`tagest & tagmove` User's Manual

John Sibert[1]
Joint Institute of Marine and Atmospheric Research
University of Hawai'i at Manoa
Honolulu, HI 96822 U.S.A.

February 7, 2015

[1]sibert@hawaii.edu

see TagestUserMan.tex for macro definitions

Introduction

Work on this software began in the late 1980s using DOS computers with 16-bit word length and segmented architecture. As a result there are quite a few "legacy" issues such as odd (i.e. short and cryptic) file names and certain hard coded restrictions in the code. Previous versions of this software had been compiled using Borland, Zortech and gnu compilers under Windows and gnu compilers under linux. Currently the only supported compiler is g++ under linux, but porting to MinGW under Windows is probably feasible.

Adam and Sibert (2004) describe an extension of this software that uses a neural network to compute the movement and mortality parameters from fields of environmental variables. This documentation is largely confined to the "regional" model.

Installation

Environment Variables

Set these to the appropriate paths for your computer in `~/.bashrc`.

```
export JAVA_HOME=/usr/local/java
export JNIGRAPHICS_PATH=/usr/local/jnigraphics
export LD_LIBRARY_PATH=/usr/local/java/jre/lib/amd64:/usr/local/java/jre/lib/amd64
export ADMB_HOME=/usr/local/admb
export ETOPO_HOME=/home/other/etopo
export OSTYPE
export HOSTNAME
```

Preprocessors

The `tagest & tagmove` software expects data in specific formats. The preprocessors read data from original sources and create files for direct input to `tagest & tagmove`.

Discretization errors occur inevitably in the numerical solution of differential equations. In `tagest & tagmove`, these errors are manifested when fishing occurs in the same computational element as a coastline. Ocean modelers often address this problem by increasing the spatial resolution of their models or by adjusting the land mask to shift the problematical stretch of coastline. These options are available in `tagest & tagmove`. In addition, the preprocessors off the opportunity to shift 'events" (fishing effort, tag release or tag recapture) that are reported to have occurred on "land" to nearby positions in the ocean. The shift list file, discussed below, enables position shifting to occur in a consistent way for all fishing effort, tag release or tag recapture data.

Preprocessor input files

Note the geographic position refers to the southwest corner of the geographic square.

file_nam.dat This short, 4-line, file contains a list of file name roots to be used by the preprocessors. Here is an example from an SPC application.

```
skj16
recaps_skj
dates_pt
fleet_ps
```

- **skj16** The first record is the root file name used for all preprocessor output files including the `.par` file for `tagest & tagmove`. The name should reflect the specific model domain.
- **recaps_skj** The second record is the root name for the tag release and recapture file to which the "`.dat`" will be appended. In this example the preprocessor will look for a file named for a file named `recaps_skj.dat` in the parent directory.
- **dates_pt** The third record specifies a file containing the starting and ending dates to be included in the analysis. This file contains two records, the first record contains the starting year and month, e.g. `2005 01` and the second record contains the ending year and month in the same format, e.g. `2012 12`. This example will cause the preprocessor to create `tagest & tagmove` data files encompassing the period from January 2005 through December 2012.

fleet_ps The fourth record specifies the name of a file containing the fleets to be considered in the analysis. For example `fleet_ps.dat` contains 18 records like "`JPPS_PT`" which tells the preprocessors to look for files like `JPPS_PT.dat` in the parent directory.

dates_pt.dat As specified in `file_nam.dat`, starting and ending dates for the analysis

```
2005 01
2014 12
```

fleet_ps.dat As specified in `file_nam.dat`, list of fleet names used in the analysis. Here is an example with 18 fleets.

```
CNPS_PT
EPPS_PT
ESPS_PT
FMPS_PT
IDPS_PT
JPPS_PT
KIPS_PT
KRPS_PT
MHPS_PT
NZPS_PT
PGPS_PT
PHPS_PT
SBPS_PT
SVPS_PT
TVPS_PT
TWPS_PT
USPS_PT
VUPS_PT
```

skj16.prn This file contains a complete description of the model domain for the identifier specified in the first record of `file_nam.dat`. The first record is a code indicating the file version code. Currently the only valid value of this code is "`#v20r`". This file contains the number of grid cells in the longitude and latitude dimensions, the spatial resolution, the southwest corner of the, the minimum number of months a liberty for each cohort, the specific starting year and month, number of tag releases (probably ignored), and the number of fleets. This file also contains the "grid map" or land mask consisting of an integer for each cell in the model domain. A '0' in any cell indicates land and integers > 0 indicate the region number. The final record in the file indicates the boundary conditions imposed on west, east, south and north boundaries; '1' indicates reflective, '0' indicates open.

In the following example `.prn` file, the model domain specifies 80 cells in the longitude dimension and 35 cells in the latitude dimension. The lines have been shortened from 80 cells to 10 for readability, but there are 35 lines. The spatial resolution is 1 degree (60 nautical miles), and the southwest corner is 115.0E $\times$ 20.0°S. **Degrees are always decimal degrees, never in degrees and minutes.**

```
#v20r
#   M    N  deltax  deltay  sw_long sw_lat
   80   35     60     60     115E    20S
#  nmonth  start_year  start_month  nrelease  nfleet
     12       2006          8           5000      18
# grid map
#   1    2    3    4    5    6    7    8    9   10  ...
#
    2    2    2    2    2    2    0    2    2    2  ...
    2    2    2    2    2    2    2    0    0    2  ...
    2    2    2    2    2    2    2    2    2    2  ...
    2    2    2    2    2    2    2    0    2    2  ...
    2    2    2    2    0    2    2    2    0    2  ...
    2    2    2    0    2    2    2    2    2    2  ...
    2    2    0    2    2    2    2    2    2    2  ...
    2    2    2    2    2    2    2    2    2    0  ...
    0    0    0    2    2    2    2    2    2    0  ...
    0    0    0    2    2    2    2    2    2    2  ...
    0    0    0    0    2    2    2    2    2    2  ...
    0    0    0    2    2    2    2    2    2    2   ...
    0    0    0    2    2    2    2    2    2    2  ...
    0    0    0    0    2    2    2    2    2    2  ...
    0    0    0    2    2    0    0    0    0    3  ...
    0    0    0    3    0    3    3    3    3    3  ...
    0    0    0    3    0    0    0    0    3    3  ...
    0    0    0    3    0    0    0    3    3    3  ...
    0    0    0    3    0    3    0    0    3    3  ...
    0    0    3    3    0    3    3    0    3    3  ...
    0    0    3    3    0    3    3    3    3    3  ...
    0    3    3    3    3    3    3    3    3    3  ...
    3    3    3    3    3    3    3    3    3    3  ...
    0    0    0    0    0    0    0    0    0    0  ...
```

```
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
          0    0    0    0    0    0    0    0    0    0  ...
#
#  boundary conditions:
#  west_bndry    east_bndry    south_bndry    north_bndry
       1             1              1              1
```

skj16.pos This file contains the position shifting list. It is automatically generated by `makeifr` and `makepar`. If it is absent, it is created interactively. If it is present, it is augmented as additional positions are added. Each record consists of longitude latitude pairs indicating "from" and "to" for the position shifts. Normally the user will not need to edit this file since it is generated and maintained automatically by the preprocessors. In the example below, all "events" (fishing effort, tag release or tag recapture) that occur at position 146°E×7°S, which is on land as defined by the land mask, will be shifted to position 146°E×6°S.

```
146E 7S 146E 6S
135E 4S 135E 3S
```

skj16.cst A file of closed polygons used to map coastlines by the jnigraphics library in `tagmove`. It is generated by running `gmt2cst`.

`makemap`

This program will generate a "land mask", a matrix of 1s and 0s indicating water and land respectively. Input to `makemap` is the first line of the `.prn` file mentioned above. The software reads the etopo5[1] data base, file `etopo5.dos` in a directory pointed to by the `ETOPO_HOME` environment variable. It scores a

[1]www.ngdc.noaa.gov/mgg/global/etopo5.HTML

computational element as being on land if the number of 5 minute geographic squares in the data base is greater than 0. By default `makemap` generates a land mask for each of three threshold elevations, 0m, -10m and -100m, and writes each land mask on the file `makemap.log`. The land mask can be copied from the `.log` file into the relevant part of the `.prn` file for input into other preprocessors.

The land mask is the starting point for the "gridmap" section of the `.par` files. Perhaps the easiest way to create a gridmap is to import the land mask into a spreadsheet to assign appropriate region numbers to the ocean (non-zero) cells.

`makeifr`

This program builds the indexed regional fishery record data structure (the "ifr") and saves it to a file. Fishing does not occur in every possible cell in the model domain. Fishing data are, therefore, relatively sparse, and it is inefficient to store effort data for the entire domain for every fleet and every date. The ifr is indexed by year, month and fleet so that by requesting effort data for a specific combination of year, month and fleet, it is possible to quickly obtain a matrix of fishing effort over the model domain.

Change to the directory where the preprocessor input files are stored and type "makeifr" at the prompt. The program will start running, and if data reporting effort on land are encountered, you will be prompted to enter a alternate position to which to shift the effort. The `.pos` file is updated on normal termination. If the `.pos` file is not present on startup, `makeifr` will issue a message and terminate. The workaround for this problem is to create an empty `.pos` file, either by using the linux command `touch` or by using a text editor.

`makepar`

This program builds the recapture, (`.tag`) and the `.par` files. It uses the information in the `.pos` file to shift release and recapture positions consistently with the shifting in the ifr. If additonal shift positions are required, you will be prompted in the same was as in `makeifr`. If additional shifting does occur, the `.pos` file is updated, and you should run `makeifr` again to ensure that tag positions and effort positions are consistently shifted.

gmt2cst

Creates a `.cst` file for use by the simulators.

The following script will invoke GMT to extract coastline data for a model domain bounded by 115°E×29°S and 195°E×14°N and save the points to the file `grid16gmt.dat`

```
#!/bin/bash
REGION=-R115/195/-20/15
echo $REGION
PROJECTION=-JM6i
echo $PROJECTION
#extract coastline data for gmt2cst
pscoast $REGION $PROJECTION -Di -W -M > grid16gmt.dat
```

Then type `gmt2cst  grid16gmt.dat 115 -20 60 35` to build a `.cst` file for a 60×35 model domain for use by the simulators. This software is not yet completely reliable.

Simulation

rtagmove

Implementation of ADRM tag movement simulator.

walk

Implementation of biased random walk simulation using the same input files as `rtagmove`

Estimation

tagest

Generalized ADModel Builder implementation of ADRM estimation procedure.

`rtagest25`

Original version of ADRM estimation simulator with adjoint code. Supplanted by the ADMB version and no longer supported. This version has not been kept up to date with the ADMB version but may be useful in some circumstances for testing.

Postprocessors

`obstac`

Working Directories

Project

Model Domain

Fits

File Structure

"Raw" Data

Effort Data

Effort data records consist of 7 fields. From left to right the fields are

1. year/month - 4 byte year, slash, two byte month with leading zero.
2. latitude - legitimate floating point number **in decimal degrees, no minutes or seconds** with 'N' or 'S' appended.
3. longitude - legitimate floating point number with 'E' or 'W' appended.
4. effort
5. catch
6. gear type
7. flag

```
2005/01 05.5S 164.0E 0.5 3.7 PS JP
2005/01 00.5S 159.0E 0.2 2 PS JP
2005/01 02.5S 162.0E 1.8 80 PS JP
2005/01 01.5S 162.0E 3 115 PS JP
2005/01 00.5N 162.0E 0 0 PS JP
2005/01 08.5S 163.0E 0 0 PS JP
2005/01 03.5S 163.0E 0.8 13.7 PS JP
2005/01 01.5S 163.0E 0.2 25 PS JP
```

Tag Files

Tag release and recapture data are combined in a single file.

Release records consist of 4 fields, from left to right:

1. year/month - 4 byte year, slash, two byte month with leading zero.
2. latitude - same format as effort.
3. longitude - same format as effort.
4. number of tagged fish released.

The release record is followed by a list of recaptures from the release cohort. Release records consist of 6 fields, from left to right:

1. tag number - any ascii string.
2. year/month of recapture.
3. latitude - same format as effort.
4. longitude - same format as effort.
5. gear - same format as effort.
6. flag - same format as effort.

It is OK for fields to be missing or entered as "NA"

```
2006/08 06.50S 154.00E 36
                       P00017 2006/11 NA NA PS TW
                       P00020 2006/12 NA NA PS SY
                       P00023 2007/01 NA NA PS PH
                       P00005 2007/04 NA NA PS CN
2006/08 05.50S 149.00E 430
```

```
P03938 2006/09 05.50S 149.00E PS PG
P03883 2006/09 05.50S 149.00E PS PG
P03886 2006/09 05.50S 149.00E PS PG
P03887 2006/11 04.00S 148.50E PS PG
P03890 2006/09 05.50S 148.50E PS PG
P03903 2006/09 05.50S 149.00E PS PG
```

Domain-specific Data

`.par` Files

Model options

Likelihood, control flag 11

Several likelihood functions are available in `tagest`. The most useful is the Poisson (set control flag 11 to 2). This option appears to be applicable in most applications. Many of the other likelihood options are either deprecated or are of limited use.

$$L = \frac{\widehat{C}^C e^{-\widehat{C}}}{C!}, \tag{1}$$

where C is the observed number of tag returns and $\widehat{C}$ is the predicted number of tag returns.

The only other useful likelihood option is the negative binomial which has several different forms. Setting control flag 11 to 24 will invoke the negative binomial likelihood estimating a separate dispersion parameter for each fishing fleet. Setting control flag 11 to 25 will invoke the negative binomial likelihood with a separate dispersion parameter for each fishing fleet, but the dispersion parameter will be held constant at to value in the `.par` file and not estimated. Use of the negative binomial likelihood requires on additional field in the `.par` file for each fleet.

Cohort pooling, control flag 13

A tag "cohort" is defined as all of the tags released during one time step in a computational element of the model domain, for example, all of the tags released in a 1° geographic square during one month. Typically there are 100s of tag cohorts in an analysis. The likelihood for each cohort would ideally be computed separately for each cohort and summed over all cohorts, but such an approach would produce impossibly long derivative chains and would not be a feasible computation.

One alternative is to pool the cohorts by releasing tags into a single pooled cohort at the appropriate time and place in the model domain. The likelihood is computed only once. Set control flag 13 to 1 for pooled cohorts.

A second alternative is to aggregate the release cohorts in to "monthly" release units. Under this option, the likelihood is computed separately for each monthly release unit and summed over all release units. Set control flag 13 to 2 to use monthly cohorts.

The option not to pool the cohorts (flag 13 set to 0) is used in the simulator to compute the "predicted" recaptures for a tag attrition curve.

Months after last cohort, `nmonth`

`tagest` is essentially a tag attrition model. The `nmonth` variable in the `.par` file specifies the number of months after the release of the last tag cohort to continue the likelihood computation. The optimum time to continue the computation appears to be between 24 and 36 months.

ADI iterations per month, control flag 6

Although the ADI algorithm is unconditionally stable, transient instabilities occasional occur if there are large spatial gradients in tag density such as in the first one or two months after release of a large cohort. Such cases are indicated by a banded patter in the tag density display in `tagmove`. The default value is 4 ADI iterations per month. Increasing this number to 6 usually removes the problem (but also increases the length of the derivative chain).

"Special" mortality, control flag 19

This option is a kludge to model unique mortality, for instance on younger fish, in some regions. The default value, 0, implies no special morality. Set the value of control flag 19 to the region number for which it is necessary to estimate a unique mortality. This option was developed for the Philippines domestic yellowfin fishery where large numbers of juvenile fish were tagged and it was hypothesized that the natural mortality might be higher for younger fish. Use of this option requires an additional field in the `.par` file.

Catchability estimation, control flag 25

Setting this flag to 2 (the default) causes `tagest` to estimate a unique catchability coefficient, q_f, for each fleet. Setting the flag to 1, causes `tagest` to assume that catchability if constant over all fleets.

Flag 25	Action
0	Catchability not estimated; constant at input values.
1	Catchability assumed equal for all fleets; $q_f = q_1; f = 2 \ldots n$.
2	Unique catchability coefficient estimated for each fleet.

Reporting rate estimation, control flag 28

Flag 28	Action
0	Reporting rate not estimated; constant at input values.
1	Unique reporting coefficient estimated for each fleet.

Generally speaking reporting rates are confounded with catchability coefficients so that it usually impossible to estimate both. If independent estimates of reporting rate are available, the should be used, and flag 27 set to zero.

In some cases, it may be possible to assume uniform catchability over all fleets, such as the FAD-based purse seine fishery. In such cases it may be possible to estimate reporting rates by setting flag 25 to 1; see Table 1.

Baranov catch equation, control flag 27

This option switches the computation of predicted tag recapture between the simple approximation (0) and the Baranov (1) form:

Flag 27	Action
0	$\widehat{C}_{ijtf} = F_{ijtf} P_{ijt}$
1	$\widehat{C}_{ijtf} = F_{ijtf} P_{ijt} \frac{1-e^{-(F_{ijtf}+M)}}{F_{ijtf}+M}$

Table 1: Results of attempts to estimate reporting rate for skipjack in the purse seine fishery. In fit p05, reporting rate is held constant at 0.5 (a figure similar to that reported by Hoyle, 2011) over all fleets, and a unique catchability is estimated for each fleet. In fit p02, a single catchability and separate reporting rates are estimated for each fleet. The negative log likelihood decreases from 20342 to 20340 with an addition of a single parameter, a significant improvement at the 95% confidence level.

Fit	Flag 25	Flag 28	Flag 53	n	M	q_{jpps}	R_{jpps}	$-\log L$
p05	2	0	2	95	0.073	0.0047	**0.5**	20342
p02	1	1	2	96	0.063	0.063	0.325	20340
p03	1	1	1	96		*no solution*		
p04	1	1	0	96	0.168	0.0265	1.0	16846

Table 2: Comparision of liklihood values and estimates of natural mortality and Japanese purse seine catchability coefficient.

Fit	Flag 27	Flag 53	$-\log L$	$\widehat{M}$	$\hat{q}_{jpps}$
11	0	0	16864	0.185	0.028
12	0	1	17335	0.182	0.028
14	0	2	20372	0.099	0.020
0b	1	0	16821	0.185	0.034
0f	1	1		*no solution*	
10	1	2	20327	0.094	0.024

From the results presented in Table 2 use of the Baranov form has little effect on the parameter estimates. Further, its use in conjunction of flag 53 requires the Barnov equation be solved for F by some iterative method. To date the Newton-Raphson method has not been notably useful.

Normalize Fishing Effort, control flag 51

This option normalizes the fishing effort for each fleet to the mean fishing effort for that fleet. Control flag 52 is set by the software and indicates whether the catchability coefficients have been appropriately rescaled. The user should not manually change control flag 52.

Let E_{fijt} be the reported fishing effort for fleet f in computational element (1° square) (ij) in month t. Let $n_f = |E_{fijt} > 0|$, the total number of time×area strata where fishing effort was reported. Then

$$\overline{E_f} = \frac{\sum_{ijt} E_{fijt}}{n_f} \qquad (2)$$

is the mean effort for fleet f, and the normalized fishing effort is

$$E'_{fijt} = \frac{E_{fijt}}{\overline{E_f}}. \qquad (3)$$

The average normalized fishing effort is 1 for all fleets.

Fishing mortality for use in the catch equations and total mortality is computed as

$$F_{ijtk} = q_k \cdot E'_{ijtk}, \qquad (4)$$

and `tagest` estimates q_k appropriately. This transformation appears to stabilize the estimation of q_k. The average fishing mortality in a any 1° square, i. e. the fishing mortality resulting from application of average effort, is thus simply q_k.

Sibert and Hampton (2003) used normalized fishing effort to compute average fishing mortality for each fleet. "Average fishing mortality rate was computed by multiplying the estimated catchability coefficients by the proportion of the total fished area exploited by each fleet."

$$\overline{F_k} = q_k \cdot \frac{n_l}{\sum_l n_l} \qquad (5)$$

where n is defined as above.

Effortless recaptures, control flag 53

Tags are occasionally reported from positions and dates for which no fishing effort has been reported. The causes of such reports are not entirely clear, but there are several possibilities: (1) simple errors in recording dates and positions; (2) tag recovered under circumstances where it is difficult or impossible to infer the data and position of recapture (from the well of a purse seiner at unloading); (3) delays in reporting fishing effort. Lack of fishing effort makes it impossible to compute fishing mortality and the predicted tag recaptures such that $\widehat{C}$ in equation (1) is zero. Thus, lack of fishing effort has no direct effect on the likelihood. Lack of fishing effort (and fishing mortality), as an indirect effect on the likelihood through the population of tags a liberty. The effortless recaptures are not removed from the population because fishing mortality computed to be zero for these recaptures. The tagged fish are remain in the population and are reflected in the likelihood through an inflated number of predicted recaptures at other locations and times. The result can be estimates of M biased upward and estimates of q biased downward.

Effortless tag recapture has been a persistent feature of the tag recapture data since 1977 ranging from 5% to 20% of recaptures; see Table 3. Most of the effortless recaptures can be attributed to specific fleet, which appear to have been dilatory in reporting their fishing effort. In the case of the PTTP recaptures, the proportion of effortless recaptures is highest in 2010. This result indicates that the probable cause is delay in receiving and processing fishing effort data from the fishing fleets (Table 4). If so, the the proportion of effortless recaptures will likely decrease with time.

Early versions (through svn revision 2761) of `tagest & tagmove` attempted to correct the effortless recapture problem by shifting the observed recapture position to an adjacent computational element. The strategy is fairly effective provided there is nearby fishing effort, but ignores the problem of misspecified dates.

Current versions (post revision 2830) set a flag in the recapture record as it is read into the software to indicate whether fishing effort is recorded for the tag return. There are three options for accommodating effortless recaptures controlled by Control Flag 53:

Table 3: Historical development of effortless recaptures by fishing fleet. The numbers for the SSAP and RTTP are based on the same model domain used in Sibert and Hampton (2003). The numbers for the PTTP are from a different model domain.

Fleet	SSAP	RTTP	PTTP
fjpl	367	7	
jppl	42	35	
kipl		8	
pgpl	3		
sbpl	2	42	
cnps			17
fmps			22
idps			110
jpps	1	22	33
kips			2
krps		23	23
mhps			1
pgps			72
phps		169	224
sbps		108	267
twps		13	48
usps	1	41	73
vups			778
phdo		17	
iddo		6	
Without E	461	491	1674
With E	1926	9161	9062
	19.3%	5.1%	15.6%

Table 4: Proportion of effortless recaptures over the duration of a tagging project. "NA" indicates no recaptures reported during the month. "0" indicates no effortless recaptures. The time span indicated by the number of years indicates the total period at liberty over which the `tagest` likelihood was evaluated.

Year	Month of Year											
	1	2	3	4	5	6	7	8	9	10	11	12
1977	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0	0
1978	NA	0.00435	0	0.00694	0	0	0.844	0.97	0.795	0.731	0.0435	0.0192
1979	0.0455	0.222	0.133	0	0	0.00505	0.0139	0.00862	0	0	0	0.0952
1980	0.125	0.312	0.167	0.206	0.0625	0	0	0.0588	0	0	0.333	0
1981	0.2	0.2	0.333	NA	1	0	0	0.5	NA	NA	NA	NA
1982	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1983	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1984	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1989	NA	NA	NA	NA	NA	NA	0	0.00518	0	0.00909	0.0625	0
1990	0	0	0.202	0.082	0.0423	0.107	0.0249	0.028	0.00676	0.0222	0.0726	0.125
1991	0.125	0.0251	0.0174	0.0533	0.121	0.0863	0.0234	0.124	0.0341	0.13	0.154	0.199
1992	0.152	0.0506	0.0161	0.0353	0.0918	0.127	0.0155	0.00253	0.0226	0.00245	0.015	0.0916
1993	0.0116	0.0588	0.037	0	0	0	0	0	0.143	0.111	0	0.5
1994	0	0	0	NA	0	NA	NA	NA	NA	NA	NA	NA
1995	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1996	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
2006	NA	NA	NA	NA	NA	NA	NA	0	0.00277	0	0	0.0562
2007	0.107	0.0556	0.0896	0.248	0.0259	0.0503	0.0483	0.0417	0.31	0.0227	0.234	0.115
2008	0.67	0.39	0.306	0.185	0.143	0.233	0.0704	0.308	0.123	0.0769	0.121	0.131
2009	0.331	0.357	0.111	0.134	0.0959	0.339	0.112	0.376	0.221	0.207	0.297	0.0667
2010	0.259	0.394	0.714	0.167	0.667	0	0	1	1	1	1	NA
2011	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Flag 53	Action
0	Recaptures without reported fishing effort are ignored.
1	Fishing mortality set to value required to produce the observed tag recaptures given the population density of tags. If $C = F \times P$, then $F = C/P$. (Not currently working for Baranov catch equation, flag 27.)
2	Fishing effort assumed to equal average fishing effort for fleet. $F = q_f \times \bar{E}_f$.

These options have several effects on the model as shown Table 2. The larger number of recaptures in the model cause the negative log likelihood to be larger. Estimates of natural mortality ($\widehat{M}$) are slightly lower and estimates of Japanese purse seine catchability ($\hat{q}_{jpps}$) are slightly higher. The results with respect to Control Flag 27 are discussed elsewhere. (For the purposes of this comparison, the models were fit using normalized fishing effort and assuming uniform reporting rates over all fleets equal to 1.0.)

Control Flags

The control flags, usually known as `ipars` control some of the details of the model structure, such as the likelehood, ADI parameters, and which parameters are estimated (ie "active"). The listing below documents all of the `ipars`. The default value for most flags is 0 unless noted below.

Flag 1	Maximum number of function evaluations; default = 1000.
Flag 2	Iterations between minimizer displays; default = 1.
Flag 3	Convergence criterion * 1000; default = 100.
Flag 4	Maximum number of iterations; default = 30.
Flag 5	Number of calls made to fgcomp during fit; default = 0.
Flag 6	if ¿0, number of time steps per month; default = 4.
Flag 7	fmm.scroll_flag; 1 ⇒ scroll ALL variables; 0 ⇒ overwrite screen; default = 0.
Flag 8	fit number (usually decimal equivalent to the hex file name extention number).
Flag 9	total elapsed minutes to do fit).
Flag 10	fmm.ireturn.

Flag 11	likelihood function: 0 ⇒ testlk; 1 ⇒ robust_likelihood; 2=¿ poisslk; 3=¿ least_square; 4,14,24=¿ negative binomial; 5,15,25=¿ negative binomial const.; 6=¿ exponential); // 4,14 24:24 is the ipar value.. 4 is for min and max and 24 is for initial values.. one for each fleet in the model; default = 2.
Flag 13	tag cohort pooling; 0 ⇒ no pooling; 1 ⇒ pooling; 2 ⇒ pooling by date; default = 1.
Flag 14	0 ⇒ omit month at liberty n from from likelihood (resurrected, but still bogus).
Flag 16	global average effort for simultor to generate effort files; default = 0.
Flag 17	global effort SD for simultor to generate effort files; default = 0.
Flag 18	1 ⇒ scale NB parameter by predicted recaptures; default = 0.
Flag 19	0 ⇒ no special M for any region (default); otherwise region number for special motality estimation; default = 0.
Flag 20	1 ⇒ use Bills corrected total mortality.
Flag 21	sigma active; 1 ⇒ uniform sigma; 2 ⇒ separate sigma for each region; default = 2.
Flag 22	natural mortality active; default = 1.
Flag 23	1 ⇒ (u,v) active; default = 1.
Flag 24	number of fishing mortality anomalies; deprecated; default = 0.
Flag 25	q active; 1 ⇒ uniform q; 2 ⇒ separate q for each fleet; default = 2.
Flag 26	No longer used!; default = 0.
Flag 28	reporting rate active; default = 0.
Flag 29	total tag penalty *1e8.
Flag 30	-log_10 tag scaling factor; 0 ⇒ no scaling.
Flag 31	x component curvature penalty * 1e9; default = 30.
Flag 32	y component curvature penalty * 1e9; default = 30.
Flag 33	velocity curvature penalties in effect; default = 1.
Flag 34	near zero velocity smoothing scale * 100; default = 5.

Flag 35	number of points in velocity moving average smoothing function; default = 3.
Flag 36	number of points in diffusion moving average smoothing function; default = 0.
Flag 37	100 * constant smimming speed.
Flag 38	tau and lambda amplification factor for random walk simulator.
Flag 39	effort standard deviation for simulator.
Flag 40	random number seed for simulator.
Flag 41	sigma curvature penalties in effect; default = 1.
Flag 42	sigma curvature penalty * 1e9; default = 30.
Flag 43	set PHDO effort to zero if no recaptures.
Flag 44	set IDDO effort to zero if no recaptures.
Flag 45	use monthly average effort; no longer used?.
Flag 46	bin width for displacement calculations; 0 $\Rightarrow$ 60 NMi; default = 60.
Flag 47	non-zero $\Rightarrow$ saves the non-zero viewport as a gif(pcx) file.
Flag 48	1 $\Rightarrow$ write tags as real numbers.
Flag 49	1 $\Rightarrow$ use centered scheme; 0 $\Rightarrow$ use upwind scheme; default = 0.
Flag 50	0 $\Rightarrow$ use 60 in all of deltax; 1 $\Rightarrow$ create spherical coords.
Flag 51	1 $\Rightarrow$ normalize fishing effort to fleet means.
Flag 52	DO NOT EDIT; value set automagically (1 $\Rightarrow$ catchabilities have been adjusted to normalized effort).
Flag 53	include effortless recaps in Z: 0 $\Rightarrow$ omit; 1 $\Rightarrow$ f = inverse catch equation; 2 $\Rightarrow$ f= q barE.
Flag 54	reserved for possible use in effortless recap handling.
Flag 56	southwest longitude of recruitment area; 0 $\Rightarrow$ model area.
Flag 57	southwest latitude of recruitment area; 0 $\Rightarrow$ model area.
Flag 58	northeast longitude of recruitment area; 0 $\Rightarrow$ model area.
Flag 59	northeast latitude of recruitment area; 0 $\Rightarrow$ model area.
Flag 60	total recruitment in recruitment area; 0 $\Rightarrow$.

Flag 61	population maximum for history plot.
Flag 62	catch maximum for history plot.
Flag 63	maximum for population for color scale.
Flag 64	0 $\Rightarrow$ display matrix only; 1 $\Rightarrow$ display the coastline of the region and matrix; 2 $\Rightarrow$ display coastline only.
Flag 65	0 $\Rightarrow$ continuous run; 1 $\Rightarrow$ use keyboard for next frame.
Flag 66	0 $\Rightarrow$ use no scaling for uvd; 1 $\Rightarrow$ use scales from scale.dat for uvd read in by read_seasons.
Flag 67	1 $\Rightarrow$ save the no. of releases /month /gridsquare.
Flag 68	1 $\Rightarrow$ graphics on (default); 0 $\Rightarrow$ graphics off; default = 1.
Flag 69	1 $\Rightarrow$ reads in seasonal info from year_XX.uvd files.
Flag 70	movement pattern display mode: 0 $\Rightarrow$ circles and arrows; 1 $\Rightarrow$ color map and arrows.
Flag 71	u_slope: 1 $\Rightarrow$ active, 0 $\Rightarrow$ inactive.
Flag 72	u_offset: – ditto–.
Flag 73	v_slope: – ditto–.
Flag 74	v_offset: – ditto–.
Flag 75	sigma_slope: – ditto–.
Flag 76	sigma_offset: – ditto–.
Flag 77	1 $\Rightarrow$ D^2 before scaling; 2 $\Rightarrow$ D^2 after scaling.
Flag 78	NN weight penalty in effect; 1 $\Rightarrow$ penalize first layer only; 2 $\Rightarrow$ only penalize hidden layer.
Flag 79	Weight penalty*1.e-9.
Flag 80	0 $\Rightarrow$ apply scaling function to output layer; 1 $\Rightarrow$ no scaling.
Flag 81	1 $\Rightarrow$ switch_penalty in effect; 0 $\Rightarrow$ not in effect.
Flag 82	Switch_penalty *1.e-9.
Flag 83	FAD factor model code; 0 $\Rightarrow$ do not read; 1 $\Rightarrow$ read FADmap.
Flag 84	FAD-uv factor model code; 0 $\Rightarrow$ no effect; 1 $\Rightarrow$ effect by uv computation.
Flag 85	choice of FAD-D model code; 0 $\Rightarrow$ no effect; 1 $\Rightarrow$ FAD-D logistic model; 2 $\Rightarrow$ FAD-D exponential model.
Flag 86	FAD-uv factor model code-using chi; -1 $\Rightarrow$ no effect;1 $\Rightarrow$ effect by uv computation.

Flag 87	FAD-D exponential model code-using gamma; -1 $\Rightarrow$ no effect;1 $\Rightarrow$ effect by sigma computation.
Flag 88	FAD-D logistic model code-using logslope; -1 $\Rightarrow$ no effect;1 $\Rightarrow$ effect by sigma computation.
Flag 89	FAD-D logistic model code-using inflection; -1 $\Rightarrow$ no effect;1 $\Rightarrow$ effect by sigma computation.
Flag 90	FAD-D logistic model code-using min_D: -1 $\Rightarrow$ no effect;1 $\Rightarrow$ effect by sigma computation

References to `tagest` & `tagmove`

Adam, M.S., Sibert J., 2002. Population dynamics and movements of skipjack tuna (Katsuwonus pelamis) in the Maldivian fishery: analysis of tagging data from an advection-diffusion-reaction model. Aquat. Living Resour. 15: 13-23.

Adam, M. S. and J. Sibert. 2004. Use of neural networks with advection-diffusion-reaction models to estimate large-scale movements of skipjack tuna from tagging data. SOEST Publication 04-03, JIMAR Contribution 04-350, 31 pp.

Bills, P. J. and J. Sibert, 1997. Design of tag-recapture experiments for estimating yellowfin tuna stock dynamics, mortality, and fishery interactions. SOEST Publication 97-05, JIMAR Contribution 97-313, 80 pp.

Sibert, J., J. Hampton, D. Fournier, and P. J. Bills. 1999. An advection-diffusion-reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (Katsuwonus pelamis). Can. J. Fish. Aquat. Sci. 56: 925-938.

Sibert, J., Hampton, J., 2003. Mobility of tropical tunas and the implications for fisheries management. Marine Policy 27: 87-05.

Model Equations

et $\widetilde{N}_{xytc}$ represent the density of tagged tunas (numbers of tagged fish per unit surface area) at point (x, y) in the ocean at time t of tag release cohort c. The aggregate density of tagged tunas (or simply, tags) from all cohorts released up to time t is given by

$$N_{xyt} = \sum_{c=1}^{C_t} \widetilde{N}_{xytc}. \tag{6}$$

Assuming that the tag cohorts move independently, the aggregate tag density satisfies the following partial differential equation

$$\frac{\partial N}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial N}{\partial x}\right) + \frac{\partial}{\partial y}\left(D\frac{\partial N}{\partial y}\right) - \frac{\partial}{\partial x}(uN) - \frac{\partial}{\partial y}(vN) - ZN. \tag{7}$$

Tuna movement patterns are frequently represented by arrows on maps, often with months or seasons specified, to suggest the general trend of population movement at different times and places. This practice implies that fish movement may be time and site specific. Consistent with this possibility, in our model, "regions" are defined as subdivisions of the model domain over which the movement parameters (uvD) are constant, and "seasons" are defined as periods of time during which the parameters within a region are constant. Let R_{ij} be a matrix that contains the region number for each model cell indexed by (i, j) and S^n be a vector that contains the season for each time step indexed by n. In other words, R_{ij} maps the model domain into specified regions and S^n maps calendar time to seasons. The model parameters are specified at each grid point by the following equations:

$$\begin{aligned} u_{ij}^n &= \mathbf{u}_{R_{ij}S^n} \\ v_{ij}^n &= \mathbf{v}_{R_{ij}S^n} \\ D_{ij}^n &= \mathbf{D}_{R_{ij}S^n} \end{aligned} \tag{8}$$

where $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{D}$ are matrices of parameters to be estimated, collectively referred to as movement "patterns" or "hypotheses".

Total mortality, Z, is separated into two components in a conventional manner by

$$Z_{ij}^n = M + \sum_f F_{ijf}^n \tag{9}$$

where F_{ijf}^n is the mortality due to fishing by fishing fleet f operating in computational element (i, j) during time step n and M is mortality due to other causes or "natural" mortality. Natural mortality is assumed be constant at all places over the period of time that the tagged fish are at liberty. Fishing mortality is assumed to be a simple function of observed fishing effort

$$F_{ijf}^n = q_f \times E_{ijf}^n \tag{10}$$

The predicted number of reported tags during one month is given by

$$\widehat{C}_{ijf}^n = \beta_f F_{ijf}^n N_{ij}^n \tag{11}$$

where β_f is the reporting rate, i.e., the proportion of tags captured by fleet f returned with usable recapture information and N_{ij}^n satisfies equation 7 for time step n.

Parameters are estimated by maximum likelihood using AD Model Builder (Fournier, at al 2011). Observed numbers of tag returns, C_{ijf}^n, are related to predicted numbers of tag returns, $\widehat{C}_{ijf}^n$, by a Poisson likelihood function.

$$L(\mathbf{u}, \mathbf{v}, \mathbf{D}, q, M \mid C_{ijf}^n, E_{ijf}^n) = \prod_{ijnf} \left[\frac{(\widehat{C}_{ijf}^n)^{C_{ijf}^n} e^{-\widehat{C}_{ijf}^n}}{C_{ijf}^n!} \right] \tag{12}$$