx(sage)=1;
for(int i=sage;i<sage+2;i++)

    lx(i+1) = lx(i)*value(sa(i));
    rt(syr-i+(sage-1)) = value(U(syr,i+1))/lx(i+1);

return(rt);

FUNCTION write_mcmc_report
if(nf==1)

    ofstream ofs("asmr.mcmc");
    ofs<<"m tau pterm "<<endl;
    ofstream of2("nt4.mcmc");
    ofstream of3("rt.mcmc");
    ofstream of4("pt.mcmc");

ofstream ofs("asmr.mcmc",ios::app);
ofs<<m<<" "<<tau<<" "
    <<pterm<<" "<<endl;

```

```

ofstream of2("nt4.mcmc",ios::app);
of2<<rowsum(trans(trans(U).sub(4,nage))+trans(trans(M).sub(4,nage)))<<endl;

ofstream of3("rt.mcmc",ios::app);
of3<<get_rt()<<endl;

ofstream of4("pt.mcmc",ios::app);
of4<<pt<<endl;

FUNCTION int length2age(const double& LenTag)
//SM Need to change this polynomial to the one that is used to convert length to age from
//the real data in the Rcode. DONE
//ifelse(round(-1.05952E-08*len^4+1.25697E-05*len^3-0.005058694*len^2
// +0.878738367*len-1-53.55077549)>alast,alast,

int age=(-1.05952E-08*pow(LenTag,4)+1.25697E-05*pow(LenTag,3)-0.005058694*pow(LenTag,2)+0.878738367*LenTag-53.55077549);
age<sage?age=sage:NULL;
return(age);
//return (-0.0000000019605 * pow(LenTag,4) + 0.0000019314 * pow(LenTag,3) - 0.00039319 * pow(LenTag,2) + 0.024165 * LenTag + 1.8451);
//return ceil(-0.0000000019605 * pow(LenTag,4) + 0.0000019314 * pow(LenTag,3) - 0.00039319 * pow(LenTag,2) + 0.024165 * LenTag + 1.8451);

//inverse of vonb growth, very approximate
/*int agetag = nage;
if(linf>LenTag) agetag = -1./vonbk * log(1.-LenTag/linf);
//cout<<agetag<<" "<<vonbk<<endl;
return (agetag);*/

FUNCTION void simulate_future_data()
/*
This function is used to concatenate future data (post 2009) to
the full data set (Scenario 1) using an IBM model where the MLE
values from S1 are used to condition this simulation model. Capture
probabilities will differ according to changes in the sampling
regimes (Scenarios 2-7) and should result in different data sets
for a given random number seed owing to minor differences in
capture probabilities and reduced sampling effort.

Inputs into this routine include:
- A vector of future recruits
*/
cout<<"****"<<endl<<"Simulating Future Data"<<endl<<"****"<<endl<<endl;

int i,j,k; //index for year, individual, age=k
random_number_generator rng(rseed);
int agetag, yeartag;
bool tag;
double cv_len=0.1;
double sigma=3.0;
double pt=ptcap;

dvector rt(nyr+1,nyr+nsim); //future age-2 recruits
dvector s_nt4(syr,nyr+nsim);

/*-----
The following code deals with tagging and recapturing new recruits
only.
pcap is based on the general capture probability and the age-specific
selectivity coefficients that come from each of the 7 scenarios.
I'm not completely satisfied with this approach because there is a
lot of variability in the pta matrix in terms of age-specific
selectivity coefficients.
-----*/

for(i=syr;i<=nyr;i++)

o_mta(i)=mta(i);
o_rta(i)=rta(i);

rt=2000; rt(2011)=1000;
for(i=nyr+1;i<=nyr+nsim;i++)

for(j=1;j<=rt(i);j++) //loop over individuals

tag=false;
agetag=0;
double l_dev=cv_len*randn(rng);
for(k=sage;k<=nage;k++)

int iyr = i+k-sage; //index for cohort
//cout<<"year "<<iyr<<endl;
if(iyr>nyr+nsim) break; //break if past nyr+nsim
double len=lenage(k)+l_dev*lenage(k);
double survival=exp(-value(m)*linf/len); //length based survival rate
//double pcap = pt(iyr)*plogis(len,100,6); //check selectivity from results

```

```

//Change pcap to be age-based and use lowess 1/4 estimates from average pta values
double pcap=pt*sel_coffs(k,scenario);
double xx=randu(rng);
if(xx<=pcap) //individual is captured or recaptured.

    if(!tag) //first capture

        tag=true;
        double lentag=len+sigma*randn(rng); //measured length at tagging
        //agetag=k; //add aging error here from ALK.
        //cout<<sage<<" "<<k<<endl;
        agetag=length2age(lentag); //assignment of age based on Lew's polynomial
        yeartag=iyр;

        o_mta(iyr,agetag)++;
        //mta(iyr,agetag)++;

    else //recapture

        o_rta(iyr,agetag)++;
        o_rcta(yeartag,iyr,agetag)++;

if(k>=4)s_nt4(iyr)++;
if(randu(rng)>survival) break; //fish dies
if(agetag!=0) agetag++; //fish lives another year

//cout<<"Year "<<i<<endl;

//cout<<o_mta<<endl;
//exit(1);

/*The following code uses the MLE estimates of Untagged individuals U
to generate new marks.
*/
int iage;
ivector iU=value(U(nyr)); //number of unmarked animals in the terminal year
iU(sage)=rt(nyr+1); //assume 2000 recruits in terminal year
//cout<<"iU"<<endl<<iU<<"-----"<<endl<<endl;
for(iage=sage+1;iage<=nage;iage++)

    for(j=1;j<=iU(iage);j++)

        tag=false;
        agetag=0;
        double l_dev=cv_len*randn(rng);
        for(k=iage;k<=nage;k++)

            //int iyr = nyr+k-sage+1; //index for cohort
            int iyr = nyr+1+k-iage;
            //cout<<"year "<<iyr<<endl;

            if(iyr>nyr+nsim) break; //break if past nyr+nsim
            double len=lenage(k)+l_dev*lenage(k);
            double survival=exp(-value(m)*lrf/len); //length based survival rate
            //Change pcap to be age-based and use lowess 1/4 estimates from average pta values
            double pcap=pt*sel_coffs(k,scenario);
            double xx=randu(rng);
            if(xx<=pcap) //individual is captured or recaptured.

                if(!tag) //first capture

                    tag=true;
                    double lentag=len+sigma*randn(rng); //measured length at tagging
                    //agetag=k; //add aging error here from ALK.
                    agetag=length2age(lentag); //assignment of age based on Lew's polynomial
                    //cout<<iyr<<" "<<k<<" "<<agetag<<endl;
                    yeartag=iyр;
                    o_mta(iyr,agetag)++;
                    //mta(iyr,agetag)++;

                else //recapture

                    o_rta(iyr,agetag)++;
                    o_rcta(yeartag,iyr,agetag)++;

            if(k>=4)s_nt4(iyr)++;
            if(randu(rng)>survival) break; //fish dies
            if(agetag!=0) agetag++; //fish lives another year

```

```

//cout<<o_mta<<endl;
//exit(1);
//*Next need to write the code to look for recaptures of previously marked cohorts
from nyr*/
ivector iM=value(M(nyr));
//cout<<iM<<endl;
for(iage=sage+1;iage<=nage;iage++)

    for(j=1;j<=iM(iage);j++)

        double l_dev=cv_len*randn(rng);
        for(k=iage;k<=nage;k++)

            int iyr = nyr+k-iage+1;    //index for cohort
            if(iyr>nyr+nsim) break;    //break if past nyr+nsim
            double len=lenage(k)+l_dev*lenage(k);
            double survival=exp(-value(m)*linf/len);    //length based survival rate

            //Change pcap to be age-based and use lowess 1/4 estimates from average pta values
            double pcap=pt*sel_coefs(k,scenario);
            double xx=randu(rng);
            if(xx<=pcap)

                /*All fish are tagged here, so these are only recaptures*/
                //cout<<iyr<<" "<<k<<endl;
                o_rta(iyr,k)++;
                //not sure how to deal with cohort based groups, b/c I don't know the year tagged for iM

            if(k>=4)s_nt4(iyr)++;
            if(randu(rng)>survival) break;    //fish dies

//cout<<o_rta<<endl;

ofstream ofs("SimData.dat");
ofs<<syrr<<endl<<nyr+nsim<<endl;
ofs<<sage<<endl<<nage<<endl;
ofs<<vonbk<<endl<<linf<<endl;
ofs<<"#Length-at-age"<<endl<<lenage<<endl;
ofs<<"#Term age"<<endl<<termage<<endl;
ofs<<nva<<endl<<epics<<endl;
ofs<<"#M_phz"<<endl<<m_phz<<endl;
ofs<<"#mta"<<endl<<o_mta<<endl;
ofs<<"#rta"<<endl<<o_rta<<endl;
ofs<<"#rcta"<<endl<<o_rcta<<endl;
ofs<<"#eof"<<endl<<999<<endl;
ofs.close();

//Simulation values used for comparing estimates
ofstream ofss("SimValues.dat");
ivector yr(syr,nyr+nsim);
yr.fill_seqadd(syr,1);
dvector rtt(syr,nyr+nsim);

rtt(syr,nyr)=value(column(U,sage));
rtt(nyr+1,nyr+nsim)=rt(nyr+1,nyr+nsim);
rtt(nyr)=iU(sage);
s_nt4(syr,nyr)=value(rowsum(trans(trans(U).sub(4,nage))+trans(trans(M).sub(4,nage))));

ofss<<"yr"<<endl<<yr<<endl;
ofss<<"rt"<<endl<<rtt<<endl;
ofss<<"nt4"<<endl<<s_nt4<<endl;
cout<<"*** ---FINISHED WRITING SIMULATED DATA--- ***"<<endl;

FUNCTION void simulation_model()
    random_number_generator rng(rseed);
    long ntag=sum(mta);
    long recaps=sum(rta);
    mta.initialize();
    rta.initialize();
    rcta.initialize();

    int i,j,k;    //index for year, individual,age;
    int ssyr=syr-(nage-sage+1);    //birth year of oldest cohort in syr;
    int agetag,yeartag;
    bool tag;    //boolean values for capture and recapture
    double cv_len=0.1;
    double sigma=3.0;
    dvector rt(ssyr,nyr);
    dvector pt(ssyr,nyr);    //capture probability
    dvector nt4(ssyr,nyr);    //Age-4+ abundance.
    dmatrix tmp_mta(syr,nyr,sage,nage); tmp_mta.initialize();

```

```

if(rseed % 2)rt=3500+700; else rt=3500-700;

//SM Notes: get historical recruits and capture probabilities from Scenario 1
//Keep historical data the same and project new program from 2009 forwards.
//I.e., make this like an MSE evaluating changes in capture probabilities
//associated with changes in sampling design.

rt(ssyr-10,nnyr)=sim_rt;
//rt(2009)=3500-700;
pt=0.2; pt(ssyr,ssyr-1)=0;
pt(ssyr,nnyr)=sim_pt(ssyr,nnyr);

for(i=ssyr; i<nnyr; i++)

    for(j=1; j<=rt(i); j++) //loop over individuals

        tag=false;
        agetag=0;
        double l_dev=cv_len*randn(rng);
        for(k=sage; k<=nage; k++)

            int iyr = i+k-sage; //index for cohort
            //cout<<"year "<<iyr<<endl;
            if(iyr>nnyr) break; //break if past nnyr
            double len=lenage(k)+l_dev*lenage(k);
            double survival=exp(-value(m)*linf/len); //length based survival rate
            double pcap = pt(iyr)*plogis(len,100,6); //check selectivity from results
            double xx=randu(rng);
            if(xx<=pcap) //individual is captured or recaptured.

                if(!tag) //first capture

                    tag=true;
                    double lentag=len+sigma*randn(rng); //measured length at tagging
                    agetag=k; //add aging error here from ALK.
                    //agetag=length2age(lentag);
                    yeartag=iyr;

                    mta(iyr,agetag)++;

                else //recapture

                    rta(iyr,agetag)++;
                    rcta(yeartag,iyr,agetag)++;

            if(k>=4)nt4(iyr)++; //age-4 or older
            if(randu(rng)>survival) break; //fish dies
            if(agetag!=0) agetag++; //fish lives another year

cout<<"*****"<<endl;
cout<<"** Observed Simulated"<<endl;
cout<<"** -Marks released: "<<ntag<<" "<<sum(mta)<<endl;
cout<<"** -Marks recaptured: "<<recaps<<" "<<sum(rta)<<endl;
cout<<"*****"<<endl;
cout<<"*****"<<endl;

//-- Write true abundance of age 4+ fish and age-2 recruits
ofstream ofs("SimValues.dat");
ivector yr(ssyr,nnyr);
yr.fill_seqadd(ssyr,1);
ofs<<"yr"<<endl<<yr<<endl;
ofs<<"rt"<<endl<<rt<<endl;
ofs<<"nt4"<<endl<<nt4<<endl;

write_simulation_data();

FUNCTION void write_simulation_data()
//-- This routine writes simulation data

ofstream ofs("SimulationData.dat");
ofs<<ssyr<<endl<<nnyr<<endl;
ofs<<sage<<endl<<nage<<endl;
ofs<<vonbk<<endl<<linf<<endl;
ofs<<"#Length-at-age"<<endl<<lenage<<endl;
ofs<<"#Term age"<<endl<<termage<<endl;
ofs<<nva<<endl<<epics<<endl;
ofs<<"#M_phz"<<endl<<m_phz<<endl;
ofs<<"#mta"<<endl<<mta<<endl;
ofs<<"#rta"<<endl<<rta<<endl;
ofs<<"#rcta"<<endl<<rcta<<endl;
ofs<<"#eof"<<endl<<999<<endl;

cout<<"*** ---FINISHED WRITING SIMULATED DATA--- ***"<<endl;

```

The following is the ASMR.cxx file that contains the probability models (e.g., Poisson model) used in the ASMR analysis.

```
/*
*/
#include<admodel.h>

//normal distribution
dvariable dnorm(const dvariable& x, const double& mu, const double& std)

    double pi=3.141593;
    return 0.5*log(2.*pi)+log(std)+0.5*square(x-mu)/(std*std);

//plogis
double plogis(const double& x, const double& mu, const double& std)

    return 1./(1.+mfexp((mu-x)/std));

dvariable plogis(const dvariable& x, const double& mu, const dvariable& std)

    return 1./(1.+mfexp((mu-x)/std));

dvar_vector plogis(const dvector& x, const dvariable& mu, const dvariable& std)

    return 1./(1.+mfexp((mu-x)/std));

dvector plogis(const dvector& x, const double& mu, const double& std)

    return 1./(1.+mfexp((mu-x)/std));

dvar_vector plogis(const dvar_vector& x, const dvariable& mu, const dvariable& std)

    return 1./(1.+mfexp((mu-x)/std));

dvar_vector eplogis(const dvar_vector& x, const dvariable& alpha, const dvariable& beta, const dvariable& gamma)

    //exponential logistic based on Grant Thompson (1994) Paper, CJFAS.
    return (1./(1.-gamma))*pow((1.-gamma)/gamma,gamma)*elem_div(exp(alpha*gamma*(beta-x)),1.+exp(alpha*(beta-x)));

dvector eplogis(const dvector& x, const double& alpha, const double& beta, const double& gamma)

    //exponential logistic based on Grant Thompson (1994) Paper, CJFAS.
    return (1./(1.-gamma))*pow((1.-gamma)/gamma,gamma)*elem_div(exp(alpha*gamma*(beta-x)),1.+exp(alpha*(beta-x)));

//log normal distribution
dvariable dlnorm(const dvariable& x, const double& mu, const double& std)

    double pi=3.141593;
    return 0.5*log(2.*pi)+log(std)+log(x)+square(log(x)-mu)/(2.*std*std);

dvariable dpois(const double& k, const dvariable& lambda)

    return -k*log(lambda)+lambda + gammln(k+1.);

dvariable dpois(const dvector& k, const dvar_vector& lambda)

    /* A modification to the poisson distribution
       where we first loop over observations k and
       get the indexes for non-zero values.
    */
    int i,imin,imax;
```

```

imin=k.indexmin();
imax=k.indexmax();
dvariable loglike=0.;
for(i = imin; i<=imax;i++)
    if(k(i)>0) loglike += lambda(i)-k(i)*log(lambda(i));
//return sum(lambda - elem_prod(k,log(lambda)));
return loglike;

dvariable dnbinom(const dvector& x, const dvar_vector& lambda, const dvariable& tau, dvector& residual)

//the observed counts are in x
//lambda is the predicted count
//tau is the overdispersion parameter
RETURN_ARRAYS_INCREMENT();
int i,imin,imax;
double o=1e-30;
imin=x.indexmin();
imax=x.indexmax();
dvariable loglike=0.;
residual.initialize();

for(i = imin; i<=imax; i++)
    if(x(i)>0)
        dvariable p=tau/(tau+lambda(i));
        loglike += gammln(tau+x(i)) - gammln(tau) //- gammln(x(i)+1)
                +tau*log(p) + x(i)*log(1.-p);
        residual(i) =value(x(i)-lambda(i))/sqrt(o+lambda(i)+square(lambda(i))/tau) ) ;

//cout<<"OK in dnbinom"<<endl;
RETURN_ARRAYS_DECREMENT();
return(-1.0 *loglike);

dvariable dpois_residual(const dvector& k, const dvar_vector& lambda, dvector& residual)

/* A modification to the poisson distribution
where we first loop over observations k and
get the indexes for non-zero values.

This was not done in the original ASMR model
so I have commented it out for now, and added
a tiny number to lambda.
*/
RETURN_ARRAYS_INCREMENT();
double o=1.e-30;
int i,imin,imax;
imin=k.indexmin();
imax=k.indexmax();
dvariable loglike=0.;
for(i = imin; i<=imax;i++)

    //if(k(i)>0)
    loglike += lambda(i)-k(i)*log(lambda(i)+o);
    //cout<<k(i)<<" "<<lambda(i)<<endl;
    residual(i) = value((k(i)-lambda(i))/sqrt(lambda(i)+o));
    //

//return sum(lambda - elem_prod(k,log(lambda)));
RETURN_ARRAYS_DECREMENT();
return loglike;

dvar_matrix ALK(dvar_vector mu, dvar_vector sig, dvector x)

//This function returns an Age-Length Key
int i, j;
dvariable z1;
dvariable z2;
int si,ni; si=mu.indexmin(); ni=mu.indexmax();
int sj,nj; sj=x.indexmin(); nj=x.indexmax();
dvar_matrix pdf(si,ni,sj,nj);
pdf.initialize();
double xs=0.5*(x[sj+1]-x[sj]);
for(i=si;i<=ni;i++) //loop over ages

    for(j=sj;j<=nj;j++) //loop over length bins

        z1=((x(j)-xs)-mu(i))/sig(i);
        z2=((x(j)+xs)-mu(i))/sig(i);
        pdf(i,j)=cumd_norm(z2)-cumd_norm(z1);
    //end nbins
    //pdf(i)/=sum(pdf(i));
//end nage
pdf/=sum(pdf);
return(pdf);

```