BACKWARDS FRAM (bkFRAM; FRAMVS2.18)

Introduction

Backwards FRAM is a utility that finds FRAM starting cohorts if estimates of escapements (Coho) or (extreme-)terminal run sizes (Chinook) and fishery catches (landed and non-retention) are provided. Starting cohorts are initial FRAM run sizes before natural mortality, fishing mortality, and maturation. The program iteratively adjusts stock recruit scalars (a surrogate for starting cohorts, see Equation 1) and runs FRAM forward until the resulting escapements or terminal run sizes match the targets. For Chinook, bkFRAM calculates age 3 to 5 starting cohorts, for Coho, starting cohorts are in units of January Age 3 fish. For Chinook, target abundances are generally closely aligned with historical definitions of terminal or extreme terminal run sizes (TRS, ETRS) used for pre-season forecasts or post-season accounting (see Table 1) and exclude age 2s.

Post-season bkFRAM runs are generally conducted to create the starting cohorts that result in observed escapements (or ETRS, TRS) given known fishery impacts. These cohorts are needed to calculate post-season exploitation rates. Pre-season bkFRAM runs are generally conducted to create starting cohorts that result in forecasted (extreme-)terminal run sizes. These pre-season runs will likely be populated with recent year average fishery impacts to obtain 'best' estimates of pre-fishing abundances.

Methods

BkFRAM can be run for an individual stock or a combination of stocks (including all stocks). It requires a "seed" run that contains desired fishery impacts (observed values for post-season runs or recent year average values for pre-season runs). The seed run will also contain starting cohorts for stocks where this parameter is known; i.e. the forecast is already in starting cohort units rather than terminal run size units.

BkFRAM can run in three modes. The mode is selected using a flag in the input process. Flags are stock-specific, allowing all three flags (values = 0,1,2) to be used in a single bkFRAM model run.

- Mode 0 (flag 0): This mode does not use bkFRAM to find starting stock scalars. Instead, FRAM uses the starting cohort values from the existing "StockRecruit" table.
- Mode 1 (flag 1): This mode uses an algorithm to iteratively adjust stock recruit scalars until the target abundance is achieved. Target abundances are adipose mark specific. This method is used when the mark rate of a stock is known.
- Mode 2 (flag 2): This mode is selected when target mark rates are unavailable. Algorithms from mode 1 are used to find the starting stock recruit scalars that result in the combined target abundance (marked and unmarked components). The program will then apply mark rates derived from the existing "StockRecruit" table to split starting cohorts into marked and unmarked components.

Core Algorithms

In FRAM, run specific starting cohorts are related to base period starting cohort. This is done by multiplying the base period cohort times a stock recruit scalar resulting in the desired starting cohort.

For more information about FRAM algorithms and the base period please see "Fishery Regulation and Assessment Model (FRAM) – Technical Documentation for Chinook and Coho – October 2008".

Computing Starting Cohorts

Equation 1:

$$StockScalar_s = \frac{Cohort_s}{BPCohort_s}$$

or

$$Cohort_s = StockScalar_s * BPAbundance_s$$

Iteratively Adjusting Stock Recruit Scalars (Mode 1 and 2)

For Chinook

The Chinook algorithm uses forward (regular) FRAM calculations to solve for the stock recruit scalar needed to obtain a target terminal run size, given known time step fishery catches, natural mortality rates, and maturation rates.

The only parameter changing between iterations are stock-specific catches, resulting in new recruit scalars, resulting in new stock catches, etc.

To back-calculate starting cohorts given a known terminal run size, the catch summaries (PTCatch, TermCatch) entering the main equation have to strictly follow FRAM terminal run size definitions and handling (i.e. terminal versus pre-terminal treatment, inclusion or exclusion of incidental mortality from TRS definitions, etc.).

Equation 2:

$$\begin{aligned}StockScalar_{(s,a,t1)} = ((&BkTarget_{(s,a)} + (PTCatch_{(s,a,t1)} * MatRate_{(s,a,t1)} + (PTCatch_{(s,a,t1)} * (1 - MatRate_{(s,a,t1)}) * (1 - M_{(a,t2)}) + PTCatch_{(s,a,t2)}) * MatRate_{(s,a,t2)} + ((PTCatch_{(s,a,t1)} * (1 - MatRate_{(s,a,t1)}) * (1 - M_{(a,t2)}) + PTCatch_{(s,a,t2)}) * (1 - MatRate_{(s,a,t2)}) * (1 - M_{(a,t3)}) + PTCatch_{(s,a,t3)}) * MatRate_{(s,a,t3)})) + TermCatch_{(s,a,t1)} + TermCatch_{(s,a,t2)} + TermCatch_{(s,a,t3)})/((1 - M_{(a,t1)}) * MatRate_{(s,a,t1)} + (1 - M_{(a,t1)}) * (1 - MatRate_{(s,a,t1)}) * (1 - M_{(a,t2)}) * MatRate_{(s,a,t2)} + (1 - M_{(a,t1)}) * (1 - MatRate_{(s,a,t1)}) * (1 - M_{(a,t2)}) * (1 - MatRate_{(s,a,t2)}) * (1 - M_{(a,t3)}) * MatRate_{(s,a,t3)})/BPCohort_{(s,a)}\end{aligned}$$

For Coho

The Coho algorithm is simpler than that utilized for Chinook, as maturation only occurs in the final time step (October-December) and is always 100%. Similar to Chinook the calculation is based on adding time step mortalities of a stock to the escapement target, expanding for natural mortality occurring in each time step, and working backwards from time 5 to time 1. The

mortalities are calculated by summing landed, drop-off, and non-retention mortalities over all fisheries for a stock and time step.

Mark-selective fishery bias correction calculations used for Coho produce an error when the exploitation rate exceeds 100%. This frequently occurs during early iterations, because the abundances from the seed run (usually the pre-season run) are not related to the post-season catches; i.e. low pre-season forecast but high post-season terminal catches. To avoid errors, the first iteration initiates starting cohorts at 1000 times base period abundance. Additionally, the first 7 iterations are run without bias correction to get sufficiently close to the target escapement, before adding mark selective fishing bias calculations for all remaining iterations. Thus in coho, the seed abundances are only used for stocks that are not adjusted during bkFRAM (Flag = 0) or for calculating mark rates (Flag = 2).

Equation 3 & 4:

$$StockMort(s,t) = \sum_f \text{LandedCatch}(s,a,f,t) + \text{MSFLandedCatch}(s,a,f,t) + \text{NonRetention}(s,a,f,t) + \text{MSFNonRetention}(s,a,f,t) + \text{DropOff}(s,a,f,t) + \text{MSFDropOff}(s,a,f,t)$$

$$\text{StockScalar}_{(s,a)} = ((((((\text{BkTarget}(s) + \text{StockMort}(s,t_5)) / (1 - M(a,t_5)) + \text{StockMort}(s,t_4)) / (1 - M(a,t_4)) + \text{StockMort}(s,t_3)) / (1 - M(a,t_3)) + \text{StockMort}(s,t2)) / (1 - M(a,t_2)) + \text{StockMort}(s,t_1)) / (1 - M(a,t_1))) / \text{BaseCohortSize}(s,a)$$

Where,

a	=	*Age*
BkTarget	=	*Backwards FRAM run size target; escapement or (extreme)terminal run size*
BPCohort	=	*Base period starting cohort*
Cohort	=	*Starting Cohort*
FRAM_Escapement	=	*Escapement or (extreme)terminal run size resulting from a forward FRAM run*
f	=	*Fisheries*
M	=	*Natural Mortality*
MatRate	=	*Maturation Rate*
PTCatch	=	*Mortality summed over all preterminal fisheries*
s	=	*Stock*
StockMort	=	*Mortality summed over all fisheries for a stock and time step*
StockScalar	=	*Stock Recruit Scalar*
t	=	*Time step; t1 = Oct-April, t2 = May-June, t3 = July-Sep*
TermCatch	=	*Terminal catch summed over all terminal fisheries*

Calculations for Unknown Mark Rates (Mode 2)

When mark rates are unknown, and thus the marked and unmarked components of a stock are unknown, FRAM uses mark rates from the starting cohorts of the "seed run" (usually the final

pre-season model run), found in the "StockRecruit" table of the MS Access database. It applies these to the combined stock (marked plus unmarked) starting cohort to iteratively compute new stock recruit scalars. Rather than targeting a marked or unmarked escapement value, the algorithm adds marked and unmarked FRAM escapements of a stock and compares these to the target value.

Unlike Coho (Figure 2), the Chinook bkInput screen (Figure 5) has a row for total stock abundance (marked plus unmarked). A flag of 2 in the total stock row signals the use of mark rates from the "seed run". For Coho, a flag of 2 in either the marked or unmarked row will treat the input value as a total combined (marked plus unmarked) stock abundance and split it into marked and unmarked components using mark rates from the initial "seed" model run. If there are conflicting flags for the same stock, the program will produce an error message. If using flag 2 for a stock, the corresponding marked or unmarked component should be flagged as "0".

BkFRAM and Age-2: Chinook

For Chinook, the user can select whether to use age-3 recruit scalars for age-2 starting cohorts. A checkbox is located on the 'Backwards FRAM Run Menu'. A check mark will result in using the same recruit scalar for age-2s that was calculated in the final iteration for age-3s.

X Age 2 from 3

Convergence Criteria

The program iterates until the FRAM escapement or terminal run size is within one fish of the target stock abundance and then terminates the iteration process, unless the user specifies a number of iterations on the 'Backwards FRAM Run Menu' that is smaller than the number of iterations needed to achieve convergence.

Equation 5:

$$|\text{FRAM\_Escapement}_{(s,a)} - \text{BkTarget}_{(s,a)}| <= 1$$

Testing and Supplemental Iteration Information

When bkFRAM iterations are complete, a text file called 'BackFRAMCheck.txt' is created in the same directory as the FRAM database in use. This file contains information about the number of iterations performed to reach convergence and a comparison of FRAM calculated run sizes to run size targets for each stock and each iteration.

Example of the BackFRAMCheck file:

Iteration #1

Stk#	*Age*	*FRAM-TRS*	*Target-TRS*	*ScaleFactor*	*New-Scalar*	*Old-Scalar*	*Cohort*	*StockName*
1	*3*	*2951*	*2950*	*0.9997*	*0.1534*	*0.1534*	*8879*	*U-NkSm FF*
1	*4*	*1403*	*1403*	*0.9999*	*0.1262*	*0.1262*	*2107*	*U-NkSm FF*

Running BkFRAM

1. Open the FRAM program, select the FRAM MS Access database file that contains the desired seed run, then open the FRAM model run you will use as your seed run from the database.

2. On the 'Main Menu' select 'Post Season Run'

   Post Season Run

3. This will bring up the 'Backwards FRAM Run Menu'
   From here one can either select to set the Backwards Abundance Targets, by clicking

   Target Escapements

   or if the target values are already set, select the number of desired iterations and press 'Start Iterations'. The default number of iterations is 99. This number will rarely be reached, as iterations automatically terminate when the convergence criteria is met.

   Start Iterations

   For Chinook, upon selection of the 'Start Iterations' buttons a message box pops-up requiring the user to select whether to use TAMI catches for the run. If the FRAM run already has the correct TAMI catches stored, either because the catches were entered directly into FRAM or because FRAM has already been run forward with the correct TAMM (TAMI catches are then automatically saved into the FRAM database), then the answer would be "No".
4. Setting Target Escapements (Figure 2 and 5).
   Selecting the 'Target Escapements' button brings up the 'Target Escapements for Backwards FRAM' menu.
   <u>For Chinook</u>
   To enter target terminal run sizes manually: Enter terminal run sizes by stock, age, and mark status and flag as "1" in the correspondingly labeled cells. If the mark rate is unknown, enter the combined (marked plus unmarked) abundance in the "Total" row and flag as "2" (i.e. will use mark rates from "seed run"). If you do not wish to overwrite exisiting cohort sizes for a stock, flag relevant row(s) (Total, Marked, Unmarked) as "0".
   To enter target terminal run sizes using an Excel file: Target run sizes can also be loaded from a MS Excel template file designed for model inputs ("BkFRAM_ChinookTemplate...xls") by selecting the 'Import Escapements' button and then locating and selecting the file with the desired values.

   Import Escapements

   Escapement values can also be exported by selecting the top left corner of the table grid in the software interface and pressing 'Ctrl C' to copy and 'Ctrl V' to paste.

   Note: The button "Load Back-FRAM Catch" is currently not functional for Chinook in the bkFRAM menu. This functionality is not needed as catches can be loaded through the forward FRAM Main Menu, Edit Model Run, Input Menu-Fishery, Quota/Scalers or Non-Retention sections from a template file.

   <u>For Coho</u>
   To enter target escapements manually: Enter escapements by stock and mark status and flag as "1" in the correspondingly labeled cells. Coho escapement values should exclude

jacks (age 2), as coho are assumed to be age 3 in the FRAM model framework. If the mark rate is unknown, enter the combined (marked plus unmarked) escapement in either the marked or unmarked stock row and flag as "2" (will use mark rates from "seed run"). If you do not wish to overwrite exisiting cohort sizes for a stock, flag relevant row(s) (Marked, Unmarked) as "0".

To enter target escapements using an Excel file: Escapement values can also be loaded from a MS Excel template file designed for model inputs ("Coho Escape….xls") by selecting the 'Import Escapements' button and then locating and selecting the file with the desired values.

Import Escapements

The template file workbook needs to contain a worksheet tab called "FRAMEscape2". The escapement values within the worksheet should be entered and organized as shown below:

FRAM Unit coho escapement summary			
FRAM Stock		**1999**	**2000**
Age 3 nkskrw unmarked		953	3,530
Age 3 nkskrw marked			
Age 3 kendlh unmarked			
Age 3 kendlh marked		2,561	3,100
Age 3 skokmh unmarked			
Age 3 skokmh marked		29,179	31,591

A column exists for every year desired for updating model inputs. The program will request a single "Year" to load from the Excel worksheet.

Escapement values can be also be exported by selecting the 'Export to Spreadsheet' button on the 'Target Escapements for Backwards FRAM' menu using the same workbook template file needed for importing. The program will also request the "Year" of data to export.

Export to Spreadsheet

The 'Load Back-FRAM Catch' button is working for Coho in bkFRAM, although catches are usually loaded into the model run through the forward FRAM Main Menu, Edit Model Run, Input Menu-Fishery, Quota/Scalers or Non-Retention sections from a template file. The necessary file for using the 'Load Back-FRAM Catch' button will likely be titled "BackFRAM_Catch…xls", with each individual year on a separate worksheet titled "Year"FRAM (i.e. 2017FRAM).

Load Back-FRAM Catch

Once the file is selected, the program requests a run year to load.

The catch values within the worksheet should be entered and organized as shown below:

Return Year 2001

FRAM Catch Unit	FW comm unit	TP1 Jan-June		TP2 July		TP3 Aug		TP4 Sept		TP5 Oct-Dec		**Total**
No Cal Trm		#N/A		#N/A		#N/A		#N/A		#N/A		**#N/A**
Cn Cal Trm		#N/A		#N/A		#N/A		#N/A		#N/A		**#N/A**
Ft Brg Spt		325		110		2		0		0		**437**

Once the input values are updated, select 'Exit' to return to the 'Backwards FRAM Run Menu' to 'Start Iterations'.

5. After iterations are complete, the programs returns a result screen (Figures 3 and 6) which lists target escapements, resulting BkFRAM escapements, and new stock recruit scalars side by side in order to assess whether values have sufficiently converged.
6. Saving targets and resulting stock recruit scalars.
   After running BkFRAM, backwards targets, flags, and new stock recruit scalars can be saved by selecting **'Save BkFRAM Targets and new Recruit Scalars'** from the 'Backwards FRAM Run Menu'. This brings up the following message box: "*This action saves BkFRAMTargets as well as Recruit Scalars. To save, please follow instructions of next menu."* Then the 'Save-Menu' pops-up, where the user can select to replace the existing run, save as a new run, or cancel save. If the user selects the 'Save BkFRAM Targets and New Recruit Scalars' button after the targets are entered or importet, but before iterations are run, the save action will only save the new targets.

Appendix:

Noteworthy changes to historical bkFRAM

- The previous version of bkFRAM had 4 flags (instead of 3) for computing starting cohorts. There were two methods (flag 1 and flag 3) to compute starting cohorts. Flag 3 was added after a weakness with the flag 1 method was discovered. The flag 1 method was flawed for stocks that matured in more than one time step. The flag 3 method also had shortcoming, because it relied on a linear algorithm to approximate non-linear processes, occasionally leading to non-convergence. Modelers would usually choose method 1 for all stocks with maturation in one time steps and method 3 for the remaining stocks. For compatibility with old FRAM model runs, a flag 3 will still process in new bkFRAM, but the calculation used for adjusting the stock recruit scalars follows those outline for flag 1 in this document.
- Prior to this version of bkFRAM, for method 2 (bkFRAM targets are for the combined marked and unmarked stock), the mark rates from the existing "StockRecruit" table would have been applied to the escapement target, thus turning a pre-fishing mark rate into a post-fishing mark rate. This likely resulted in overestimating mark rates when pre-terminal mark-selective fisheries were modeled. The new method 2 uses pre-fishing mark rates on pre-fishing abundances and is generally more compatible with the origin of mark-rate forecasts, which are usually derived from mark-rate at release information.
- Prior to FRAMVS version 2.19, for method 2, running bkFRAM would have resulted in the computation of marked and unmarked targets for stocks missing a mark rate. These targets would have been saved at the end of a bkFRAM run as the new targets, thus overwriting the original targets and the flag would have been changed to 1. This is no longer the case. The original escapement target is saved and never overwritten. This preserves the original escapement input and prevents saving a new target that may not have fully converged on the desired value.

		Input		BkFRAM Result	Program Saves	
Method	**Stock**	**BkTarget**	**BkFlag**	**Escapement**	**BkTarget**	**BkFlag**
Old	Tulalip UM	0	0	198	198	1
Old	Tulalip M	2000	2	1800	1800	1
New	Tulalip UM	0	0	198	0	0
New	Tulalip M	2000	2	1800	2000	2

Visual Basic Code in FRAM program for BkFRAM: PSEUDO CODE

CHINOOK

'when pressing "START ITERATIONS" button

BackChinFRAM (main subroutine)

- **Call** BackChinArrays (define BkRuns = **BackwardsChinook**)

For BackFRAMIteration = 1 To NumBackFRAMIterations

- **Call** Check_CHINOOK_TerminalRun
  - **Call** RunCalcs()'forward calcualtions
  - **Call** SumChinTermRun (calculates **TermChinRun**)'add catches
  - **Compare BackwardsChinook to TermChinRun and compute Recruit Scalars**

Next Iter

End BackChinFRAM

```
Class FVS_BackwardsTarget
'Read in Backwards Targets
BackwardsChinook(Stk, Age) = BFTargetGrid.Item(Age - 1, Stk - 1).Value
Call BackChinookFram(BackFRAMIteration, NumBackFRAMIterations)
      Call BackChinArrays()
```

- Set Backwards Chinook Terminal Run Names

```
ReDim TermRunName(37)
      TermRunName(1) = "Nook/Samish Fall"
```

- Set TermStock Array for holding bkTargets for either total, marked, or unmarked stock components (3 values per 39 FRAM stocks)

```
ReDim TermStockNum(NumStk + NumStk / 2 - 1)
TermStockNum(1) = -1 'NookSam Total
TermStockNum(2) = 1  'NookSam UM
TermStockNum(3) = 2  'NookSam M
TermStockNum(4) = -2 'NookSpring Total
TermStockNum(5) = 3  'NF NookSpring UM
TermStockNum(6) = 4  'NF NookSpring M
TermStockNum(7) = 5  'SF NookSpring UM
TermStockNum(8) = 6  'SF NookSpring M
```

- Set TFish Arrays defining fisheries to include in terminal run size definition

```
ReDim TFish(NumStk / 2, 10)
  TFish(1, 1) = 3 : TFish(1, 2) = 39 : TFish(1, 3) = 40 :
  TFish(1, 4) = 73
```

- Set TTime Array defining time steps to include in terminal run definition

```
ReDim TTime(NumStk / 2, 2)
  TTime(1, 1) = 3
```

```
'Compute mark rates of the original mode run
     For TRun = 1 To NumStk + NumChinTermRun
         For Age = 3 To 5
           StartRate(Stk, Age) = StockRecruit(Stk, Age, 1) / (StockRecruit(Stk, Age, 1) + StockRecruit(Stk + 1,
           Age, 1)) ' UM
           StartRate(Stk + 1, Age) = StockRecruit(Stk + 1, Age, 1) (StockRecruit(Stk, Age, 1) + StockRecruit(Stk
           + 1, Age, 1)) ' M
         Next Age
     Next TRun
'Start iterations
For BackFRAMIteration = 1 To NumBackFRAMIterations
    Call Check_CHINOOK_TerminalRun 'this is where all the magic happens
     Call RunCalcs() 'run forward to re-calculate catches by stock
           For TRun = 1 To NumStk + NumChinTermRuns
                 If BackwardsFlag(TRun) = 2 Then 'combined M and UM
                       Call SumChinTermRun 'sums catches
                         For J = StartNum To EndNum
                             Stk = TermStockNum(J)
                             TSum = TermRunStock(Stk)
                             For Age As Integer = 3 To 5
                             '- Sum Escapement
                                For TStep As Integer = 1 To 3
                                   TermChinRun(J, Age) = TermChinRun(J, Age) + Escape(Stk, Age, TStep)
                                Next TStep
                             '- Sum Terminal Fishery Catches
                                For I = 2 To TFish(TSum, 1) + 1
                                'Loop through stock specific fisheries and calculate the FRAM terminal run size
                                which uses bkFRAM definitions and is directly compared to the bkFRAM target
                                   Fish = TFish(TSum, I)
                                   For TStep = TTime(TSum, 1)To TTime(TSum, 2)
                                      TermChinRun(J,Age)= TermChinRun(J, Age) + LandedCatch(Stk, Age, Fish,
                                      TStep) + MSFLandedCatch(Stk, Age, Fish, TStep)
                                   Next TStep
                                Next I
                              'sum separately over terminal and preterminal fisheries not part of TRS
                              definition. These catches are needed to expand TRS to Starting Cohort
                              For TStep = 1 To NumSteps - 1
```

```
                For Fish = 1 To NumFish - 2
                'exclude esc, fw net & sport
                   Select Case Fish 'for fisheries part of TRS definition
                      Case TFish(TSum,2),TFish(TSum,3),TFish(TSum,4),TFish(TSum,5),
                      TFish(TSum,6),TFish(TSum,7),TFish(TSum,8),TFish(TSum,9),TFish(TSum,10)
                         Select Case TStep
                            Case TTime(TSum, 1), TTime(TSum, 2)
                              'add incidental mortality from terminal fisheries, because they
                              are not part of the bkFRAM TRS definition
                              AgeTSCatchTerm+= Shakers(Stk,Age,Fish,TStep) + NonRetention +
                              DropOff + MSFShakers + MSFNonRetention + MSFDropOff(
                            Else 'add moratlity from non terminal time steps
                              AgeTSCatchTerm += LandedCatch + Shakers + NonRetention +
                              DropOff + MSFLandedCatch + MSFShakers + MSFNonRetention +
                              MSFDropOff
                            End If
                         End Select
                   Case Else 'fisheries not part of TRS definition
                      If TerminalFisheryFlag(Fish, TStep) = 0 Then
                         AgeTSCatch += LandedCatch + Shakers + NonRetention + DropOff  +
                         MSFLandedCatch + MSFShakers + MSFNonRetention + MSFDropOff
                      Else
                         AgeTSCatchTerm  += LandedCatch + Shakers + NonRetention + DropOff  +
                         MSFLandedCatch + MSFShakers + MSFNonRetention + MSFDroOff
                      End If
                   End Select
                Next Fish
             Next TStep
          Next Age
       Next J
    End SumChinTermRun
    For Age = 3 To 5
        'sum marked and unmarked catches
          AgeTSCatch(Stk, Age, TStep) = AgeTSCatch(Stk) + AgeTSCatch (Stk + 1)
          AgeTSCatchTerm(Stk, Age, TStep) = AgeTSCatchTerm(Stk) + AgeTSCatchTerm(Stk + 1)
        'main bkFRAM algorithm
          StockRecruit(Stk, Age,1) = ((BackwardsChinook(TRun,Age)+((AgeTSCatch(Stk,Age,1)) *
          MaturationRate(Stk,Age,1)+((AgeTSCatch(Stk,Age,1))* (1- MaturationRate(Stk,Age,1))
          * (1-NaturalMortality(Age,2))+(AgeTSCatch(Stk,Age,2)))* MaturationRate(Stk,Age,2)
          +(((AgeTSCatch(Stk,Age,1))*(1-MaturationRate(Stk,Age,1))*(1-NaturalMortality(Age,
          2))+(AgeTSCatch(Stk,Age,2)))*(1-MaturationRate(Stk,Age,2))*(1-NaturalMortality(Age,
```

```
                                3))+(AgeTSCatch(Stk,Age,3)))*MaturationRate(Stk,Age,3)))+AgeTSCatchTerm(Stk,Age,1)
                                + AgeTSCatchTerm(Stk,Age,2)+AgeTSCatchTerm(Stk,Age,3))/
                                ((1-NaturalMortality(Age,1)) * MaturationRate(Stk,Age,1)+(1-NaturalMortality
                                (Age,1))*(1-MaturationRate(Stk,Age, 1))*(1-NaturalMortality
                                (Age,2))*MaturationRate(Stk,Age,2)+(1-NaturalMortality(Age, 1))*(1-MaturationRate
                                (Stk,Age,1))*(1-NaturalMortality(Age,2))*(1-MaturationRate (Stk,Age,2))*(1-
                                NaturalMortality(Age,3))*MaturationRate(Stk,Age,3))/
                                BaseCohortSize(Stk, Age)
                               'split into marked and unmarked
                                StockRecruit(Stk+1,Age,1) = StockRecruit(Stk, Age, 1) * StartRate(Stk + 1, Age)
                                StockRecruit(Stk, Age, 1) = StockRecruit(Stk, Age, 1) * StartRate(Stk, Age)
                               'sum marked and unmarked terminal runs
                                TermChinRun(TRun, Age) = TermChinRun(TRun + 1, Age) + TermChinRun(TRun + 2, Age)
                    ElseIf BackwardsFlag(TRun) = 1 Then
                          'same as Flag 2 above but catches and TRS not added for marked and unmarked. StockRecruit
                          is not split.
                    EndIf
               Next TRun
          'Iterate until bkTarget is within one Chinook of FRAM Terminal Chinook Run
            For TRun = 1 To NumStk + NumChinTermRun
                    For Age = 3 To 5
                        If BackwardsChinook(TRun, Age) > 0 And BackwardsFlag(TRun) <> 0 Then
                            If Math.Abs(BackwardsChinook(TRun, Age) - TermChinRun(TRun, Age)) > 1 Then
                                DoneIterating = DoneIterating + 1
                            End If
                        End If
                    Next Age
            Next TRun
          If DoneIterating = 0 Then
             Exit For
          End If
Next BackFRAMIteration
```

COHO

```
Public Class FVS_BackwardsFram
Private Sub StartIterationsButton_Click
      'Setting flags and targets for stock aggregates with "Flag 2"
      For BackFRAMIteration = 1 To NumBackFRAMIterations
            Call RunBackFRAM() '(in Sub RunCalcs)
                  Call ScaleCohort()'Initialize StartingCohort as 1000 * BPCohort
                        For TStep = 1 to NumSteps
                              Call NatMort()
                              Call CompCatch(PTerm)
                              Call IncMort(PTerm)
                              Call Mature()
                              Call CompCatch(Term)
                              Call IncMort(Term)
                              Call CompEscape()
            Call Check_BackwardsTarget
                  Compare BackwardsTarget to Escape and iteratively compute Recruit Scalars
      Next BackFRAMIteration
```

```
Public Class FVS_BackwardsFram
Private Sub StartIterationsButton_Click(ByVal sender As Object, ByVal e As System.EventArgs
    For Stk As Integer = 1 To NumStk
        If BackwardsFlag(Stk) = 2 Then ' use starting mark rate on starting cohorts rather than escape targets
           If (Stk Mod 2) = 0 Then
               '- Marked Target ... process combined target in unmarked stock spot
               If BackwardsFlag(Stk - 1) = 0 Then
                  SumScalers = StockRecruit(Stk, 3, 1) + StockRecruit(Stk - 1, 3, 1)
                      If SumScalers = 0 Then
                            MsgBox("Error - Backwards Stock FLAG = 2 points to Stock Scalers = ZERO" & vbCrLf &
                            Exit Sub
                      End If
                  RunBackwardsTarget(Stk - 1) = BackwardsTarget(Stk) / 2
                  RunBackwardsTarget(Stk) = RunBackwardsTarget(Stk - 1)
                  RunBackwardsFlag(Stk - 1) = 2
                  RunBackwardsFlag(Stk) = 2
               Else ' creates error message when both the marked and unmarked stock component have a flag of 2
                  MsgBox("FLAG = 2 - Error for Backwards FRAM Target Esc" & vbCrLf & "Stock# " & Stk.ToString & "
                  Exit Sub
               End If
```

```
        Else
           '- UnMarked Target ...
           SumScalers = StockRecruit(Stk, 3, 1) + StockRecruit(Stk + 1, 3, 1)
           If BackwardsFlag(Stk + 1) = 0 Then
              If SumScalers = 0 Then
                 MsgBox("Error - Backwards Stock FLAG = 2 points to Stock Scalers = ZERO" & vbCrLf & "Stock
                 Exit Sub
              End If
              RunBackwardsTarget(Stk + 1) = BackwardsTarget(Stk) / 2
              RunBackwardsTarget(Stk) = RunBackwardsTarget(Stk + 1)
              RunBackwardsFlag(Stk + 1) = 2
              RunBackwardsFlag(Stk) = 2
              Stk = Stk + 1
           Else
              MsgBox("FLAG = 2 - Error for Backwards FRAM Target Esc" & vbCrLf & "Stock# " & Stk -
              Exit Sub
           End If
        End If
     End If
For BackFRAMIteration = 1 To NumBackFRAMIterations
   Call RunBackFRAM()
      Call ScaleCohort()
         ' prevent ER from exceeding 100% otherwise MSF bias corrected equation produce error
         If BackFRAMIteration < 8 Then 'don't start bias calculations until target escapemetns are close
           MSFBiasFlag = False
         Else
           MSFBiasFlag = SaveInitialFlag
         End If
         If BackFRAMIteration < 2 Then
           For Stk = 1 To NumStk
               If BackwardsFlag(Stk) > 0 Or RunBackwardsFlag(Stk) > 0 Then
                   Cohort(Stk, Age, PTerm, 1) = 1000 * BaseCohortSize(Stk, Age)
               End If
           Next Stk
      For TStep = 1 To NumSteps
         Call NatMort()
         Call CompCatch(PTerm)
         Call IncMort(PTerm)
         Call Mature()
         Call CompCatch(Term)
         Call IncMort(Term)
```

```
        Call CompEscape()

    Call Check_BackwardsTarget(BackFRAMIteration, NumBackFRAMIterations)
        For Stk = 1 To NumStk
           For Fish = 1 To NumFish
              For timestep = 1 To NumSteps
                 StockMort(Stk, timestep) += LandedCatch(Stk, 3, Fish, timestep) + MSFLandedCatch(Stk, 3,
                 Fish, timestep) + NonRetention(Stk, 3, Fish, timestep) + MSFNonRetention(Stk, 3, Fish,
                 timestep) + DropOff(Stk, 3, Fish, timestep) + MSFDropOff(Stk, 3, Fish, timestep)
              Next
           Next
        Next Stk

    For Stk As Integer = 1 To NumStk
       If StockRecruit(Stk, Age, 1) <> 0 And BackwardsTarget(Stk) <> 0 And BackwardsFlag(Stk) = 1 Then
          StockRecruit(Stk, Age, 1) = ((((((BackwardsTarget(Stk) + StockMort(Stk, 5)) / (1 -
          NaturalMortality(3, 5)) + StockMort(Stk, 4)) / (1 - NaturalMortality(3, 4)) + StockMort(Stk, 3)) /
          (1 - NaturalMortality(3, 3)) + StockMort(Stk, 2)) / (1 - NaturalMortality(3, 2)) + StockMort(Stk,
          1)) / (1 - NaturalMortality(3, 1))) / BaseCohortSize(Stk, Age)
          If BackwardsTarget(Stk) > 0 Then
             If Math.Abs(BackwardsTarget(Stk) - Escape(Stk, Age, TStep)) > 1 Then
                DoneIterating = DoneIterating + 1
             End If
          End If
       Else
          If RunBackwardsFlag(Stk) = 2 Then 'combined marked and unmarked target
             StockRecruit(Stk, Age, 1) = ((((((RunBackwardsTarget(Stk) * 2 + StockMort(Stk, 5) +
             StockMort(Stk + 1, 5)) / (1 - NaturalMortality(3, 5)) + StockMort(Stk, 4) + StockMort(Stk + 1,
             4)) / (1 - NaturalMortality(3, 4)) + StockMort(Stk, 3) + StockMort(Stk + 1, 3)) / (1 -
             NaturalMortality(3, 3)) + StockMort(Stk, 2) + StockMort(Stk + 1, 2)) /
             (1 - NaturalMortality(3, 2)) + StockMort(Stk, 1) + StockMort(Stk + 1, 1)) / (1 -
             NaturalMortality(3, 1)) * InitialCohort(Stk) / (InitialCohort(Stk) + InitialCohort(Stk + 1))) /
             BaseCohortSize(Stk, Age)

Next BackFRAMIteration
```

Table 1. Backwards FRAM Target Run Size Definitions: Chinook

This table represents which Chinook FRAM fisheries (by FisheryID numeric values and generic titles) and time steps (horizontal) are included in the run size definition of each stock (by StockID numeric values and StockName) (vertical). A "yes" denotes that the landed catch of ages 3-5 Chinook in fishery and time steps are added to the age 3-5 run to the river (escapement + freshwater catch). T1 equals October-April time step, T2 equals May-June time step, and T3 equals July-September time step.

Stock Name	Stk #	25 Willapa Net	39,40 NT/T B'ham Bay Net	46,47 NT/T Skagit Net	48 8D Sport	49,50 NT/T 8A Net	51,52 NT/T 8D Net	58,59 NT/T 10,11 Net	60 10A Sport	61 10A Net	62 10E Sport	63 10E Net	65,66 NT/T HC Net	68,69 NT/T 13 Net	70,71 NT/T 13A Net	T1	T2	T3
Nooksack/Samish Fall	1, 2		yes															yes
NF Nooksack Spr	3,4,5,6		yes															yes
Skagit Summer/Fall Fing	7,8			yes													yes	yes
Skagit Summer/Fall Year	9,10			yes													yes	yes
Skagit Spring Year	11,12			yes													yes	yes
Snohomish Fall Fing	13,14																	yes
Snohomish Fall Year	15,16																	yes
Stillaguamish Fall Fing	17,18																	yes
Tulalip Fall Fing	19,20				yes		yes											yes
Mid PS Fall Fing	21,22								yes	yes	yes	yes						yes
UW Accelerated	23,24																	yes
South Puget Sound Fall Fing	25,26													yes	yes			yes
South Puget Sound Fall Year	27,28								yes	yes	yes	yes		yes	yes			yes
White River Spring Fing	29,30																yes	yes
Hood Canal Fall Fing	31,32												yes					yes
Hood Canal Fall Year	33,34												yes					yes
JDF Tribs. Fall	35,36																	yes
CR Oregon Hatchery Tule	37,38																	yes
CR Washington Hatchery Tule	39,40																	yes
Lower Columbia River Wild	41,42																	yes

Stock Name	Stk #	25 Willapa Net	39,40 NT/T B'ham Bay Net	46,47 NT/T Skagit Net	48 8D Sport	49,50 NT/T 8A Net	51,52 NT/T 8D Net	58,59 NT/T 10,11 Net	60 10A Sport	61 10A Net	62 10E Sport	63 10E Net	65,66 NT/T HC Net	68,69 NT/T 13 Net	70,71 NT/T 13A Net	T1	T2	T3
CR Bonneville Pool Hatchery	43,44																	yes
Columbia R Upriver Summer	45,46																yes	yes
Columbia R Upriver Bright	47,48																	yes
Cowlitz River Spring	49,50															yes		
Willamette River Spring	51,52															yes		
Snake River Fall	53,54																	yes
Oregon North Coast Fall	55,56																	yes
WCVI Total Fall	57,58																	yes
Fraser River Late	59,60																	yes
Fraser River Early	61,62																	yes
Lower Georgia Strait	63,64																	yes
White River Spring Year	65,66																yes	yes
Lower Columbia Naturals	67,68																	yes
Central Valley Fall	69,70																	yes
WA North Coast Fall	71,72																	yes
Willapa Bay	73,74	yes																yes
Hoko River	75,76																	yes
Mid OR Cst	77,78																	yes

Figure 1. Main Coho Backwards FRAM Run Menu

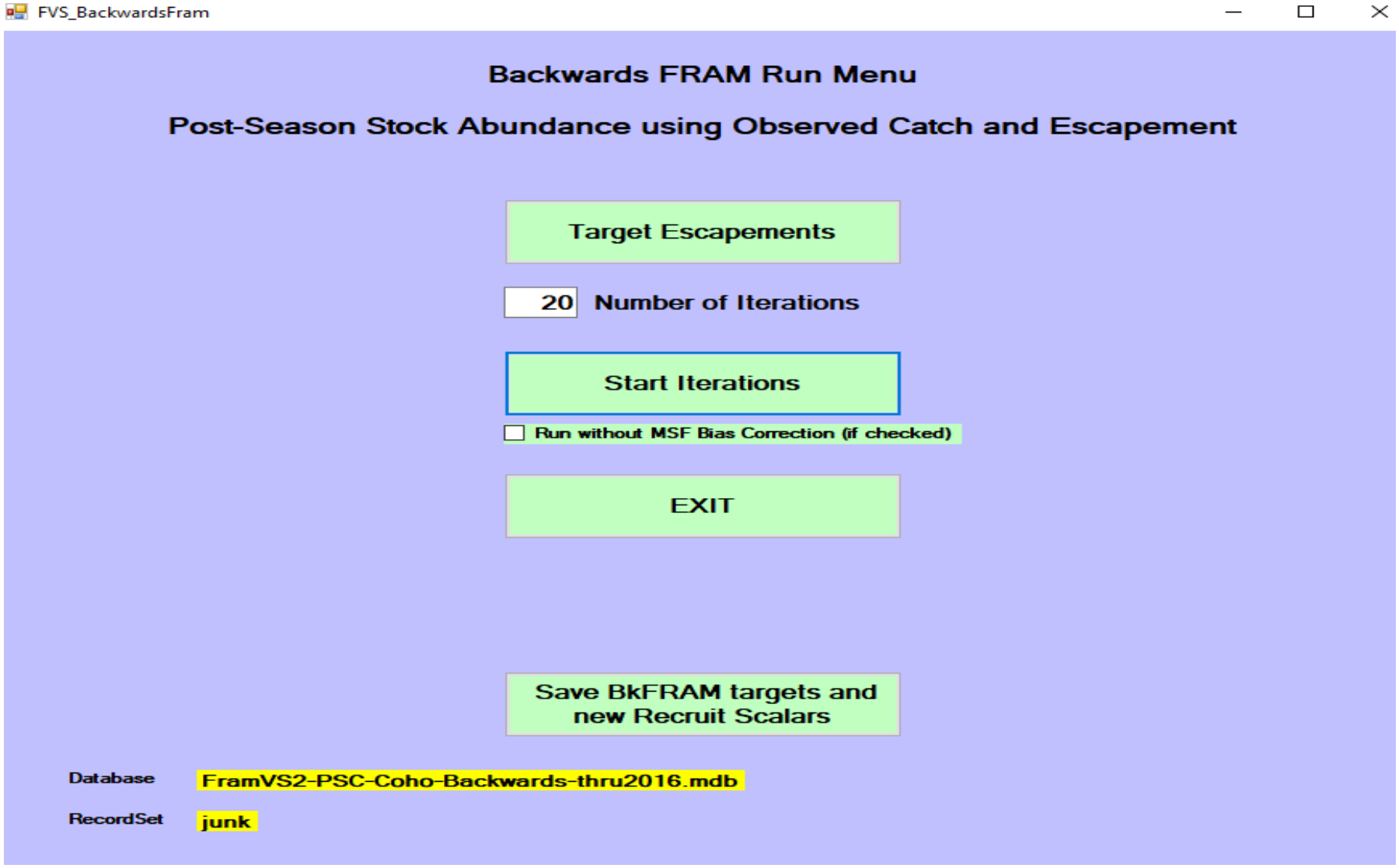
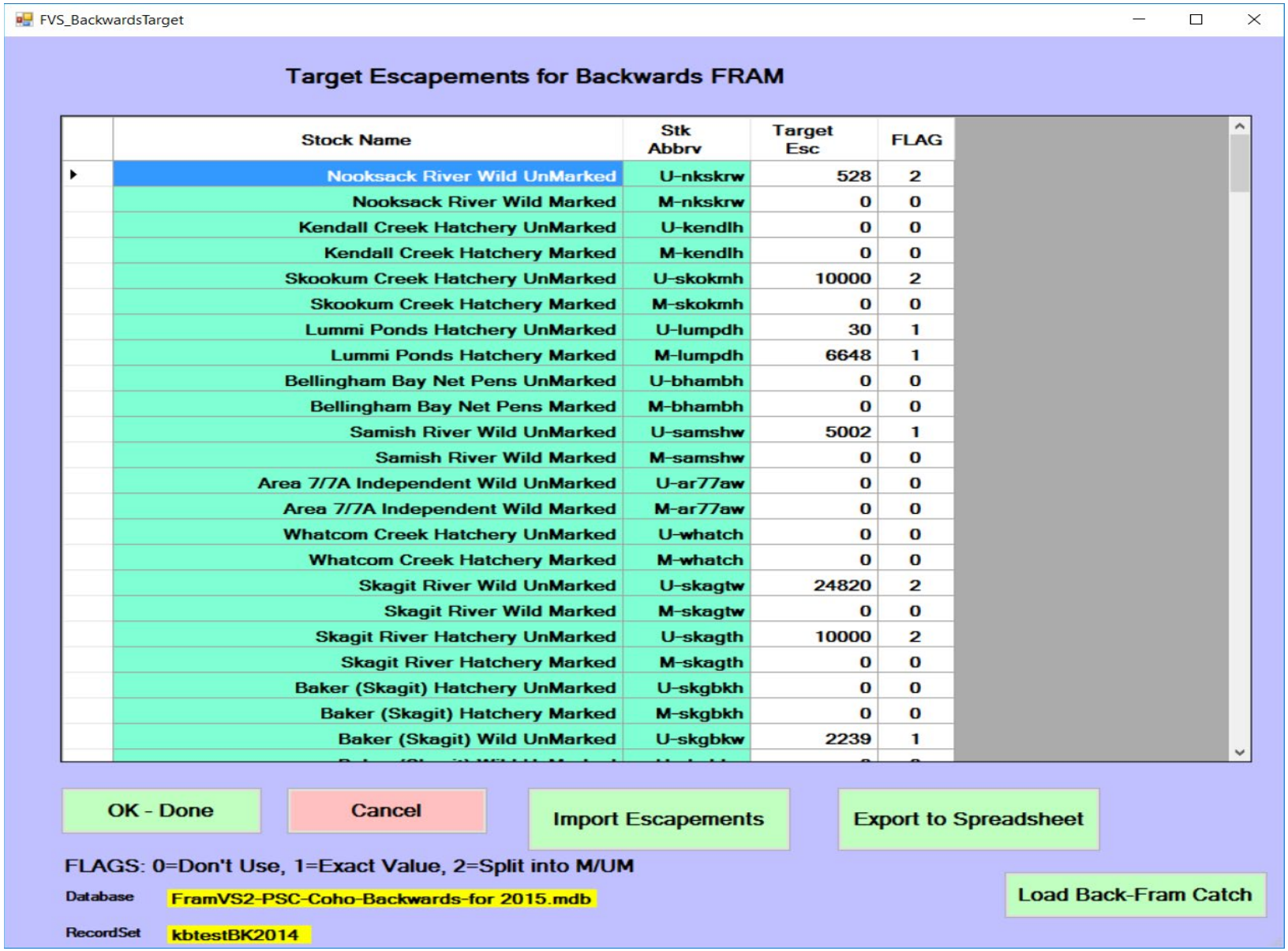
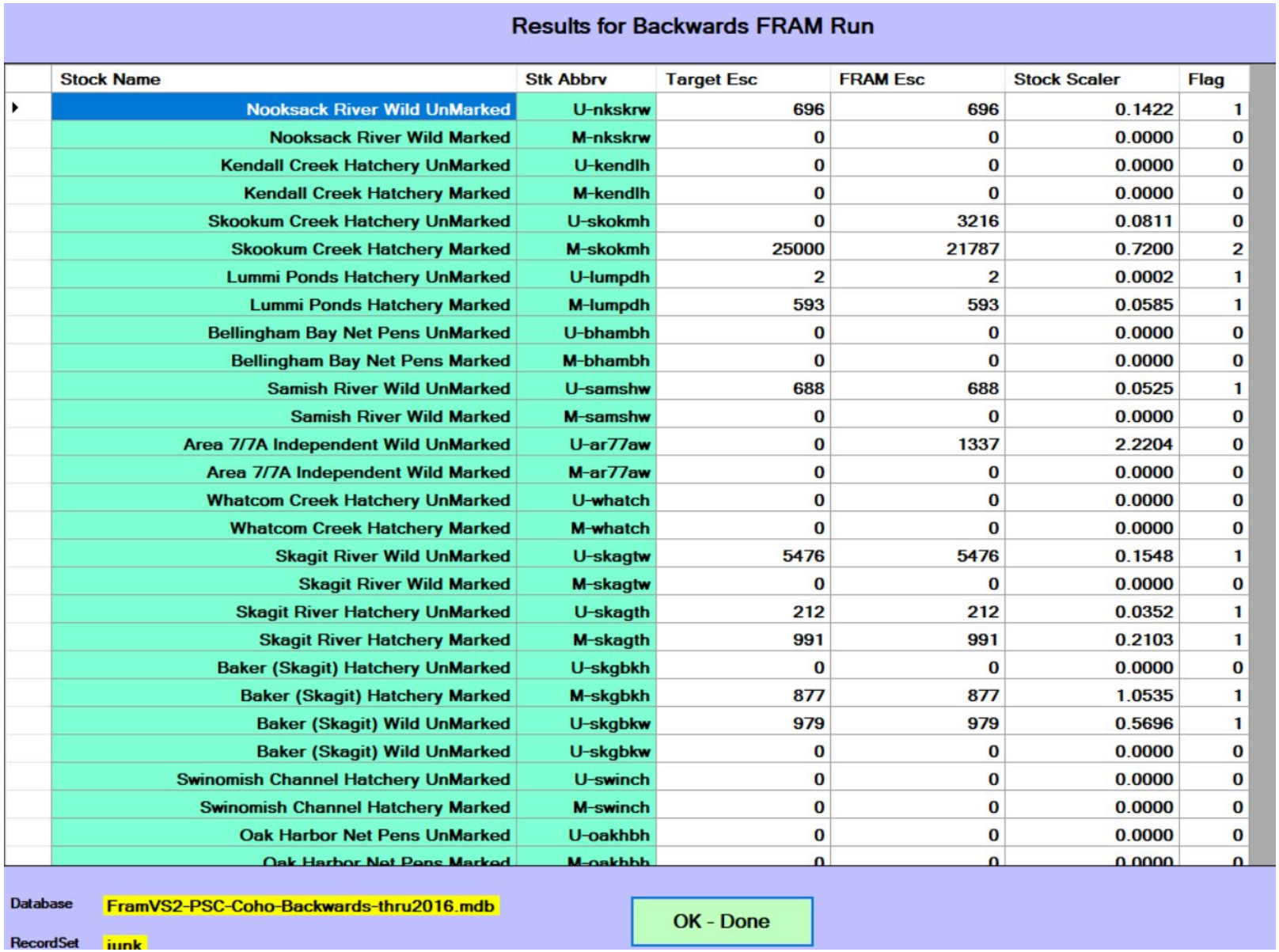


Figure 2. Coho Backwards FRAM Target Escapements Input Screen.



Stock Name	Stk Abbrv	Target Esc	FLAG
Nooksack River Wild UnMarked	U-nkskrw	528	2
Nooksack River Wild Marked	M-nkskrw	0	0
Kendall Creek Hatchery UnMarked	U-kendlh	0	0
Kendall Creek Hatchery Marked	M-kendlh	0	0
Skookum Creek Hatchery UnMarked	U-skokmh	10000	2
Skookum Creek Hatchery Marked	M-skokmh	0	0
Lummi Ponds Hatchery UnMarked	U-lumpdh	30	1
Lummi Ponds Hatchery Marked	M-lumpdh	6648	1
Bellingham Bay Net Pens UnMarked	U-bhambh	0	0
Bellingham Bay Net Pens Marked	M-bhambh	0	0
Samish River Wild UnMarked	U-samshw	5002	1
Samish River Wild Marked	M-samshw	0	0
Area 7/7A Independent Wild UnMarked	U-ar77aw	0	0
Area 7/7A Independent Wild Marked	M-ar77aw	0	0
Whatcom Creek Hatchery UnMarked	U-whatch	0	0
Whatcom Creek Hatchery Marked	M-whatch	0	0
Skagit River Wild UnMarked	U-skagtw	24820	2
Skagit River Wild Marked	M-skagtw	0	0
Skagit River Hatchery UnMarked	U-skagth	10000	2
Skagit River Hatchery Marked	M-skagth	0	0
Baker (Skagit) Hatchery UnMarked	U-skgbkh	0	0
Baker (Skagit) Hatchery Marked	M-skgbkh	0	0
Baker (Skagit) Wild UnMarked	U-skgbkw	2239	1

Figure 3. Coho Backwards FRAM Results Screen.



Results for Backwards FRAM Run

Stock Name	Stk Abbrv	Target Esc	FRAM Esc	Stock Scaler	Flag
Nooksack River Wild UnMarked	U-nkskrw	696	696	0.1422	1
Nooksack River Wild Marked	M-nkskrw	0	0	0.0000	0
Kendall Creek Hatchery UnMarked	U-kendlh	0	0	0.0000	0
Kendall Creek Hatchery Marked	M-kendlh	0	0	0.0000	0
Skookum Creek Hatchery UnMarked	U-skokmh	0	3216	0.0811	0
Skookum Creek Hatchery Marked	M-skokmh	25000	21787	0.7200	2
Lummi Ponds Hatchery UnMarked	U-lumpdh	2	2	0.0002	1
Lummi Ponds Hatchery Marked	M-lumpdh	593	593	0.0585	1
Bellingham Bay Net Pens UnMarked	U-bhambh	0	0	0.0000	0
Bellingham Bay Net Pens Marked	M-bhambh	0	0	0.0000	0
Samish River Wild UnMarked	U-samshw	688	688	0.0525	1
Samish River Wild Marked	M-samshw	0	0	0.0000	0
Area 7/7A Independent Wild UnMarked	U-ar77aw	0	1337	2.2204	0
Area 7/7A Independent Wild Marked	M-ar77aw	0	0	0.0000	0
Whatcom Creek Hatchery UnMarked	U-whatch	0	0	0.0000	0
Whatcom Creek Hatchery Marked	M-whatch	0	0	0.0000	0
Skagit River Wild UnMarked	U-skagtw	5476	5476	0.1548	1
Skagit River Wild Marked	M-skagtw	0	0	0.0000	0
Skagit River Hatchery UnMarked	U-skagth	212	212	0.0352	1
Skagit River Hatchery Marked	M-skagth	991	991	0.2103	1
Baker (Skagit) Hatchery UnMarked	U-skgbkh	0	0	0.0000	0
Baker (Skagit) Hatchery Marked	M-skgbkh	877	877	1.0535	1
Baker (Skagit) Wild UnMarked	U-skgbkw	979	979	0.5696	1
Baker (Skagit) Wild UnMarked	U-skgbkw	0	0	0.0000	0
Swinomish Channel Hatchery UnMarked	U-swinch	0	0	0.0000	0
Swinomish Channel Hatchery Marked	M-swinch	0	0	0.0000	0
Oak Harbor Net Pens UnMarked	U-oakhbh	0	0	0.0000	0
Oak Harbor Net Pens Marked	M-oakhbh	0	0	0.0000	0

Database FramVS2-PSC-Coho-Backwards-thru2016.mdb

RecordSet junk

OK - Done

Figure 4. Chinook Backwards FRAM Run Menu.

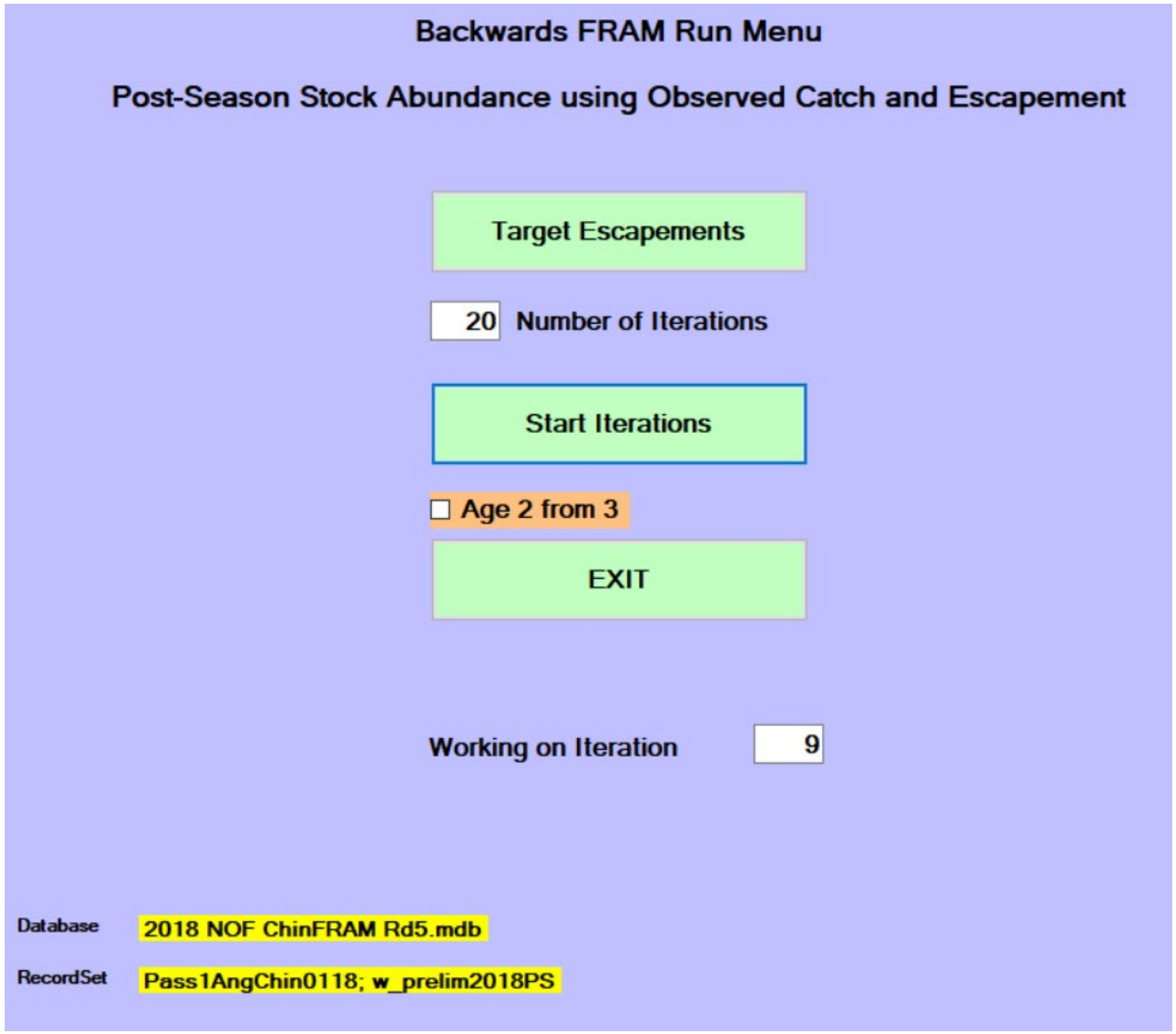
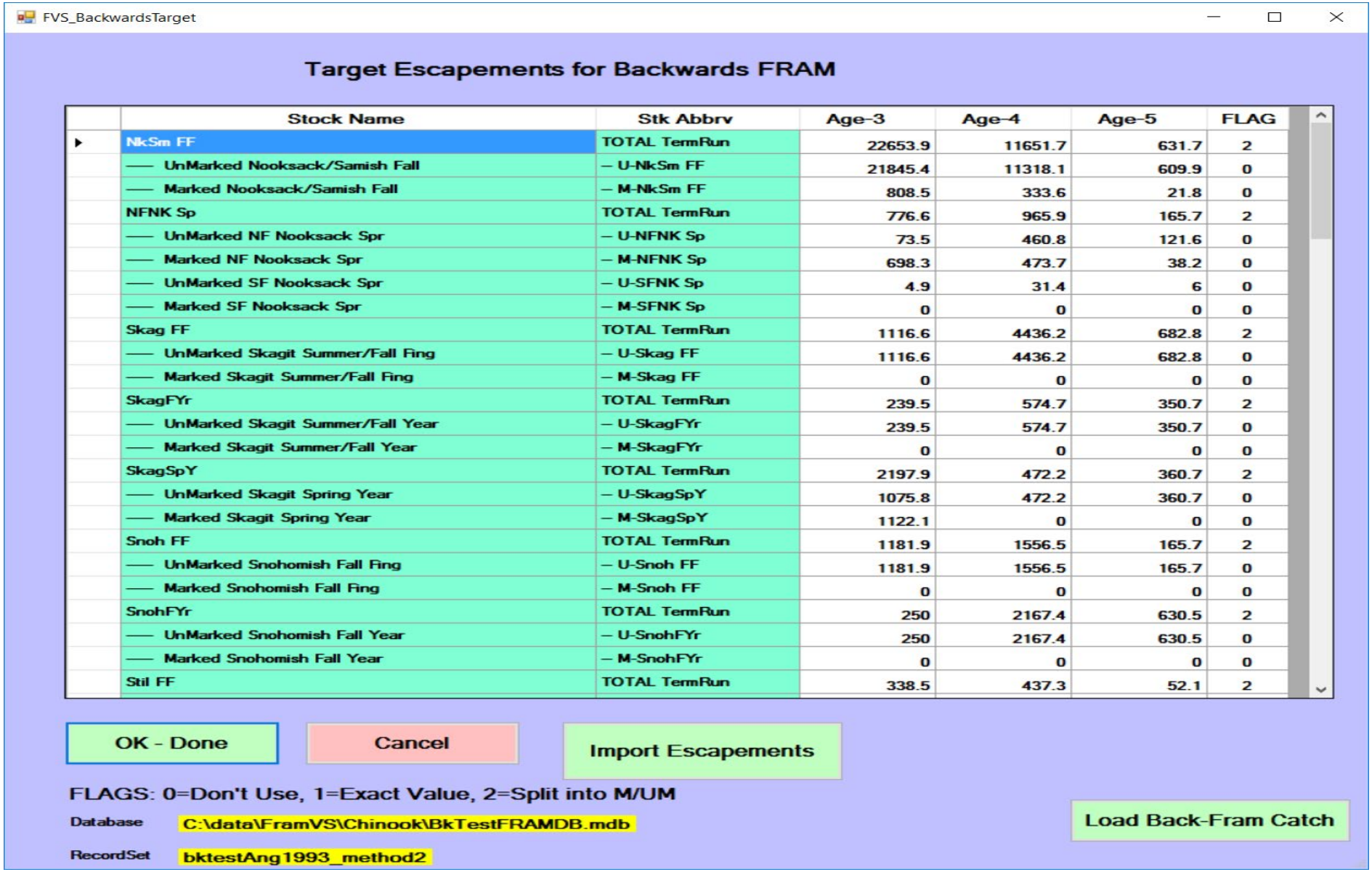
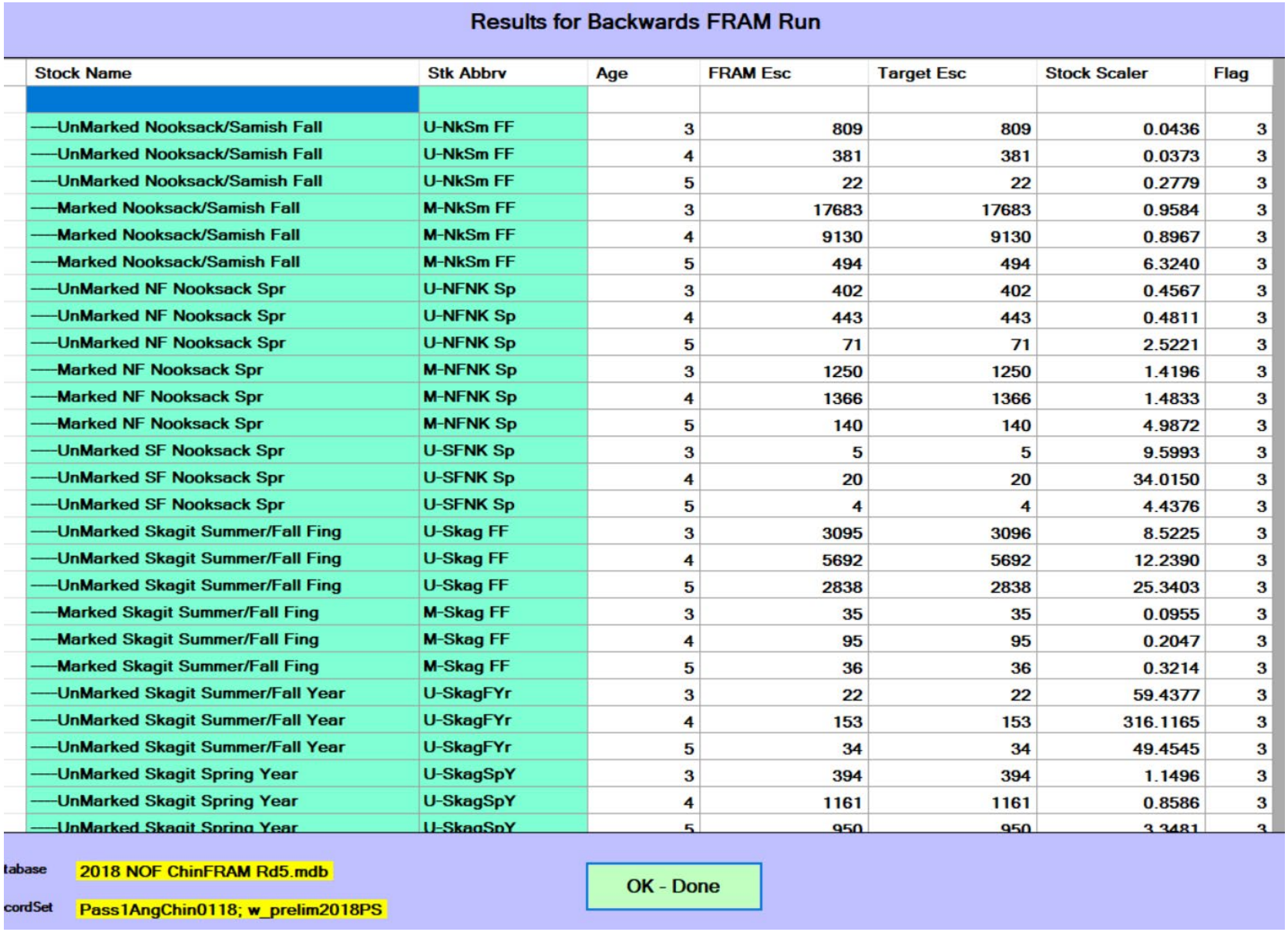


Figure 5. Chinook Backwards FRAM Target Escapements Input Screen.



Stock Name	Stk Abbrv	Age-3	Age-4	Age-5	FLAG
NkSm FF	TOTAL TermRun	22653.9	11651.7	631.7	2
—— UnMarked Nooksack/Samish Fall	– U-NkSm FF	21845.4	11318.1	609.9	0
—— Marked Nooksack/Samish Fall	– M-NkSm FF	808.5	333.6	21.8	0
NFNK Sp	TOTAL TermRun	776.6	965.9	165.7	2
—— UnMarked NF Nooksack Spr	– U-NFNK Sp	73.5	460.8	121.6	0
—— Marked NF Nooksack Spr	– M-NFNK Sp	698.3	473.7	38.2	0
—— UnMarked SF Nooksack Spr	– U-SFNK Sp	4.9	31.4	6	0
—— Marked SF Nooksack Spr	– M-SFNK Sp	0	0	0	0
Skag FF	TOTAL TermRun	1116.6	4436.2	682.8	2
—— UnMarked Skagit Summer/Fall Fing	– U-Skag FF	1116.6	4436.2	682.8	0
—— Marked Skagit Summer/Fall Fing	– M-Skag FF	0	0	0	0
SkagFYr	TOTAL TermRun	239.5	574.7	350.7	2
—— UnMarked Skagit Summer/Fall Year	– U-SkagFYr	239.5	574.7	350.7	0
—— Marked Skagit Summer/Fall Year	– M-SkagFYr	0	0	0	0
SkagSpY	TOTAL TermRun	2197.9	472.2	360.7	2
—— UnMarked Skagit Spring Year	– U-SkagSpY	1075.8	472.2	360.7	0
—— Marked Skagit Spring Year	– M-SkagSpY	1122.1	0	0	0
Snoh FF	TOTAL TermRun	1181.9	1556.5	165.7	2
—— UnMarked Snohomish Fall Fing	– U-Snoh FF	1181.9	1556.5	165.7	0
—— Marked Snohomish Fall Fing	– M-Snoh FF	0	0	0	0
SnohFYr	TOTAL TermRun	250	2167.4	630.5	2
—— UnMarked Snohomish Fall Year	– U-SnohFYr	250	2167.4	630.5	0
—— Marked Snohomish Fall Year	– M-SnohFYr	0	0	0	0
Stil FF	TOTAL TermRun	338.5	437.3	52.1	2

Figure 6. Chinook Backwards FRAM Results Screen.



Results for Backwards FRAM Run

Stock Name	Stk Abbrv	Age	FRAM Esc	Target Esc	Stock Scaler	Flag
---UnMarked Nooksack/Samish Fall	U-NkSm FF	3	809	809	0.0436	3
---UnMarked Nooksack/Samish Fall	U-NkSm FF	4	381	381	0.0373	3
---UnMarked Nooksack/Samish Fall	U-NkSm FF	5	22	22	0.2779	3
---Marked Nooksack/Samish Fall	M-NkSm FF	3	17683	17683	0.9584	3
---Marked Nooksack/Samish Fall	M-NkSm FF	4	9130	9130	0.8967	3
---Marked Nooksack/Samish Fall	M-NkSm FF	5	494	494	6.3240	3
---UnMarked NF Nooksack Spr	U-NFNK Sp	3	402	402	0.4567	3
---UnMarked NF Nooksack Spr	U-NFNK Sp	4	443	443	0.4811	3
---UnMarked NF Nooksack Spr	U-NFNK Sp	5	71	71	2.5221	3
---Marked NF Nooksack Spr	M-NFNK Sp	3	1250	1250	1.4196	3
---Marked NF Nooksack Spr	M-NFNK Sp	4	1366	1366	1.4833	3
---Marked NF Nooksack Spr	M-NFNK Sp	5	140	140	4.9872	3
---UnMarked SF Nooksack Spr	U-SFNK Sp	3	5	5	9.5993	3
---UnMarked SF Nooksack Spr	U-SFNK Sp	4	20	20	34.0150	3
---UnMarked SF Nooksack Spr	U-SFNK Sp	5	4	4	4.4376	3
---UnMarked Skagit Summer/Fall Fing	U-Skag FF	3	3095	3096	8.5225	3
---UnMarked Skagit Summer/Fall Fing	U-Skag FF	4	5692	5692	12.2390	3
---UnMarked Skagit Summer/Fall Fing	U-Skag FF	5	2838	2838	25.3403	3
---Marked Skagit Summer/Fall Fing	M-Skag FF	3	35	35	0.0955	3
---Marked Skagit Summer/Fall Fing	M-Skag FF	4	95	95	0.2047	3
---Marked Skagit Summer/Fall Fing	M-Skag FF	5	36	36	0.3214	3
---UnMarked Skagit Summer/Fall Year	U-SkagFYr	3	22	22	59.4377	3
---UnMarked Skagit Summer/Fall Year	U-SkagFYr	4	153	153	316.1165	3
---UnMarked Skagit Summer/Fall Year	U-SkagFYr	5	34	34	49.4545	3
---UnMarked Skagit Spring Year	U-SkagSpY	3	394	394	1.1496	3
---UnMarked Skagit Spring Year	U-SkagSpY	4	1161	1161	0.8586	3
---UnMarked Skagit Spring Year	U-SkagSpY	5	950	950	3.3481	3

tabase 2018 NOF ChinFRAM Rd5.mdb

cordSet Pass1AngChin0118; w_prelim2018PS

OK - Done

Table 2. Backwards FRAM Chinook Stocks

BkFRAM ID	Stk Name	FRAM ID	BkFRAM ID	Stk Name	FRAM ID	BkFRAM ID	Stk Name	FRAM ID
1	TOTAL TermRun	0	39	TOTAL TermRun	0	78	TOTAL TermRun	0
2	-- U-NkSm FF	1	40	-- U-SPS Fyr	27	79	-- U-Cowl Sp	53
3	-- M-NkSm FF	2	41	-- M-SPS Fyr	28	80	-- M-Cowl Sp	54
4	TOTAL TermRun	0	42	TOTAL TermRun	0	81	TOTAL TermRun	0
5	-- U-NFNK Sp	3	43	-- U-WhiteSp	29	82	-- U-Will Sp	55
6	-- M-NFNK Sp	4	44	-- M-WhiteSp	30	83	-- M-Will Sp	56
7	-- U-SFNK Sp	5	45	TOTAL TermRun	0	84	TOTAL TermRun	0
8	-- M-SFNK Sp	6	46	-- U-HdCl FF	31	85	-- U-Snake F	57
9	TOTAL TermRun	0	47	-- M-HdCl FF	32	86	-- M-Snake F	58
10	-- U-Skag FF	7	48	TOTAL TermRun	0	87	TOTAL TermRun	0
11	-- M-Skag FF	8	49	-- U-HdCl FY	33	88	-- U-OR No F	59
12	TOTAL TermRun	0	50	-- M-HdCl FY	34	89	-- M-OR No F	60
13	-- U-SkagFYr	9	51	TOTAL TermRun	0	90	TOTAL TermRun	0
14	-- M-SkagFYr	10	52	-- U-SJDF FF	35	91	-- U-WCVI Tl	61
15	TOTAL TermRun	0	53	-- M-SJDF FF	36	92	-- M-WCVI Tl	62
16	-- U-SkagSpY	11	54	TOTAL TermRun	0	93	TOTAL TermRun	0
17	-- M-SkagSpY	12	55	-- U-WhtSpYr	37	94	-- U-FrasRLt	63
18	TOTAL TermRun	0	56	-- M-WhtSpYr	38	95	-- M-FrasRLt	64
19	-- U-Snoh FF	13	57	TOTAL TermRun	0	96	TOTAL TermRun	0
20	-- M-Snoh FF	14	58	-- U-Hoko Rv	39	97	-- U-FrasREr	65
21	TOTAL TermRun	0	59	-- M-Hoko Rv	40	98	-- M-FrasREr	66
22	-- U-SnohFYr	15	60	TOTAL TermRun	0	99	TOTAL TermRun	0
23	-- M-SnohFYr	16	61	-- U-OR Tule	41	100	-- U-LwGeo S	67
24	TOTAL TermRun	0	62	-- M-OR Tule	42	101	-- M-LwGeo S	68
25	-- U-Stil FF	17	63	TOTAL TermRun	0	102	TOTAL TermRun	0
26	-- M-Stil FF	18	64	-- U-WA Tule	43	103	-- U-LColNat	69
27	TOTAL TermRun	0	65	-- M-WA Tule	44	104	-- M-LColNat	70
28	-- U-Tula FF	19	66	TOTAL TermRun	0	105	TOTAL TermRun	0
29	-- M-Tula FF	20	67	-- U-LCRWild	45	106	-- U-CentVal	71
30	TOTAL TermRun	0	68	-- M-LCRWild	46	107	-- M-CentVal	72
31	-- U-MidPSFF	21	69	TOTAL TermRun	0	108	TOTAL TermRun	0
32	-- M-MidPSFF	22	70	-- U-BPHTule	47	109	-- U-WA NCst	73
33	TOTAL TermRun	0	71	-- M-BPHTule	48	110	-- M-WA NCst	74
34	-- U-UWAc FF	23	72	TOTAL TermRun	0	111	TOTAL TermRun	0
35	-- M-UWAc FF	24	73	-- U-UpCR Su	49	112	-- U-Willapa	75
36	TOTAL TermRun	0	74	-- M-UpCR Su	50	113	-- M-Willapa	76
37	-- U-SPSd FF	25	75	TOTAL TermRun	0	114	TOTAL TermRun	0
38	-- M-SPSd FF	26	76	-- U-UpCR Br	51	115	-- U-MidORCst	77
			77	-- M-UpCR Br	52	116	-- M-MidORCst	78