GMSE: an R package for generalised management strategy evaluation

Supporting Information 5

A. Bradley Duthie[1,3], Jeremy J. Cusack[1], Isabel L. Jones[1], Erlend B. Nilsen[2], Rocío A. Pozo[1], O. Sarobidy Rakotonarivo[1], Bram Van Moorter[2], and Nils Bunnefeld[1]

[1] Biological and Environmental Sciences, University of Stirling, Stirling, UK [2] Norwegian Institute for Nature Research, Trondheim, Norway [3] alexander.duthie@stir.ac.uk

Integration and simulation with fisheries

Early development of management strategy evaluation (MSE) models originated in fisheries (Polacheck et al., 1999; Smith et al., 1999; Sainsbury et al., 2000). Consequently, fisheries-focused software for MSE has been extensively developed, including R libraries that focus on the management of species of exceptional interest, such as the Atlantic Bluefin Tuna (*Thunnus thynnus*) (ABFTMSE; Carruthers and Butterworth, 2018b,a), and Indian Ocean Bigeye (*T. obesus*) and Yellowfin (*T. albacares*) Tuna (MSE-IO-BET-YFT; Kolody and Jumppanen, 2016). The largest of all such libraries is the Fisheries Library in R (FLR), which includes an extensive collection of tools targeted for fisheries science. The FLR library has been used in over a hundred publications (recent publications include Jardim et al., 2018; Mackinson et al., 2018; Utizi et al., 2018), and includes an MSE framework for evaluating different harvest control rules.

As part of the ConFooBio project, a central focus of GMSE is on simulating the management of populations of conservation interest, with a particular emphasis on understanding conservation conflict; further development of GMSE is expected to continue with this as a priority, further building upon the decision-making algorithms of managers and users to better understand how conflict arises and can potentially be resolved. Hence, GMSE is not intended as a substitute for packages such as FLR, but the integration of these packages with GMSE could make use of GSME's current and future simulation capabilities, and particularly the genetic algorithm. Such integration might be possible using the `gmse_apply` function, which allows for custom defined submodels to be used within the GMSE framework, and with default GMSE submodels. Hence, GMSE might be especially useful for modelling the management of fisheries under conditions of increasing harvesting demands and stakeholder conflict. We do not attempt such an ambitious project here, but instead show how such a project could be developed through integration of FLR and `gmse_apply`.

Here we follow a Modelling Stock-Recruitment with FLSR example, then integrate this example with `gmse_apply` to explore the behaviour of simulated fishers who are goal-driven to maximise their own harvest. We emphasise that this example is provided only as demonstration of how GMSE can potentially be integrated with already developed fisheries models, and is not intended to make recommendations for management in any population.

Integrating with the Fisheries Library in R (FLR)

The FLR toolset includes a series of pacakges, with several tutorials for using them. For simplicity, we focus here on a model of stock recruitment to be used as the population model in `gmse_apply`. This population model will use sample data and one of the many available stock-recruitment models available in FLR, and a custom function will be written to return a single value for stock recruitment. Currently, `gmse_apply` requires that submodels return subfunction results either as scalar values or data frames that are structured in the same way as GMSE submodels. But interpretation of scalar values is left up to the user (e.g., population model results could be interpreted as abundance or biomass; manager policy could be interpreted as cost of harvesting or as total allowable catch). For simplicity, the observation (i.e., estimation) model will simply be the stock reported from the population model with error, and the manager model will be a total allowable catch calculated from the stock-recruitment relationship that accounts for the number of fishers in the system. The user model, however, will employ the full power of the default GMSE user function to simulate user actions. We first show how a custom function can be made that applies the FLR toolset to a population model.

Modelling stock-recruitment for the population model

Here we closely follow a tutorial from the FLR project. To build the stock-recruitment model, the following packages are needed.

```
install.packages(c("ggplot2"));
install.packages(c("FLCore"), repos="http://flr-project.org/R");
install.packages(c("ggplotFL"), repos="http://flr-project.org/R");
```

To start, we need to read in the `FLCore` and `ggplotFL` libraries.

```
library(FLCore);
```

```
## Loading required package: lattice
```

```
## FLCore (Version 2.6.7, packaged: 2018-04-17 09:12:42 UTC)
```

```
library(ggplotFL);
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:FLCore':
##
##     %+%
```

```
## Warning: replacing previous import 'ggplot2::%+%' by 'FLCore::%+%' when
## loading 'ggplotFL'
```

The data below include sample results for recruitment and spawning-stock biomass (SSB) of North Sea herring.

```
data(nsher);
```

These data are structured using `FLStock`, an S4 class of R objects. The code below creates an `FLSR` object, needed to create a stock-recruitment model.

```
summary(nsher);
```

```
## An object of class "FLSR"
##
## Name:
## Description:
## Quant: age
## Dims:  age   year    unit    season  area    iter
##  1   45  1   1   1   1
##
## Range:  min  minyear max maxyear
##  0   1960    0   2004
##
```

```
## rec          : [ 1 45 1 1 1 1 ], units =  10^3
## ssb          : [ 1 45 1 1 1 1 ], units =  t*10^3
## residuals    : [ 1 45 1 1 1 1 ], units =  NA
## fitted       : [ 1 45 1 1 1 1 ], units =  10^3
##
## Model:   rec ~ a * ssb * exp(-b * ssb)
## Parameters:
##     params
## iter   a       b
##    1 119 0.00945
## Log-likelihood:  15.862(0)
## Variance-covariance:
##              a            b
##   a 255.3388181 1.808870e-02
##   b   0.0180887 1.992659e-06
```

Notice that under `Model`, a Ricker model of stock-recruitment has already been fitted (many stock-recruitment models are available in FLCore; see the FLSR tutorial for more about fitting different models) and parameterised with Maximum Likelihood Estimation (MLE). The code below shows how to apply the Beverton-Holt model model as an alternative, and the `fmle` function parameterises the Beverton-Holt model model using the data in `nsher` and MLE.

```
model(nsher) <- bevholt();
nsher        <- fmle(nsher);
```

```
##   Nelder-Mead direct search function minimizer
## function value for initial parameters = -10.336211
##   Scaled convergence tolerance is 1.54022e-07
## Stepsize computed as 501.110000
## BUILD              3 44.842344 -11.603908

## Warning in log(x@.Data): NaNs produced

## HI-REDUCTION       5 31.685209 -11.603908

## Warning in log(x@.Data): NaNs produced

## HI-REDUCTION       7 17.913114 -11.603908

## Warning in log(x@.Data): NaNs produced

## HI-REDUCTION       9 5.415279 -11.603908

## Warning in log(x@.Data): NaNs produced

## HI-REDUCTION      11 -3.412974 -11.603908
## HI-REDUCTION      13 -8.018030 -11.603908
## LO-REDUCTION      15 -10.336211 -11.603908
## LO-REDUCTION      17 -11.081040 -11.603908
## EXTENSION         19 -11.295930 -12.061705
## LO-REDUCTION      21 -11.603908 -12.061705
## REFLECTION        23 -11.813826 -12.087620
## REFLECTION        25 -12.061705 -12.199591
## LO-REDUCTION      27 -12.087620 -12.199591
## LO-REDUCTION      29 -12.158184 -12.199591
## LO-REDUCTION      31 -12.191726 -12.199591
## HI-REDUCTION      33 -12.192269 -12.199591
## HI-REDUCTION      35 -12.197784 -12.199591
## LO-REDUCTION      37 -12.198015 -12.199591
```

```
## HI-REDUCTION      39 -12.199555 -12.199776
## REFLECTION        41 -12.199591 -12.200058
## HI-REDUCTION      43 -12.199776 -12.200092
## HI-REDUCTION      45 -12.200058 -12.200142
## HI-REDUCTION      47 -12.200092 -12.200155
## HI-REDUCTION      49 -12.200142 -12.200160
## HI-REDUCTION      51 -12.200155 -12.200177
## HI-REDUCTION      53 -12.200160 -12.200177
## LO-REDUCTION      55 -12.200171 -12.200179
## HI-REDUCTION      57 -12.200177 -12.200179
## HI-REDUCTION      59 -12.200178 -12.200179
## HI-REDUCTION      61 -12.200179 -12.200179
## HI-REDUCTION      63 -12.200179 -12.200179
## HI-REDUCTION      65 -12.200179 -12.200179
## Exiting from Nelder Mead minimizer
##     67 function evaluations used
```

```r
summary(nsher);
```

```
## An object of class "FLSR"
##
## Name:
## Description:
## Quant: age
## Dims:  age   year    unit    season  area    iter
##  1   45  1   1   1   1
##
## Range:  min  minyear max maxyear
##  0   1960    0   2004
##
## rec           : [ 1 45 1 1 1 1 ], units =  10^3
## ssb           : [ 1 45 1 1 1 1 ], units =  t*10^3
## residuals     : [ 1 45 1 1 1 1 ], units =
## fitted        : [ 1 45 1 1 1 1 ], units =  10^3
##
## Model:   rec ~ a * ssb/(b + ssb)
## Parameters:
##     params
## iter    a    b
##    1 6736 52.2
## Log-likelihood:  12.2(0)
## Variance-covariance:
##               a          b
##   a 1746206.65 22481.4333
##   b   22481.43   359.7195
```

model(p4sr) <- ricker(); p4sr2 <- fmle(p4sr)

a <- params(p4sr2)[[1]]; b <- params(p4sr2)[[2]];

rec(p4sr2); ssb(p4sr2);

rec_year <- as.numeric(attributes(rec(p4sr2))*dimnames*year); ssb_year <- as.numeric(attributes(ssb(p4sr2))*dimnames*year); recruitment <- NULL; for(year in 1:length(rec_year)){ recruitment[year] <- rec(p4sr)[[year]]; } spawning_stock_biomass <- NULL; for(year in 1:length(ssb_year)){ spawning_stock_biomass[year] <- ssb(p4sr)[[year]]; }

```
rec <- a * spawning_stock_biomass * exp(-b * spawning_stock_biomass);
rec(p4sr)[[51]] <- 4300
```

References

Carruthers, T. and Butterworth, D. (2018a). ABT-MSE : An R package for atlantic bluefin tuna management strategy evaluation. *Collective Volume of Scientific Papers ICCAT*, 74(6):3553–3559.

Carruthers, T. and Butterworth, D. (2018b). Performance of example management procedures for atlantic bluefin tuna. *Collective Volume of Scientific Papers ICCAT*, 73(6):3542–3552.

Jardim, E., Eero, M., Silva, A., Ulrich, C., Pawlowski, L., Riveiro, I., Holmes, S. J., Ibaibarriaga, L., Alzorriz, N., Citores, L., Scott, F., Uriarte, A., Carrera, P., Duhamel, E., and Mosqueira, I. (2018). Testing spatial heterogeneity with stock assessment models. *PLoS One*, 13:e0190891.

Kolody, D. and Jumppanen, P. (2016). IOTC Yellowfin and Bigeye Tuna Management Strategy Evaluation: Phase 1 Technical Support Project Final Report. Technical Report June, CSIRO: Oceans & Atmosphere.

Mackinson, S., Platts, M., Garcia, C., and Lynam, C. (2018). Evaluating the fishery and ecological consequences of the proposed North Sea multi-annual plan. *PLoS One*, 13:e0190015.

Polacheck, T., Klaer, N. L., Millar, C., and Preece, A. L. (1999). An initial evalutaion of management strategies for the southern bluefin tuna fishery. *ICES Journal of Marine Science*, 56(6):811–826.

Sainsbury, K. J., Punt, A. E., and Smith, A. D. (2000). Design of operational management strategies for achieving fishery ecosystem objectives. *ICES Journal of Marine Science*, 57(3):731–741.

Smith, A. D. M., Sainsbury, K. J., and Stevens, R. A. (1999). Implementing effective fisheries-management systems – management strategy evaluation and the Australian partnership approach. *ICES Journal of Marine Science*, 56(6):967–979.

Utizi, K., Notti, E., Sala, A., Buzzi, A., Rodella, I., Simeoni, U., and Corbau, C. (2018). Impact assessment of EMFF measures on Good Environmental Status (GES) as defined by Italy. *Marine Policy*, 88:248–260.