# ABT-MSE: AN R PACKAGE FOR ATLANTIC BLUEFIN TUNA MANAGEMENT STRATEGY EVALUATION

T. Carruthers[1] and D. Butterworth[2]

*SUMMARY*

*Software for developing and testing management procedures is presented including worked examples*[*].

*RÉSUMÉ*

*Le logiciel pour développer et tester les procédures de gestion est présenté, y compris des exemples détaillés*[*].

*RESUMEN*

*Se presenta el programa informático para desarrollar y probar procedimientos de ordenación y se incluyen los ejemplos desarrollados.*[*]

*KEYWORDS*

*Management Strategy Evaluation, bluefin tuna, operating model, management procedure, software*

[1] IOF, 2202 Main Mall, University of British Columbia, Vancouver, B.C., Canada, V6T 1Z4; t.carruthers@oceans.ubc.ca

[2] Department of Mathematics and Applied Mathematics, University of Cape Town, Rondebosch 7701, South Africa. doug.butterworth@uct.ac.za

[*] The development of the MSE for North Atlantic bluefin tuna remains an ongoing process at this time. This document is therefore not final, but relates to the extent of development of the work immediately prior to the September 2017 bluefin session which preceded the ICCAT SCRS meeting.

[*] Le développement de la MSE pour le thon rouge de l'Atlantique Nord reste à ce stade un processus en cours. Ce document n'est donc pas définitif, mais se rapporte à l'ampleur du développement des travaux immédiatement avant la session sur le thon rouge de septembre 2017 qui a précédé la réunion du SCRS de l'ICCAT.

[*] En este momento prosigue el proceso de desarrollo de la MSE para el atún rojo del Atlántico norte. Por tanto, este documento no es final, sino que está relacionado con la medida de desarrollo de este trabajo inmediatamente antes de la sesión de atún rojo de septiembre de 2017, que precedió a la reunión del SCRS de ICCAT.

## 1. Introduction

A Management Strategy Evaluation (MSE, Butterworth 1999, Cochrane 1998) approach has been proposed for Atlantic bluefin tuna as a suitable framework for providing robust management advice consistent with the precautionary approach (GBYP 2017a).

A critical step in MSE is the development of candidate management procedures (CMPs) which can provide management advice from fishery data. MSE processes are strengthened by comparative testing of multiple CMPs developed by scientists. To facilitate this, an R MSE package has been developed to enable design and testing of CMPs for Atlantic bluefin tuna (ABT-MSE).

In this paper a series of worked examples demonstrate how the R framework may be used to test CMPs. A comprehensive user guide (Carruthers 2017) is available from a GitHub repository where all code and data are also freely available. A brief installation guide is included in the Appendix of this document. For a full description of operating model equations and parameters we refer users to the Trial specifications document (CMG 2017) and other supporting papers (Carruthers *et al.*, 2016). See GBYP (2017b) for a summary of the data used by the operating models.

## 2. Methods

### Format of simulated data

In the ABT-MSE framework, CMPs must access simulated data and provide a TAC recommendation. Various data are simulated and stored in an object *dset*, that can be accessed by CMPs (**Table 1**). The principal data types that may be used by MPs are provided for both East and West management areas and include previous TAC recommendations and observed relative abundance indices.

CMPs often use indices of relative abundance as the primary basis for adjusting the TAC. In total 7 indices have at present been agreed as potential inputs to be simulated in the ABT-MSE framework (Carruthers and Butterworth, in press a) (**Table 2**).

### MP design

In the ABT-MSE framework, management procedures are functions that have two arguments, the first is the simulation number x, the second is the simulated data set. There are two remaining requirements, the first is that the last line of the MP function is the TAC recommendation (a point value) and that immediately after the MP it is assigned the class 'MP'. Two simple constant catch MPs are provided in **Figure 1**, an example of an index target MP (EMP1, Carruthers and Butterworth, in press b) is provided in **Figure 2**.

### MP testing

Before attempting to apply an MP in the MSE you can test it using simulated data to check for errors (e.g. **Figure 4**). A number of example datasets are included in the ABT-MSE package for testing purposes.

### Running an MSE and calculating performance

In relatively few lines an MSE can be run and performance plotted and saved to disk (**Figure 5**).

## 3. Acknowledgements

This work was carried out under the provision of the ICCAT Atlantic Wide Research Programme for Bluefin Tuna (GBYP), funded by the European Union, several ICCAT CPCs, the ICCAT Secretariat and by other entities (see: http://www.iccat.int/GBYP/en/Budget.htm). The contents of this paper do not necessarily reflect the point of view of ICCAT or other funders and in no ways anticipate ICCAT future policy in this area.

**References**


Butterworth, D.S., Punt, A.E., 1999. Experiences in the evaluation and implementation of management procedures. ICES J. Mar. Sci. 56, 985-998.

Carruthers, T.R. 2017. ABT-MSE: Atlantic Bluefin Tuna Management Strategy Evaluation (v2.1). ICCAT Atlantic Wide Research Programme for Bluefin Tuna (GBYP). User guide available at: https://htmlpreview.github.com/?https://github.com/ICCAT/abft-mse/blob/master/ReadMe.html

Carruthers, T.R., Butterworth, D. (in press a). Summary of a Reference Set of Conditioned Operating Models for Atlantic Bluefin Tuna. Document SCRS/2017/223: 18 p.

Carruthers, T.R., Butterworth, D. (in press b). Performance of Example Management Procedures for Atlantic Bluefin Tuna. Document SCRS/2017/224: 10 p.

Carruthers, T.R., Kimoto, A., Powers, J., Kell, L., Butterworth, D., Lauretta, M. and Kitakado, T. 2016. Structure and estimation framework for Atlantic bluefin tuna operating models. Col. Vol. Sci. Pap. ICCAT, 72(7): 1782-1795.

CMG. 2017. Specifications for MSE trials for bluefin tuna in the North Atlantic. GBYP Core Modelling Group. ICCAT Atlantic Wide Research Programme for Bluefin Tuna. Available at: https://github.com/ICCAT/abft-mse/tree/master/Manuals_and_design_documents/Trial Specifications.docx [accessed September 2017]

Cochrane, K L., Butterworth, D.S., De Oliveira, J.A.A., Roel, B.A., 1998. Management procedures in a fishery based on highly variable stocks and with conflicting objectives: experiences in the South African pelagic fishery. Rev. Fish. Biol. Fisher. 8, 177-214.

GBYP. 2017a. ICCAT Atlantic Wide Research Programme for Bluefin Tuna. Available online at: http://www.iccat.int/GBYP/en/index.htm [accessed September 2017]

GBYP. 2017b. Data to inform operating models for North Atlantic bluefin tuna. ICCAT Atlantic Wide Research Programme for Bluefin Tuna. Available at: https://drive.google.com/drive/folders/0B0TXcs-MLRl3anc2Sjc0Yjk1ZTA [accessed September 2017]

**Table 1**. Principal simulated data in the simulated dataset object.

| Name | Description | Dimensions |
|---|---|---|
| Cobs | Observed annual catches | sim x year |
| TAC | Historical TAC recommendations | sim x year |
| Iobs | Observed relative abundance indices | sim x index x year |
| CAA | Catch-At-Age samples | sim x age x year |

**Table 2.** The indices simulated the MSE framework.

| No | Name | Area | Type | Description |
|---|---|---|---|---|
| 1 | JPN_LL_NEAtl2 | East | Fishery-dependent | Japanese Longline in the North East Atlantic |
| 2 | FR_AER_SUV | East | Fishery-independent | French Aerial Survey |
| 3 | MED_LAR_SUV | East | Fishery-independent | Mediterranean Larval Survey |
| 4 | MED_AER_SUV | East | Fishery-independent | Mediterranean Aerial Survey |
| 5 | JPN_LL2 | West | Fishery-dependent | Japanese longline in the western Atlantic |
| 6 | US_RR_66_114 | West | Fishery-dependent | US Rod and Reel 66cm – 114cm West Atlantic |
| 7 | GOM_LAR_SUV | West | Fishery-independent | Gulf of Mexico Larval Survey |

```
Const_Cur_TAC = function(x, dset){          # Calculate TAC from simulated data dset for simulation x
  dset$TAC[x, 1]                            # TAC is set to the first ever (current, 2016) TAC level
}
class(Const_Cur_TAC) = "MP"                 # Assign Const_Cur_TAC a class 'MP'


MeanCat <- function(x, dset){               # Calculate TAC from simulated data dset for simulation x
  mean(dset$Cobs[x, ])                      # TAC is set to the mean historical observed catches
}
class(MeanCat) = "MP"                       # Assign MeanCat a class 'MP'
```

**Figure 1.** Two constant catch MPs. Management procedures are functions that must have two arguments, the first of which is the simulation number *x*, the second is the simulated data, *dset*. The first MP 'Const_Cur_TAC' sets the new TAC recommendation to the first ever (current) TAC recommendation for simulation *x*. The second 'MeanCat' is simply the mean historical annual catches for simulation *x*.

```r
EMP1 = function(x, dset){                          # Calculate TAC from simulated data dset for simulation x

  Jtarg = 4.8                                      # Index target level
  ny = dim(dset$Iobs)[3]                           # Last year of index observations

  Jmu = mean(dset$Iobs[x, 1, (-4:0)+ny])           # Mean of index 1 (JPN_LL_NEAtl2) over last five years
  Jratio = Jmu/Jtarg                               # Ratio of current mean index / target

  cury = dim(dset$TAC)[2]                          # Last year of past TAC recommendations
  previousTAC =  dset$TAC[x, cury]                 # Get previous TAC for simulation x

  if(Jratio > 0.6 & Jratio < 1.4){                # If Jratio is greater than 0.6 and less than 1.4

    TAC = previousTAC                              # No change in TAC

  }else if(Jratio < 0.6){                          # If Jratio is less than 0.6

    TAC = previousTAC * 0.9                        # New TAC is 10% lower than previous TAC

  }else{                                           # If Jratio is greater than 1.4

    TAC = previousTAC * 1.1                        # New TAC is 10% greater than previous TAC

  }

  TAC                                              # Last line of MP is the TAC recommendation

}

class(EMP1) = "MP"                                 # Assign EMP1 a class 'MP'
```

**Figure 2**. Example Management Procedure 1 represented in R code. Management procedures are functions that must have two arguments, the first of which is the simulation number *x*, the second is the simulated data, *dset*. The last line of every MP function in the ABT-MSE framework must be the TAC recommendation. The MP must also be assigned the right class 'MP' after the function is defined.

```r
EMP2 <- function(x, dset,                          # Calculate TAC from simulated data dset for simulation x
                 IndexNo = 11, Jtarg = 0.6,        # Index is #11, (GOM_LAR_SUV), target index level is 0.6
                 lup = 0.05, ldown = 0.15,         # TAC change fraction of slope in index
                 pup = 0.05, pdown = 0.15){        # TAC change fraction of ratio of recent index to Jtarg

  ny = dim(dset$Iobs)[3]                           # Last year of index observations
  Ind = dset$Iobs[x,1,(-5:0)+ny]                   # Last six years of index observations

  linmod = lm(y ~ x, data = data.frame(y = log(Ind), x = 1:6)) # fit a log-linear model
  slp = linmod$coefficients[2]                     # log-linear slope in index

  Jratio = mean(dset$Iobs[x, IndexNo, (-4:0)+ny]) / Jtarg  # Ratio of recent Index / Jtarg

  cury = dim(dset$TAC)[2]                          # Last year of past TAC recommendations
  previousTAC = dset$TAC[x, cury]                  # Get previous TAC for simulation x

  if(slp > 0){                                     # If index slope is positive
    smod = lup*slp
  }else{                                           # If index slope is negative
    smod = ldown*slp
  }

  if(Jratio > 1){                                  # If recent mean Index is greater than Jtarg
    Jmod = pup*(Jratio-1)
  }else{                                           # If recent mean Index is less than Jtarg
    Jmod = pdown*(Jratio-1)
  }

  Tmod<-Jmod+smod                                  # Total TAC modification

  if(Tmod > 0.15) Tmod = 0.15                      # Maximum upward change is 15%
  if(Tmod < (-0.15)) Tmod = -0.15                  # Maximum downward change is 15%

  previousTAC*(1+Tmod)                             # Adjust previous TAC

}

class(EMP2e)<-"MP"                                 # Assign EMP1 a class 'MP'
```

**Figure 3**. Example Management Procedure 2 represented in R code.

```
library(ABTMSE)                                  # Load library
loadABT()                                        # Load all the package data

nsim = nrow(dset_example_East$TAC)               # Get the number of example simulations

sapply(1:nsim, EMP1, dset = dset_example_East) # Make sure EMP1 works with an example dataset

sapply(1:nsim, EMP2, dset = dset_example_West) # Make sure EMP1 works with an example dataset
```

**Figure 4.** MP testing.

```
library(ABTMSE)                             # Load library
loadABT()                                   # Load all the package data
sfInit(parallel = T, cpus=detectCores()) # Start up the cluster for parallel computing

MPs = list(c("MeanCat", "MeanCat"),      # First MP is mean historical catches in the East and West
           c("EMP1",    "EMP2"))         # Second MP is EMP1 in the East and EMP2 in the West

myMSE = new("MSE", OM_1, MPs=MPs)        # Run MSE with OM_1

plot(myMSE)                                 # Projection plot
PPlot(myMSE)                                # Performance plot
Tplot(myMSE)                                # Trade-off plot
perf = getperf(myMSE)                       # Calculate the mean performance tables

write.csv(perf[[1]], "C:/East_perf.csv") # Write the eastern performance table to disk
write.csv(perf[[2]], "C:/West_perf.csv") # Write the western performance table to disk

save(myMSE, "C:/temp/myMSE.Rdata")       # Save the MSE object
```

**Figure 5.** Running an MSE and plotting results.

***Software installation***

Download and install the latest version of R:

https://cran.r-project.org/bin/windows/base/

Download and install the latest version of RStudio:

https://www.rstudio.com/products/rstudio/download/#download

***Package installation***

Save the library file 'ABTMSE_2.1.0.tar.gz' to disk and then install from the R prompt in RStudio

```
> install.packages("C:/Downloads/ABTMSE_2.1.0.tar.gz", repos = NULL, type="source")
```

***Required at the start of each R session***

```
> library(ABTMSE)                                  # load the ABT-MSE library
> loadABT()                                        # load all of the data objects
> sfInit(parallel = TRUE, cpus = detectCores())    # setup multicore processing
```

***Check package installation***

```
> checkMSE = new('MSE')                            # run a test MSE
> plot(checkMSE)                                   # plot the results
```

***Getting help***

```
> readme()                                         # open the user guide in your internet browser
> class?MSE                                        # get help on a class of ABTMSE objects
> class?OM
```

***Finding objects***

```
> avail('OM')                                      # list all of the available operating models
> Design                                           # examine the design of the reference operating models
```