

## Grym: A new open-source implementation of the generalised yield model for flexible stock assessments

D. Maschette<sup>1,2✉</sup>, S. Wotherspoon<sup>1,2</sup>, P. Ziegler<sup>2</sup>, S. Thanassekos<sup>3</sup>  
K. Reid<sup>1,4</sup>, S. Kawaguchi<sup>2</sup>, D. Welsford<sup>5</sup>

<sup>1</sup> Fisheries and Aquaculture Centre  
Institute for Marine and Antarctic Studies  
University of Tasmania  
20 Castray Esplanade Battery Point, 7001, Tasmania  
Australia.

<sup>2</sup> Department of Climate Change, Energy, the Environment and Water  
Australian Antarctic Division  
203 Channel Highway, Kingston, 7050, Tasmania  
Australia.

<sup>3</sup> Commission for the Conservation of Antarctic Marine Living Resources  
Hobart, 7000, Tasmania  
Australia.

<sup>4</sup> Ross Analytics PTY LTD  
1 Lynden Road, Bonnet Hill, Tasmania 7053  
Australia.

<sup>5</sup> Department of Climate Change, Energy, the Environment and Water  
Office of the Science Convener  
GPO Box 3090, Canberra ACT 2601  
Australia.  
Email contact: [dale.maschette@aad.gov.au](mailto:dale.maschette@aad.gov.au)

### Introduction

Constable and de la Mare (1996) presented a generalised yield model (GYM), a stock assessment model, for assessing the status of fish stocks under various levels of uncertainty using either stochastic or deterministic projections. The model was further developed over the subsequent eight years and used as the primary tool for setting catch limits for all target species including krill, toothfish and icefish, and a number of bycatch species in the CAMLR Convention Area (Constable 2004; SC-CAMLR, 2004, paragraph 4.11). In addition to setting catch limits, the GYM has also been used to evaluate management strategies, for example Ziegler et al., (2011) or Welsford (2011). While statistically fitted stock assessment methods have been used for toothfish assessments since 2010, the GYM software is still being used to provide advice on catch limits for some icefish fisheries as well as a number of bycatch species (SC-CAMLR, 2018, paragraph 3.61).

In 2019, the Scientific Committee of CCAMLR noted the need to implement a revised management strategy that would incorporate contemporary information across a range of spatial and temporal scales to improve the likelihood of achieving CCAMLR's conservation objective for the krill fishery. As part of this process, the Scientific Committee also outlined a work plan and schedule for implementing the adopted strategy for krill management (SC-CAMLR-2019, paragraphs 3.16 and 3.34, Tables 1 and 2).

An action item and priority for the Scientific Committee's working groups as part of the krill work plan was the reimplementing of the GYM in an open-source code (SC-CAMLR, 2019, paragraph 13.4, Table 1). This is due to: (i) a lack of consensus on the utility of more current models, (ii) the GYM being the most recent software used to estimate catch limits consistent with the CCAMLR decision rules, and (iii) the FORTRAN

implementation becoming increasingly difficult to maintain on modern computers. A discussion of the implementation of the CCAMLR decision rules is outside the scope of this paper, but for more details on the decision rules see Constable et al. (2000) or the krill fishery management approach (CCAMLR, 2020).

Here, we present the Grym package (Wotherspoon and Maschette, 2020). Designed to work within R (R Core Team, 2020), the Grym was built with the aim to provide a toolbox of functions that replicate the existing core functionality of the GYM software (compiled version 501E; Constable, 2004). By combining different functions, the effects of different decisions, for example varying recruitment relationships, can be tested. The Grym can be used as the underlying package for a range of stock assessment and management strategy evaluation applications to facilitate assessments run by either non-specialist programmers or assessment scientists.

### Implementation

The GYM software (Constable and de la Mare, 1996; Constable, 2004) is a single monolithic executable that performs either stochastic or deterministic stock projections under a variety of assumed recruitment, mortality and harvesting. The precise form of the assumed models of recruitment, mortality and harvesting used in the projection is specified through a complex set of configuration files read at the beginning of each analysis.

Where the FORTRAN code of the latest GYM executable (compiled version 501E; Constable 2004) differed from the intent of the GYM described by Constable and de la Mare (1996), the Grym functionality was written to be consistent with the intent described by Constable and de la Mare (1996). This approach has led to some differences in model outputs between the GYM executable and the equivalent functions of the Grym package, which are discussed in more detail below.

The Grym performs stochastic stock projection under a variety of assumed recruitment, mortality and harvesting. However, the Grym is an R package that provides separate functions for each of the major computational tasks required for the projections. Similar to the GYM, which used checkboxes,

here, the user provides the ‘control logic’ for the Grym, which calls these functions to perform an assessment in the desired configuration.

The core functionality of the GYM is to project stock abundance, biomass and yield in each age class forward in time over a single year. The GYM projects forward over time by integrating a system of differential equations with an adaptive Runge–Kutta scheme. The GYM assumes the number  $N_{a,y}(t)$ , biomass  $B_{a,y}(t)$  and yield  $Y_{a,y}(t)$  of individuals of age  $a$  in year  $y$  at time  $t$  within the year satisfy the system of differential equations (Constable and de la Mare, 1996)

$$\begin{aligned} \frac{dN_{a,y}}{dt} &= -[M_y m_{a,y}(t) + F_y f_{a,y}(t)] N_{a,y} \\ \frac{dB_{a,y}}{dT} &= w(a+t) \frac{dN_{a,y}}{dt} + N_{a,y}(t) \frac{dw(a+t)}{dt} \\ \frac{dY_{a,y}}{dt} &= F_y f_{a,y}(t) B_{a,y}. \end{aligned} \quad (1)$$

Here  $w(a+t)$  represents the weight at exact age  $(a+t)$  and the total natural mortality  $M_y m_{a,y}(t)$  has been decomposed into an annual component  $M_y$  that is constant in the year and a component  $m_{a,y}(t)$  that varies within the year, and similarly the total fishing mortality  $F_y f_{a,y}(t)$  has been decomposed into an annual component  $F_y$  that is constant in the year and a component  $f(a,t,y)$  that varies within the year. Both  $m_{a,y}(t)$  and  $f_{a,y}(t)$  may also vary from year to year, and so an appropriate normalisation must be chosen for these components to ensure identifiability. The total yield in each year is the sum of the contributions from each age class at the end of the year ( $t=1$ )

$$Y_y = \sum_a Y_{a,y}(1). \quad (2)$$

The GYM solves this system of differential equations for each year by assuming the time varying functions  $m_{a,y}(t)$ ,  $f_{a,y}(t)$  and  $w(a+t)$  are piecewise linear and integrating the system with an adaptive Runge–Kutta scheme.

The Grym takes a more direct approach. The system of governing equations has a solution:

$$\begin{aligned}
N_{a,y}(t) &= N_{a,y}(0) \exp \\
&\quad \left[ -M_y \int_y^{y+t} m_{a,y}(t) d\tau - F_y \int_y^{y+t} f_{a,y}(t) d\tau \right] \\
B_{a,y}(t) &= w(a+t) N_{a,y}(t) \\
Y_{a,y}(t) &= \int_y^{y+t} F_y f_{a,y}(t) B_{a,y}(\tau) d\tau.
\end{aligned} \tag{3}$$

The Grym computes this directly by evaluating the required integrals with the composite trapezoidal rule. As  $m_{a,y}(t)$  and  $f_{a,y}(t)$  are assumed piecewise linear, this yields exact solutions for  $N_{a,y}$  and  $B_{a,y}$ .

Many quantities of interest, such as the spawning, exploitable or vulnerable biomass are computed as temporal averages over a specified reference interval. The GYM calculates temporal averages as simple sums, but for internal consistency with the above, the Grym calculates temporal averages with the composite trapezoidal rule.

To ensure that the annual scalings  $M_y$  and  $F_y$  are identifiable, some appropriate normalisation must be imposed upon the intra-annual patterns of mortality  $m_{a,y}(t)$  and  $f_{a,y}(t)$ . In the GYM, identifiability is enforced by requiring the intra-annual mortalities integrate to unity over a year; but this choice implies the same total fishing effort is expended in each year and complicates the interpretation of  $F_y$  when the length of the fishing season changes dramatically over time. For this reason, in the Grym the choice of normalisation is left to the discretion of the user.

Aside from mortality and harvesting, the key determinant of stock survival is recruitment. The GYM offers a number of options for modelling recruitment, including (i) drawing from a lognormal distribution with prescribed mean and coefficient of variation; (ii) bootstrapping from a time series of recruitment estimates derived from survey data; and (iii) the proportional recruitment model described in de la Mare (1994), and these are reproduced in the Grym.

The proportional recruitment model defines  $R$  the ratio of the number of individuals of a given reference age to all individuals of that age or older and assumes estimates of  $R$  are available from multiple surveys. The method derives estimates of natural mortality and the mean and variance of the recruitment required to reproduce the mean and

variance of  $R$  observed in the surveys. The method described by de la Mare (1994) and several extensions described in Pavez et al (2023) are implemented in the Grym.

However, de la Mare (1994) advocates resampling the mean and variance of  $R$  observed in the surveys from normal and chi-squared distributions respectively. But as  $R$  is a proportion it has bounded variation, and the sample variance of  $R$  will not follow a chi-square distribution. Resampling the variance of  $R$  from a chi-square distribution will produce variances that are unattainable, and it is likely this is the cause of the ‘premature termination’ issue reported by Kinzey, Watters and Reiss (2013). Instead, the Grym offers a parametric bootstrap alternative to simulate new values for the mean and variance of  $R$  that are consistent with the observed values. If the observed mean and variance were estimated from  $n$  independent surveys, random recruits based on the observed mean and variance are used to generate  $n$  new age structures from which  $R$ , and hence the mean and variance of  $R$ , are estimated. This procedure is guaranteed to produce realistic variances and is described in more detail in Pavez et al. (2023).

The source code for the FORTRAN compiled version 501E was available for reference in order to compare the implementation and outputs with the Grym. In addition to the standard documentation, the Grym package provides a ‘vignette’ which shows how each function is used with their relevant input parameters. The accompanying package GrymExamples contains examples for icefish, krill and toothfish assessments which are used here to compare projections from both the GYM software and the Grym package.

### Comparisons of implementations

In order to ensure consistency between implementations, the GYM software (version 501E) and the Grym package (version 1.0.0) were compared for three different scenarios:

(i) 2019 mackerel icefish (*Champsocephalus gunnari*) assessment for Division 58.5.2 (Maschette et al., 2019) representing an assessment which uses constant fishing mortality ( $F$ ) over a two-year deterministic projection.

(ii) 1996 krill assessment (Constable and de la Mare, 1996), representing an assessment which allows for a 20-year stochastic projection with constant gamma removals.

(iii) 2006 Patagonian toothfish (*Dissostichus eleginoides*) assessment for Division 58.5.2 (Welsford et al., 2006), representing an assessment which does constant catch removals over a 35-year stochastic projection.

These scenarios cover the typical range of stock assessments conducted with the GYM software within CCAMLR with each scenario testing different configurations of the Grym package functionality.

It is worth noting that some estimates by Constable and de la Mare (1996) differ from those estimates using the GYM version 501E here due to internal changes in the GYM between 1996 and 2004 (Constable, 2004). Comparisons were made using the parameters outlined in Table 2 with the Grym implementation code in Appendices 1 to 3.

The comparisons between the currently compiled GYM software and the Grym for these three types of assessments show that in most instances the Grym returns very similar results to the GYM software (Tables 3 to 6, Figures 1 and 2). The small overall differences between the estimates of the two models is primarily due to differences in the sequences of random deviates drawn in individual projection runs. Although Grym and the GYM use different integration schemes to compute projections, as noted previously the technique based on the composite trapezoidal rule used in Grym will be more precise than the Runge–Kutta scheme of the GYM.

## Conclusions

The new open-source implementation of GYM in the R package Grym produces very consistent outputs compared to the existing GYM software for three types of fisheries assessment typically used in CCAMLR. The differences in these assessments highlight the flexibility of the Grym package.

The Grym package reproduces the core functionality of the GYM software to conduct fisheries stock population projections, whilst also addressing some of the limitations in the GYM software,

which caused software failure in some models with large recruitment variability. The Grym package also improves on the GYM software by providing greater precision by directly integrating the governing equations by means of the composite trapezoidal rule.

The open source and interactive nature of R and the Grym package mean that the source code and any resulting outputs can be readily evaluated and assessed. The modular nature of the package provides increased flexibility, as the user can easily explore the effects of pairing alternative models of recruitment or mortality with existing GYM software functionality. The GrymExamples package contains a number of examples for what these may look like under various scenarios.

This modular nature also allows users to develop and test their own additional requirements for assessments as part of future developments, for example the potential for cyclical recruitment time series (Thanassekos, 2020) or incorporating multiple fleets into models.

Given the increased transparency and flexibility of the Grym package, and the demonstrated high degree of consistency in the outputs of the GYM software and Grym package; we recommend that the Grym package be used in future stock assessments which would have otherwise used the GYM software. This would include the assessment in the krill management work plan as set out by the Scientific Committee of CCAMLR (SC-CAMLR, 2019, paragraphs 3.16 to 3.18, Table 1).

## Acknowledgements

The authors would like to acknowledge that this work was conducted on the traditional lands of the muwinina people. Respecting the practice of the palawa kani (the aboriginal people of lutrawita/Tasmania) proper nouns in their languages are not capitalised. We would also like to thank Andrew Constable and Tim Lamb for comments at the initial stages of this work.

## References

- CCAMLR. 2019. *Report of the Thirty-Eighth Meeting of the Commission (CCAMLR-38)*. CCAMLR, Hobart, Australia: 196 pp.

- CCAMLR. 2021. CCAMLR's approach to managing the krill fishery (2020). *Euphausia superba* in Area 48. CCAMLR, Hobart, Australia.
- Constable, A.J. 2004. Modifications to the generalised yield model in 2004, version GYM501E. EXE. Document *WG-FSA 04/91*. CCAMLR, Hobart, Australia: 3 pp.
- Constable, A.J. and W.K. de la Mare. 1996. A generalised model for evaluating yield and the long-term status of fish stocks under conditions of uncertainty. *CCAMLR Science*, 3: 31–54.
- Constable, A.J., W.K. de la Mare, D.J. Agnew, I. Everson and D. Miller. 2000. Managing fisheries to conserve the Antarctic marine ecosystem: practical implementation of the Convention on the Conservation of Antarctic Marine Living Resources (CCAMLR). *ICES J. Mar. Sci.*, 57 (3): 778–791, doi: 10.1006/jmsc.2000.0725.
- de la Mare, W.K. 1994. Modelling Krill Recruitment. *CCAMLR Science*, 1: 49–54.
- Kinzey, D., G. Watters and C.S. Reiss. 2013. Effects of recruitment variability and natural mortality on generalised yield model projections and the CCAMLR decision rules for Antarctic krill. *CCAMLR Science*, 20: 81–96.
- Maschette, D., G. Nowara and D. Welsford. 2019. A preliminary assessment of mackerel icefish (*Champsocephalus Gunnari*) in Division 58.5.2, based on results from the 2019 random stratified trawl survey. Document *WG-FSA-2019/02*. CCAMLR, Hobart, Australia.
- Pavez, C., S. Wotherspoon, D. Maschette, K. Reid and K. Swadling. 2023. Recruitment modelling for *Euphausia superba* stock assessments considering the recurrence of years with low recruitment. *CCAMLR Science*, 24.
- R Core Team. 2020. R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria.
- SC-CAMLR. 2004. Report of the Working Group on Fish Stock Assessment. In: *Report of the Twenty-Third Meeting of the Scientific Committee (SC-CAMLR-XXIII)*, Annex 5. CCAMLR, Hobart, Australia: 339–658.
- SC-CAMLR. 2018. Fishery Report: *Champsocephalus Gunnari* Heard Island (Division 58.5.2). In: *Report of the Thirty-Seventh Meeting of the Scientific Committee (SC-CAMLR-XXXVII)*, Annex 8, Appendix T. CCAMLR, Hobart, Australia: 16 pp.
- SC-CAMLR. 2019. *Report of the Thirty-eighth Meeting of the Scientific Committee (SC-CAMLR-38)*. CCAMLR, Hobart, Australia: 433 pp.
- Thanassekos, S. 2020. RecMaker - Krill proportional recruitment time series simulator. [github.com/ccamlr/RecMaker](https://github.com/ccamlr/RecMaker).
- Welsford, D.C., A.J. Constable, T. Lamb and T. Robertson. 2006. Preliminary assessment of Patagonian toothfish, *Dissostichus eleginoides*, in the vicinity of Heard Island and McDonald Islands (Division 58.5.2), based on a survey in May–June 2006 using the generalised yield model. Document *WG-FSA-06/45 Rev. 1*. CCAMLR, Hobart, Australia.
- Welsford, D.C. 2011. Evaluating the impact of multi-year research catch limits on overfished toothfish populations. *CCAMLR Science*, 18: 47–55.
- Wotherspoon, S. and D. Maschette. 2020. Grym: R Implementation of the generalized yield model. R package version 0.1.0.9000. [github.com/AustralianAntarcticDivision/Grym](https://github.com/AustralianAntarcticDivision/Grym).
- Ziegler, P.E., D.C. Welsford and A.J. Constable. 2011. Length-based assessments revisited – why stock status and fishing mortality of long-lived species such as toothfish cannot be inferred from length-frequency data alone. *CCAMLR Science*, 17: 57–73.

## Code appendices

Please note these appendices provide code that produced the results within the attached paper. Whilst the code was fully functional at the time of publication, changes between versions of R and the packages this code rely on may mean this is not always the case.

The three appendices here are also recreated as vignettes within the Grym package. As such, the package should always have working version of these examples that have been updated to take account of changes in code dependencies. The package can be found here: <https://github.com/AustralianAntarcticDivision/GrymExamples>

APPENDIX 1

### *Champscephalus gunnari* Grym assessment code

#### Icefish Assessment Comparisons

The icefish assessment aims to determine the fishing mortality and hence a total allowable catch that yields a prescribed two-year escapement relative to an unfished population (see [https://fishdocs.ccamlr.org/SAannex\\_HIMI\\_ANI\\_2021.pdf](https://fishdocs.ccamlr.org/SAannex_HIMI_ANI_2021.pdf) for more details).

#### *Library*(Grym)

#### Projection

The current strategy is to project forward two years for a range of potential fishing mortalities, and then subsequently determine the mortality that yields the target relative escapement by inverse interpolation.

The function *icefishPr* projects forward two years for a specified range of fishing mortalities. A projection is performed for the current year and scaled to the survey data to determine the initial abundance for the following year. If the catch limit for the year has been reached, the projection is done assuming zero fishing mortality (as no further fishing can occur) but if some catch allocation remains after the survey period, it is assumed the remaining allocation will be caught between the end of the survey and the start of the following year. The initial abundances are then projected forward using the fish demographic parameters and fishery parameters for two years for a range of fishing mortalities, and annual summaries collated.

The model assumes a von Bertalanffy relationship between length and age, and a power law relationship between weight and length. The user must also supply values used to compute fishing selectivity as a function of age.

The arguments are

*M* – Natural mortality estimate.

*F* - Fishing mortalities to test

*Catch* - Remaining catch allocation after survey in survey year

*surveyN* - Relative numbers (per km<sup>2</sup>) in each cohort from survey

*surveyB* - Biomass estimate from survey

*surveyI* - Increments (ie days of season) over which survey is taken

*spawnI* - Increments (ie days of season) over which spawning numbers and biomass are estimated

*VB.t0*, *VB.K*, *VB.Linf* - Parameters for von Bertalanffy length at age relationship

*WLa*, *WLb* - Parameters for allometric weight at length relationship

*age.selectivity* - Age selectivity function.

*Fmax* - Maximum allowable fishing mortality

```
icefishPr <- function(M, F, Catch=0, surveyN, surveyB, surveyI, spawnI,
  VB.t0=0.06671238, VB.K=0.36842178, VB.
Linf=489.73706791,
  WLa=9.157E-10, WLb=3.316,
  age.selectivity=approxfun(c(0, 2.5, 3), c(0, 0, 1), ru
le=2),
  Fmax=2.5) {

  ## Ensure 0 included in test fishing mortalities
  F <- sort(union(0, F))

  ## Two year projections of 10 age classes with a daily time step
  n.yr <- 2
  n.inc <- 365
  Ages <- 1:10
  Days <- seq(0, 1, length=n.inc+1)

  ## Matrices of ages, lengths and weights for each day and age class
  as <- outer(Days, Ages, FUN="+")
  Ls <- vonBertalanffyAL(as, t0=VB.t0, K=VB.K, Linf=VB.Linf)
  ws <- powerLW(Ls, a=WLa, b=WLb)

  ## Constant intra-annual natural mortality
  ms <- matrix(1, n.inc+1, length(Ages))
  Ms <- ctrapz(ms, 1/n.inc)
  MMs <- M*Ms

  ## Within year fishing mortality is determined by an age based selec-
  tivity
  fs <- array(age.selectivity(as), dim(as))
  Fs <- ctrapz(fs, 1/n.inc)

  ### Projection to end of year from survey data
  if(Catch>0) {
    ## Adjust within-year fishing mortality for post-survey Catch
    fs0 <- rep.int(c(0, 1), c(max(surveyI), n.inc+1-max(surveyI)))
    fs0 <- fs0/trapz(fs0, 1/n.inc)*fs
    Fs0 <- ctrapz(fs0, 1/n.inc)
    pr0 <- projectC(ws, MMs, Fs0, fs0, Catch, surveyN, surveyI, surveyB, surve
```

```

yI, yield=1, Fmax=Fmax)
  if(pr0$F==Fmax) warning("Target catch could not be recovered")
} else {
  pr0 <- project(ws, MMs, 0, 0, surveyN, surveyI, surveyB, surveyI, yield=0)
  pr0$F <- 0
}
SSB0 <- meanStock(pr0$B, 1, spawnI)

## Annual cohort totals
d <- data.frame(Year=c(rep(0:n.yr, length(F))), F=0, Nf=0, Bf=0, Y=0, SSN=
0, SSB=0, Escapement=0)
k <- 0

## Project forward for prescribed fishing mortalities.
for(Fk in F) {
  ## Reset to survey year
  pr <- pr0
  d[k <- k+1, ] <- data.frame(Year=0, F=Fk, Nf=sum(final(pr$N)), Bf=sum(fi
nal(pr$B)), Y=sum(pr$Y),
  SSN=meanStock(pr$N, 1, spawnI), SSB=SSB0, Escapement=1)
  for(yr in seq_len(n.yr)) {
    ## Project
    N0 <- advance(pr$N)
    pr <- project(ws, MMs, Fk*Fs, Fk*fs, N0, yield=1)
    SSB <- meanStock(pr$B, 1, spawnI)
    d[k <- k+1, ] <- data.frame(Year=yr, F=Fk, Nf=sum(final(pr$N)), Bf=su
m(final(pr$B)), Y=sum(pr$Y),
    SSN=meanStock(pr$N, 1, spawnI), SSB=SSB, Escapement=SSB/SSB0)
  }
}
d
}
}

```

2019 assessment

Define the reference date that sets the start of the season

```

SeasonDate <- as.Date("2018-12-01")
SurveyDate <- as.Date("2019-04-04")
surveyI <- as.numeric(SurveyDate-SeasonDate)+c(0, 1)
surveyN <- c(127.106, 617.426, 1988.91, rep(0, 7))
surveyB <- 3723.761
SpawnDate <- as.Date("2019-11-30")
spawnI <- as.numeric(SpawnDate-SeasonDate) + c(0, 1)

d <- icefishPr(M=0.4, F=seq(0.11, 0.20, 0.001), Catch=0, surveyN, surveyB, sur
veyI, spawnI, VB.t0=0.06671238, VB.K=0.36842178, VB.Linf=489.73706791,
  WLa=1.078e-09, Wlb=3.286)

```



```

d2 <- d[d$Year==2,]
d2$RelEscape <- d2$Escapement/d2$Escapement[1]
F75 <- approx(d2$RelEscape[-1],d2$F[-1],0.75)$y

d2019 <- icefishPr(M=0.4,F=c(F75),Catch=0,surveyN,surveyB,surveyI,spawnI,
                  VB.t0=0.06671238,VB.
K=0.36842178,VB.Linf=489.73706791,
                  WLa=1.078e-09,WLb=3.286)
d2019$Scenario<-rep(c("Grym no fishing","Grym 75 escapement"), each=3)
d2019

```

APPENDIX 2

### *Euphausia superba* Grym assessment code

This document aims to reproduce the example presented in Constable and de la Mare (1996).

```

Library(Grym)
Library(ggplot2)
Library(dplyr)
Library(tidyr)
Library(furrr)
set.seed(31)

```

Model

The *EsuperbaProjection* function returns a function that generates a single set of random projections that differ only in gamma. The projections within each set use a common random selectivity and maturity curves, virgin biomass and recruitment series, so within a set the projections only differ by the level of fishing applied. The resulting function takes a single numerical argument that is used to identify the set.

```

EsuperbaProjection <- function(gamma=c(0,0.1,0.136,0.15,0.2),n.
years=20) {

  ## Daily time steps with 8 age classes
  nsteps <- 365
  Ages <- 0:7
  Days <- seq(0,1,length=nsteps+1)
  h <- 1/nsteps

  ## Spawning and monitoring interval
  spawnI <- 121:213
  monitorI <- 92

  ## Ages, length at age and weight at age
  ages <- outer(Days,Ages,FUN="+")
  Ls <- vonBertalanffyRAL(ages,t0=0.0,K=0.45,Linf=60,f0=0,f1=93/365)
  ws <- powerLW(Ls,1,3)

```

```

## Constant intra-annual natural mortality
ms <- matrix(1,nsteps+1,length(Ages))
Ms <- ctrapz(ms,h)
Msf <- final(Ms)

## Within year fishing pattern - season is first 90 days
fwy <- double(nsteps+1)
fwy[31:120] <- 1
fwy <- fwy/trapz(fwy,h)

B0Logsd <- sqrt(Log(1+0.3^2))

## This function performs a projection for each prescribed gamma.
function(run) {

  ## Length based maturity and selectivity - ramp width is constant
  ## but the midpoint is selected uniformly from a range.
  gs <- rampOgive(Ls,runif(1,34,40),12)
  ss <- rampOgive(Ls,runif(1,38,42),10)

  ## Construct fishing mortalities from season and selectivity
  fs <- fwy*ss
  Fs <- ctrapz(fs,h)
  Fsf <- final(Fs)

  ## Uniform natural mortalities
  M <- runif(1,0.4,1.0)
  MMs <- M*Ms

  ## Uniform (log) recruitment std dev.
  sigmaR <- sqrt(log(1+runif(1,0.4,0.6)^2))

  ## Median spawning biomass estimated from 1001 samples
  R <- matrix(rlnorm(1001*length(Msf),-sigmaR^2/2,sigmaR),1001,length
h(Msf))
  ssb0 <- spawningB0S(R,gs,ws,Ms,M,spawn=spawnI)$median

  ## Stochastic initial age structure in the absence of fishing
  N0 <- ageStructureS(rlnorm(length(Msf),-sigmaR^2/2,sigmaR),Msf,M)

  ## Recruitment series
  Rs <- rlnorm(n.years,-sigmaR^2/2,sigmaR)

  ## Matrix of annual summaries
  n <- (1+n.years)*length(gamma)
  df <- matrix(0,n,11+ncol(ages))
  colnames(df) <- c("Year","Gamma","R","N","B","B0","SSN","SSB","SSB
0","Catch","F",paste0("Na.",1:ncol(ages)))

```

```

## Initial projection assuming no fishing
pr0 <- project(ws,MMs,Nref=N0,yield=0)
pr0$F <- pr0$Y <- 0

## Initial biomass in monitoring period + log Normal error
b0 <- meanStock(pr0$B,period=monitorI)
b0 <- rlnorm(1,Log(b0)-B0Logsd^2/2,B0Logsd)

k <- 0
## Project for each gamma ratio
for(g in gamma) {
  ## Target catch
  catch <- g*b0

  ## Reset to virgin state
  pr <- pr0
  ssb <- spawningStock(pr$B,gs,spawnI)

  for(yr in 0:n.years) {

    if(yr > 0) {
      ## Recruitment depletion
      r <- min(1,ssb/(0.2*ssb0))
      ## Project over year
      N0 <- advance(pr$N,r*Rs[yr])
      pr <- projectC(ws,MMs,Fs,fs,catch,Nref=N0,yield=1,Fmax=1.5)
      #if(pr$F==1.5) return(NULL)
    }
    ssb <- spawningStock(pr$B,gs,spawnI)

    ## Collate annual summaries
    df[k<-k+1,] <- c(yr,g,initial(pr$N)[1],sum(initial(pr$N)),sum(
initial(pr$B)),b0,
                                spawningStock(pr$N,gs,spawnI),ssb,ssb0,sum(pr
$Y),pr$F,initial(pr$N))
  }
}
data.frame(Run=run,M=M,df)
}
}

```

Projection

Calling *EsuperbaProjection* returns a function that generates a single ‘un’ of the simulation

```
sim <- EsuperbaProjection()
```

The function takes a single argument that is used to label the results. Within a single call to `sim`, a common set of random maturity, selectivity, initial ages and recruitment series are used to make projections for each requested gamma.

```
df <- sim(1)
head(df)
tail(df)
```

For these parameters, the recruitment variability has greater impact than increased fishing pressure

```
library(ggplot2)
library(dplyr)
ggplot(df %>% mutate(Gamma=factor(Gamma)), aes(x=Year, y=N, colour=Gamma))
+geom_line()
```

Generate 10001 runs and bind them into one large dataframe. The runs are performed in parallel with the `furrr` library

```
plan(multiprocess)
#system.time(df <- future_map_dfr(1:10001, sim))
df <- future_map_dfr(1:10001, sim, .progress = TRUE)
saveRDS(df, "Esup96_sim.rds")

dat<-read.csv("./esup96/esup96.PG") #load GYM output
names(dat)

df %>% group_by(Gamma, Run) %>% summarize(Dep=min(SSB/SSB0)) %>%
summarize(Pr=mean(Dep < 0.2)) #Grym depletion

dat %>% group_by(Test, Trial) %>% summarize(Dep=min(SSB.Status)) %>%
summarize(Pr=mean(Dep < 0.2)) #GYM depletion

#GYM Escapement
gymesc<- dat %>% group_by(Test) %>% filter(Year %in% max(Year)) %>%
summarise(Med=median(SSB.Status))

gymtar<- gymesc %>% filter(Test==0) %>% mutate(Target=Med*0.75)

gymesc$Target<-gymtar$Target

#Grym Escapement
grymesc<- df %>% group_by(Gamma) %>% filter(Year %in% max(Year)) %>%
summarise(Med=median(SSB/SSB0))

grymtar<- grymesc %>% filter(Gamma==0) %>% mutate(Target=Med*0.75)

grymesc$Target<-grymtar$Target
```

*grymesc*  
*gymesc*

APPENDIX 3

### *Dissostichus eleginoides* Grym assessment code

This document aims to reproduce the Patagonian toothfish assessment presented in Welsford et al. (2006).

```
Library(Grym)
Library(ggplot2)
Library(dplyr)
Library(furrr)
set.seed(31)
```

Historic Data

Survey Data

Recruitment is backprojected from survey data. In this case, natural mortality is assumed constant, so it suffices to backproject the recruitment once and use the same recruitment series in all projections.

Import the survey data

```
survey.df <- read.csv(textConnection("
Survey,Year,Frac,Age,Density,SE,Area,ObsTotal,ExpTotal
1,1989,0.49,3,0.01,0.01,53383.16,70.32,74.57
1,1989,0.49,4,30.56,8.96,53383.16,70.32,74.57
1,1989,0.49,5,6.83,7.13,53383.16,70.32,74.57
1,1989,0.49,6,0.01,0.01,53383.16,70.32,74.57
1,1989,0.49,7,0.01,0.01,53383.16,70.32,74.57
2,1992,0.77,3,8.01,8.97,53383.16,67.54,85.22
2,1992,0.77,4,27.06,12.9,53383.16,67.54,85.22
2,1992,0.77,5,0.01,0.01,53383.16,67.54,85.22
2,1992,0.77,6,16.8,19.26,53383.16,67.54,85.22
2,1992,0.77,7,5.66,21.84,53383.16,67.54,85.22
3,1998,0.33,3,25.85,7.63,80660.77,373.59,371.54
3,1998,0.33,4,0.01,0.01,80660.77,373.59,371.54
3,1998,0.33,5,85.13,65.51,80660.77,373.59,371.54
3,1998,0.33,6,174.83,104.99,80660.77,373.59,371.54
3,1998,0.33,7,0.01,0.01,80660.77,373.59,371.54
3,1998,0.33,8,66.34,31.68,80660.77,373.59,371.54
4,2000,0.48,3,27.32,8.31,85693.96,198.46,200.63
4,2000,0.48,4,5.8,15.56,85693.96,198.46,200.63
4,2000,0.48,5,59.59,35.74,85693.96,198.46,200.63
4,2000,0.48,6,32.98,47.78,85693.96,198.46,200.63
4,2000,0.48,7,29.64,30.16,85693.96,198.46,200.63
5,2001,0.48,3,14.4,9.37,85693.96,207.12,206.07
5,2001,0.48,4,47.26,17.19,85693.96,207.12,206.07
```

```

5,2001,0.48,5,0.01,0.01,85693.96,207.12,206.07
5,2001,0.48,6,101.72,42.56,85693.96,207.12,206.07
5,2001,0.48,7,9.3,37.05,85693.96,207.12,206.07
6,2002,0.42,3,24.57,10.36,42063.96,142.77,140.1
6,2002,0.42,4,28.16,23.4,42063.96,142.77,140.1
6,2002,0.42,5,18.55,30.15,42063.96,142.77,140.1
6,2002,0.42,6,56.89,21.35,42063.96,142.77,140.1
7,2003,0.43,3,0.01,0.01,85123.46,234.65,231.64
7,2003,0.43,4,102.51,28.86,85123.46,234.65,231.64
7,2003,0.43,5,24.19,66,85123.46,234.65,231.64
7,2003,0.43,6,54.69,74.47,85123.46,234.65,231.64
8,2004,0.43,3,0.01,0.01,85693.96,240.42,241.79
8,2004,0.43,4,0.01,0.01,85693.96,240.42,241.79
8,2004,0.43,5,168.88,29.37,85693.96,240.42,241.79
8,2004,0.43,6,20.36,29.24,85693.96,240.42,241.79
9,2005,0.47,3,0.01,0.01,85693.96,173.09,175.94
9,2005,0.47,4,52.75,11.17,85693.96,173.09,175.94
9,2005,0.47,5,0.01,0.01,85693.96,173.09,175.94
9,2005,0.47,6,99.76,18.49,85693.96,173.09,175.94”),header=T)

```

The *SurveyAdjust* function scales the survey densities to abundance, computes the adjustment scale abundance to recruitment-based survival and forms the weighted geometric means of the estimates of recruitment.

```

SurveyAdjust <- function(survey.df,Ms,M,rec.age) {
  ## Scale density to abundance
  r <- survey.df$Area*survey.df$ObsTotal/survey.df$ExpTotal
  ab.mn <- r*survey.df$Density
  ab.se <- r*survey.df$SE

  ## Compute log survival adjustment
  inc <- ceiling((nrow(Ms)-1)*survey.df$Frac)+1
  S <- surveySurvival(survey.df$Year,survey.
df$Age,inc,inc,Ms,M,rCls=rec.age)

  ## Weight rescaled abundance by 1/cv^2
  ab.wt <- (ab.mn/ab.se)^2
  ## Compute the year of “recruitment” to the target age class
  rc.yr <- survey.df$Year-survey.df$Age+rec.age
  yr <- seq.int(min(rc.yr),max(rc.yr))
  rec.yf <- factor(rc.yr,yr)
  ## Compute the weighted geometric means
  data.frame(
    Year=yr,
    Rec=exp(tapply(ab.wt*(Log(ab.mn)-Log(S)),rec.yf,sum)/tapply(ab.
wt,rec.yf,sum)))
}

```

Form natural mortalities and compute recruitment estimates.

```

## Constant intra-annual natural mortality
nsteps <- 24
Ages <- 4:35
Days <- seq(0,1,length=nsteps+1)
h <- 1/nsteps
ms <- matrix(1,nsteps+1,length(Ages))
Ms <- ctrapz(ms,h)
M <- 0.13
recruit.df <- SurveyAdjust(survey.df,Ms,M,4)
recruit.df

```

### Catch

Import the corresponding catch data

```

catch.df <- read.csv(textConnection(
Year,Catch
1986,0
1987,0
1988,0
1989,0
1990,0
1991,0
1992,0
1993,0
1994,0
1995,3000000
1996,9044000
1997,7915000
1998,3974000
1999,4720000
2000,4984000
2001,6245000
2002,4356000
2003,3501000
2004,3048000
2005,2696000"),header=T)

```

### Growth

Growth patterns are inferred from length-at-age data.

Import the length-at age-data

```

Length.df <- read.csv(textConnection(
Age, Length
0, 197.56
1, 251.01
2, 307.54
3, 367.28

```

```

4, 430.40
5, 497.03
6, 547.46
7, 594.75
8, 641.07
9, 686.46
10, 730.91
11, 774.47
12, 817.13
13, 858.93
14, 899.88
15, 940.00
16, 979.29
17, 1017.79
18, 1055.51
19, 1092.46
20, 1128.65
21, 1164.11
22, 1198.85
23, 1232.88
24, 1266.22
25, 1298.88
26, 1330.87
27, 1362.22
28, 1392.92
29, 1423.00
30, 1452.47
31, 1481.34
32, 1509.62
33, 1537.33
34, 1564.47
35, 1591.06"),header=T)
Length.age <- approxfun(Length.df$Age, Length.df$Length,rule=2)
plot(Length.age,0,55)

```

### Interpolation

Create an age or length-based array through interpolation

```

approxArray <- function(x,y,arr,rule=2) array(approx(x,y,arr,rule=rule)
)$y,dim(arr))

```

### Selectivity

Selectivity varies with year and may be length or age based. This function creates a list of selectivity matrices together with an index vector that matches year to the appropriate selectivity.

```

mkSelectivity <- function(ages,ls) {
  select5pt <- function(x) approxArray(x,c(0,0,1,1,0),ages)
}

```



```

## Age based selectivity
ss <- list()
## 1986-1994 - age based selectivity
ss[[1]] <- approxArray(x
=c(0.0,4.1,4.9,5.8,7.0,8.4,9.8,13.7,14.9,16.1,17.3,18.4),
                        y=c(0.0,0.0,0.14,0.5,0.8,0.9,1.0,1.0,0.9,0.85
,0.4,0.3),
                        ages)
## 1995 - length based selectivity
ss[[2]] <- rampOgive(Ls,670,250)
## 1996 - age based selectivity
ss[[3]] <- select5pt(c(0.0,5.8,7.0,8.2,8.4))
## 1997 - age based selectivity
ss[[4]] <- select5pt(c(0.0,4.9,5.8,11.1,13.7))
## 1998 - age based selectivity
ss[[5]] <- select5pt(c(0.0,5.3,5.8,14.9,17.3))
## 1999-2004 - age based selectivity
ss[[6]] <- select5pt(c(0.0,4.1,8.4,16.1,17.3))

## Year 1 = 1986
list(index=setNames(c(rep(1,9),2:5,rep(6,6),1),1986:2005),ss=ss)
}

```

## Model

The *ToothfishProjection* function returns a function that generates a single set of random projections that differ only in target catch. The projections within each set use a common random selectivity and maturity curves, virgin biomass and recruitment series, so within a set the projections only differ by the level of fishing applied. The resulting function takes a single numeric argument that is used to identify the set.

The arguments are

*Catches* - the catch targets to test.

*Catch.df* - dataframe of historic catches - this must contain the catches for each year from the earliest modelled year up to the first projected year, ordered by year.

*Recruit.df* - dataframe of recruitment estimates from survey data.

*Length.df* - dataframe of length-at-age data.

*n.years* - the number of years to project ahead.

*Year1* - the earliest modelled year.

```

ToothfishProjection <- function(Catches,catch.df,recruit.df,length.
df,n.years=35,Year1=min(catch.df$Year)) {

```

```

## Daily time steps with 8 age classes
nsteps <- 24
Ages <- 4:35
plus <- 55-35
Days <- seq(0,1,length=nsteps+1)
h <- 1/nsteps

## Spawning and monitoring interval
spawnI <- 14:15
monitorI <- 1:25

## Ages, length at age and weight at age
ages <- outer(Days,Ages,FUN="+")
Ls <- approxArray(length.df$Age,length.df$Length,ages)
ws <- powerLW(Ls,2.59E-9,3.2064)

## Build selectivity matrices
sel <- mkSelectivity(ages,Ls)
current.sel <- -1

## Constant intra-annual natural mortality
ms <- matrix(1,nsteps+1,length(Ages))
Ms <- ctrapz(ms,h)
Msf <- final(Ms)
M <- 0.13
MMs <- M*Ms

## Length based maturity
gs <- rampOgive(Ls,930,300)

## Within year fishing pattern
fwy <- double(nsteps+1)
fwy[] <- 1
fwy <- fwy/mean(fwy)

## Calculate recruitment parameters from historic data
## By method of moments
rmn <- mean(recruit.df$Rec,na.rm=TRUE)
rsd <- sd(recruit.df$Rec,na.rm=TRUE)
rmn <- 3016520
rsd <- 1.62693*rmn
rLsd <- sqrt(Log(1+(rsd/rmn)^2))
rLmn <- Log(rmn)-rLsd^2/2
## By maximum likelihood
#rLmn <- mean(log(recruit.df$Rec))
#rLsd <- sd(log(recruit.df$Rec))

## Drop missing and out of range recruitment estimates
recruit.df <- recruit.df[(recruit.df$Year %in% seq(Year1,length.

```

```

out=nrow(catch.df)+n.years)) &
                                !is.na(recruit.df$Rec),]

## This function performs a projection for each prescribed gamma.
function(run) {

  ## Median spawning biomass estimated from 1000 samples
  R <- matrix(rlnorm(1000*(Length(Msf)+plus),rLmn,rLsd),1000,Length(
Msf)+plus)
  ssb0 <- spawningB0S(R,gs,ws,Ms,M,spawn=spawnI,plus=TRUE)

  ## Stochastic initial age structure in the absence of fishing
  N0 <- ageStructureS(rlnorm(Length(Msf)+plus,rLmn,rLsd),Msf,M,plus=
TRUE)
  ## Initial projection assuming no fishing
  pr <- project(ws,MMS,Nref=N0,yield=0)
  pr$F <- pr$Y <- 0

  ## Recruitment series - log Normal + known from survey
  Rs <- rlnorm(nrow(catch.df)+n.years,rLmn,rLsd)
  Rs[recruit.df$Year-Year1+1] <- recruit.df$Rec

  ## Annual summary quantities
  n <- nrow(catch.df)+n.years*Length(Catches)
  Test<-rep(Catches, each=n/Length(Catches))
  Year <- integer(n)
  Target <- R <- N <- B <- SSN <- SSB <- Catch <- F <- double(n)
  k <- 1

  ## Project over historic period
  for(yr in 1:nrow(catch.df)) {

    ## Recompute fishing mortality when selectivity changes
    if(sel$index[yr]!=current.sel) {
      current.sel <- sel$index[yr]
      ss <- sel$ss[[current.sel]]
      fs <- fwy*ss
      Fs <- ctrapz(fs,h)
    }

    ## Project over year
    Year[k] <- catch.df$Year[yr]
    Target[k] <- catch.df$Catch[yr]
    R[k] <- Rs[yr]
    N0 <- advance(pr$N,R[k],plus=TRUE)
    pr <- projectC(ws,MMS,Fs,fs,Target[k],Nref=N0,yield=1,Fmax=5,tol
=1.0E-8)

    ## Collate annual summaries

```

```

    N[k] <- sum(initial(pr$N))
    B[k] <- sum(initial(pr$B))
    SSN[k] <- spawningStock(pr$N,gs,spawnI)
    SSB[k] <- spawningStock(pr$B,gs,spawnI)
    Catch[k] <- sum(pr$Y)
    F[k] <- pr$F
    k <- k+1
  }

  ## Record pre-projection state
  pr0 <- pr

  ## Set projection selectivity
  ss <- sel$ss[[1]]
  fs <- fwy*ss
  Fs <- ctrapz(fs,h)

  ## Project for each catch
  for(catch in Catches) {
    ## Reset to pre-projection state
    pr <- pr0

    for(yr in seq(nrow(catch.df)+1,length.out = n.years)) {
      ## Project over year
      N0 <- advance(pr$N,Rs[yr],plus=TRUE)
      pr <- projectC(ws,MMS,Fs,fs,catch,Nref=N0,yield=1,Fmax=5,tol=1
        .0E-8)

      ## Collate annual summaries
      Year[k] <- yr+Year1-1
      Target[k] <- catch
      R[k] <- Rs[yr]
      N[k] <- sum(initial(pr$N))
      B[k] <- sum(initial(pr$B))
      SSN[k] <- spawningStock(pr$N,gs,spawnI)
      SSB[k] <- spawningStock(pr$B,gs,spawnI)
      Catch[k] <- sum(pr$Y)
      F[k] <- pr$F
      k <- k+1
    }
  }
  data.frame(Run=run,M=M,Year=Year,Target=Target,
    R=R,N=N,B=B,SSN=SSN,SSB=SSB,SSB0=ssb0$median,Catch=Catc
    h,F=F)
}
}

```

## Projection

Calling *ToothfishProjection* returns a function that generates a single ‘run’ of the simulation

```
sim <- ToothfishProjection(Catches=c(0, 2.8e6, 2.84e6, 2.85e6, 2.9e6, 3e6, 3.2e6),
                           catch.df=catch.df, recruit.df=recruit.
                           df, length.df=length.df)
```

The function takes a single argument that is used to label the results. Within a single call to *sim*, a common set of random maturity, selectivity, initial ages and recruitment series are used to make projections for each requested target catch.

```
df <- sim(1)
head(df)
tail(df)
```

For these parameters, the recruitment variability has greater impact than increased fishing pressure

```
ggplot(df %>% filter(Year > 2005), aes(x=Year, y=N, colour=factor(Target)))
+geom_line()
```

Generate 10001 runs and bind them into one large dataframe. The runs are performed in parallel with the *furrr* library

```
plan(multiprocess)
system.time(df <- future_map_dfr(1:10001, sim))

saveRDS(df, "TOP_df_output.rds")
```

#Spawning Stock Status escapement at quantiles

```
grymssStatus <- df %>%
  filter(Year == 2040) %>%
  mutate(Status = SSB/SSB0) %>%
  group_by(Target/1000) %>%
  summarise(q2.5 = quantile(Status, 0.025),
            q45 = quantile(Status, 0.45),
            q50 = quantile(Status, 0.50),
            q55 = quantile(Status, 0.55),
            q97.5 = quantile(Status, 0.975),
            Median = median(Status)) %>%
  mutate(Model = "Grym")
```

*grymssStatus*

#Spawning Stock Status depletion

```
grymdep <- df %>%
  filter(Year > 2006) %>%
  group_by(Target, Run) %>%
  summarize(Status = min(SSB/SSB0)) %>%
```

```
summarise(Depletion_prob=mean(Status <= 0.2))
mutate(Model="Grym")
```

%>%

*grymdep*

Table 1: Grym package functions with brief descriptions. Typing ?function-name in R will display the full documentation for each function. \* in recruitment functions denotes a given distribution, use ?prRecruitPars to see help for all available distributions.

Functions	Description
vonBertalanffyAL, vonBertalanffyLA, vonBertalanffyRAL, vonBertalanffyRAL	von Bertalanffy age-length models, parameterised as in the GYM.
powerLW, powerWL rampOgive	Power law length weight models, parameterised as in the GYM. Ramp shaped ogive function for selectivity and maturity, parameterised as in the GYM.
Trapz, ctrapz, trapzMeans project, projectC, advance, rescaleProjection	Numerical quadrature by the composite trapezoidal rule. Project the abundance, biomass and yield in each age class forward over one year.
ageStructureD, ageStructureS spawningStock, vulnerableStock, meanStock, exploitableStock, initial, final spawningB0D, spawningB0S	Compute the initial age structure of the population. Calculate the stock summaries for a given monitoring period. Estimate virgin spawning stock biomass.
prRecruits*, prRecruitPars*, prRecruitsQuantile*, prBootstrap, resampleRGYM	Generate random recruits using the proportional recruits model.
surveySurvival, surveyAdjustGYM	Adjustment of the surveyed age-class abundances to initial abundances at a reference age class.

Table 2: Input parameters for Grym and GYM software assessment comparisons conducted for mackerel icefish (*Champscephalus gunnari*), Antarctic krill (*Euphausia superba*) and Patagonian toothfish (*Dissostichus eleginoides*).

Category	Parameter	Icefish	Krill	Toothfish
Ages	First age class	1	0	4
	Last age class	10	7	35
Von Bertalanffy growth	$t_0$	0.067	0	0
	$L_\infty$	490 mm	60 mm	170.8 cm
	$k$	0.368	0.45	0.088
	Date - start growth period (dd/mm)	1-Dec	1-Nov	30-Nov
	Date - end growth period (dd/mm)	30-Nov	1-Feb	30-Nov
Weight at length (kg, mm)	Weight-length parameter - A (kg)	1.08E-09	1	2.50E-05
	Weight-length parameter - B	3.286	3	2.8
Maturity	Min. length, 50% are mature	-	34 mm	930-
	Max. length, 50% are mature	-	40 mm	-
	Range over which maturity occurs	-	12 mm	300-
Spawning Season	First day of spawning season (dd/mm)	30-Nov	1-Mar	1-Jul
	Last day of spawning season (dd/mm)	30-Nov	1-Jun	1-Jul
Mortality	Min. mean Annual M	0.4	0.4	0.16
	Max. mean Annual M	0.4	1	0.16
Recruitment	Function	-	Lognormal	Lognormal
	Mean recruitment	-	1	See Appendix 3
	Min. coefficient of variation	-	0.4	1.162
	Max. coefficient of variation	-	0.6	1.162
	Cohorts to project	127.106, 617.426, 1988.91	-	-
Fishery parameters	Age fully selected	3	-	See Appendix 3
	Age first selected	2.5	-	See Appendix 3
	Min. length, 50% Selected	-	38 mm	-
	Max. length, 50% Selected	-	42 mm	-
	Range over which recruitment occurs	-	10 mm	-
	Season	1 Dec – 30 Nov	1 Dec – 28 Feb	-
Simulation specifications	Catch between survey and season	0	0	0
	Initial biomass	3723.761	-	-
	Number of runs in simulation	1	10001	10001
Individual trial specifications	Evaluation type	Fishing mortality	Gamma	Constant catch
	Years to remove initial age structure	1	1	1
	Year prior to projection	2019	2005	-
	Reference start date in year	1-Dec	1-Nov	1-Dec
	Increments in year	365	365	24
	Years to project stock in simulation	2	20	35
	Reasonable upper bound for annual F	5	1.5	5
	Tolerance for finding F in each year	0.000001	0.000001	0.000001
Target escapement	75%	75%	50%	

Table 3: Comparison of the GYM software and Grym implementation of the 2019 mackerel icefish (*Champsocephalus gunnari*) assessment in Division 58.5.2 (Maschette et al., 2019).

Year	Spawning numbers		Spawning Biomass		Catch		Escapement	
	GYM	Grym	GYM	Grym	GYM	Grym	GYM	Grym
0	14928.0	14640.5	3986.1	3970.5	0.0	0.0	1.000	1.000
1	8673.4	8532.4	3268.0	3240.3	526.9	523.1	0.820	0.816
2	5032.5	4943.0	2369.3	2339.0	406.3	406.0	0.594	0.589

Table 4: Comparison of probability of depletion, and median spawning stock status, as well as the 10, 25, 50, 75 and 90% quantiles in the 1996 Antarctic krill (*Euphausia superba*) assessment presented in Constable and de la Mare (1996) at year 20 in the projection period under various constant gamma scenarios when fitted within the GYM software and the Grym package. Note that some estimates by Constable and de la Mare (1996) differ from those estimates using the GYM version 501E here due to internal changes in the GYM between 1996 and 2004 (Constable, 2004).

Gamma	Model	Depletion probability	Spawning stock status				
			10%	25%	Median%	75%	90%
0.000	Grym	0.000	0.75	0.86	1.00	1.17	1.35
0.000	GYM	0.000	0.76	0.87	1.00	1.16	1.34
0.100	Grym	0.096	0.35	0.51	0.68	0.86	1.04
0.100	GYM	0.097	0.35	0.50	0.67	0.85	1.04
0.136	Grym	0.267	0.18	0.36	0.55	0.75	0.94
0.136	GYM	0.271	0.18	0.35	0.54	0.74	0.94
0.150	Grym	0.340	0.14	0.30	0.50	0.71	0.90
0.150	GYM	0.347	0.15	0.29	0.49	0.69	0.89
0.200	Grym	0.567	0.07	0.18	0.34	0.55	0.76
0.200	GYM	0.571	0.08	0.19	0.33	0.54	0.76

Table 5: Comparison of the spawning stock status from the GYM software and Grym implementations of the 2006 Patagonian toothfish (*Dissostichus eleginoides*) assessment in Division 58.5.2 (Welsford et al., 2006) at year 35 in the projection period under various constant catch scenarios.

Target Catch	Model	Depletion probability	Spawning stock status				
			10%	25%	Median	75%	90%
0	GYM	0.000	0.696	0.805	0.961	1.172	1.420
0	Grym	0.000	0.699	0.814	0.972	1.176	1.420
2800	GYM	0.050	0.268	0.363	0.506	0.704	0.939
2800	Grym	0.053	0.267	0.367	0.511	0.703	0.933
2850	GYM	0.058	0.260	0.355	0.498	0.695	0.931
2850	Grym	0.060	0.259	0.359	0.503	0.695	0.924
2900	GYM	0.067	0.252	0.347	0.490	0.686	0.922
2900	Grym	0.070	0.251	0.351	0.494	0.686	0.915
3000	GYM	0.084	0.237	0.331	0.473	0.668	0.904
3000	Grym	0.089	0.235	0.334	0.477	0.668	0.897
3200	GYM	0.126	0.206	0.298	0.439	0.633	0.868
3200	Grym	0.133	0.203	0.302	0.443	0.633	0.861



Table 6: Comparison of the spawning biomass from the GYM software and Grym implementations of the 2006 Patagonian toothfish (*Dissostichus eleginoides*) assessment in Division 58.5.2 (Welsford et al., 2006) at year 35 in the projection period under various constant catch scenarios.

Target Catch	Model	Depletion probability	Spawning biomass				
			10%	25%	Median	75%	90%
0	GYM	0.000	76413	88388	105547	128772	156048
0	Grym	0.000	76129	88697	105957	128224	154769
2800	GYM	0.050	29404	39856	55541	77123	102829
2800	Grym	0.053	29087	40074	55813	76678	102062
2850	GYM	0.058	28542	38974	54639	76168	101913
2850	Grym	0.060	28213	39168	54891	75698	101037
2900	GYM	0.067	27714	38053	53716	75206	100951
2900	Grym	0.070	27354	38284	53970	74713	100026
3000	GYM	0.084	26004	36289	51839	73257	99025
3000	Grym	0.089	25572	36514	52085	72789	97978
3200	GYM	0.126	22563	32681	48123	69409	95138
3200	Grym	0.133	22121	32946	48415	68937	93929

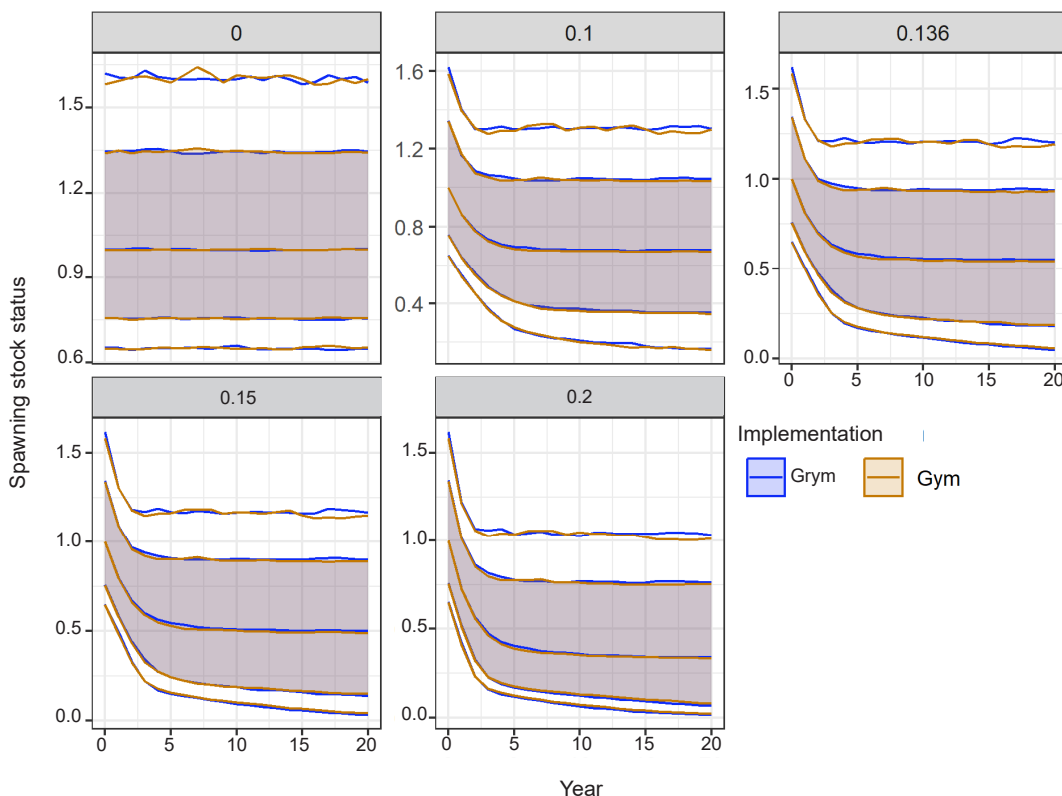


Figure 1: Grym and GYM software assessment projection comparisons for Antarctic krill (*Euphausia superba*) under various gamma tests over 20 years showing 90% and 80% confidence intervals with median.

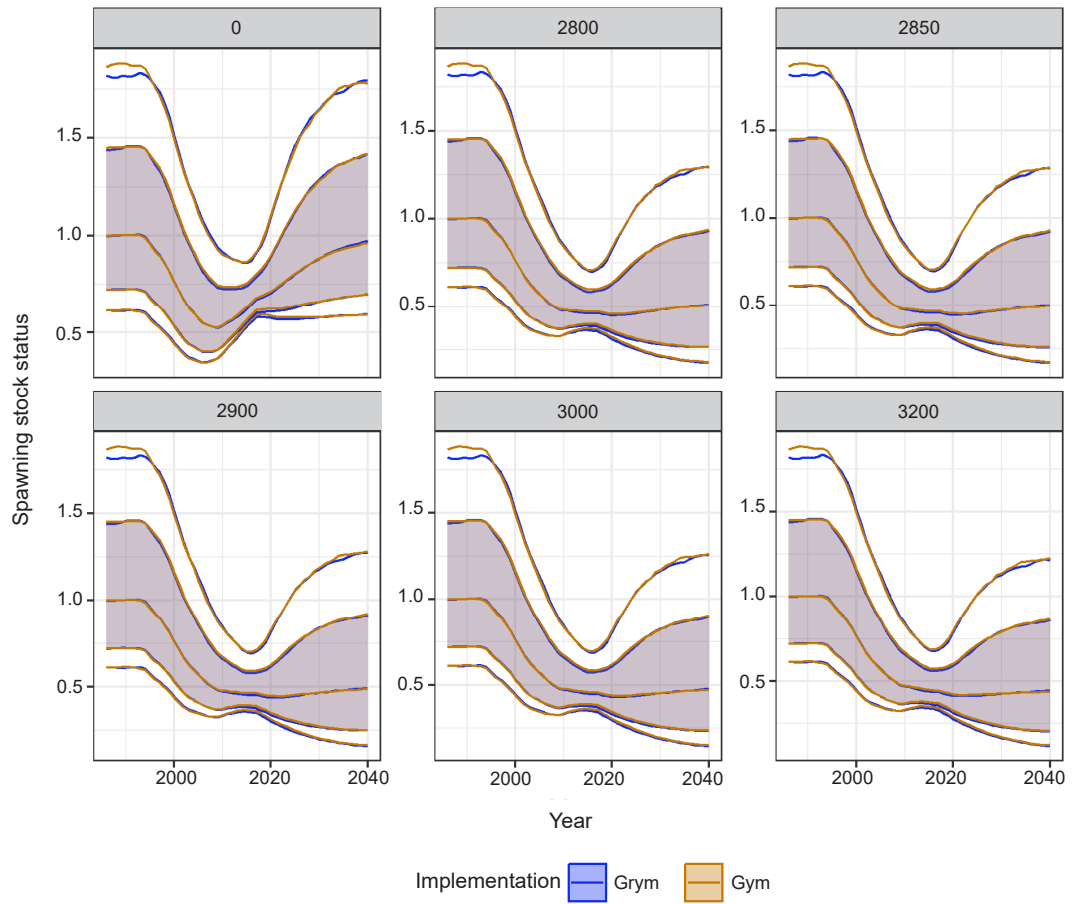


Figure 2: Grym and GYM software assessment projection comparisons for Patagonian toothfish (*Dissostichus eleginoides*) under constant catch scenarios over 35 years showing 90% and 80% confidence intervals with median.