

# A Framework for Evaluating Single-Species Assessment and Management of Multi-Species Fisheries

Adrian Hordyk\*

28 October 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Overview of the Case Study . . . . .	3
1.2	Overview of the PFMC's P* Approach . . . . .	3
1.3	Technical Details of the Framework . . . . .	4
<b>2</b>	<b>Generating Multi-Species Operating Models</b>	<b>4</b>
2.1	Stock Objects . . . . .	4
2.1.1	Blue rockfish stock . . . . .	5
2.1.2	Deacon rockfish stock . . . . .	6
2.2	Fleet Objects . . . . .	6
2.3	Obs and Imp Objects . . . . .	9
2.4	Custom Parameters . . . . .	9
2.4.1	Correlated Recruitment Deviations . . . . .	9
2.4.2	Catchability . . . . .	10
2.5	Building the MOM Object . . . . .	12
<b>3</b>	<b>Developing Management Procedures</b>	<b>12</b>
3.0.1	Single-Species Stock Assessment of Multi-Species Stock . . . . .	12
3.0.2	Stock-Specific Management . . . . .	15
<b>4</b>	<b>Closed-Loop Simulation Testing</b>	<b>17</b>
4.1	Running the Closed-Loop Simulation . . . . .	17
4.2	Examining the MMSE Object . . . . .	18
4.3	Calculating Performance of the MPs . . . . .	19
4.3.1	Mean Relative Spawning Biomass . . . . .	21
4.3.2	Mean Relative Yield for each Stock . . . . .	22
4.3.3	Probability $SB > 0.25SB_0$ in each year . . . . .	23
4.3.4	Trade-off plots of Prob $SB > 0.25SB_0$ and Yield in last 10 years . . . . .	24

---

\*Blue Matter Science Ltd, [adrian@bluematterscience.com](mailto:adrian@bluematterscience.com)

## 1 Introduction

The National Oceanic and Atmospheric Administration’s (NOAA) [National Standard 1](#) states that fish stocks can be grouped together into complexes for reasons including “*where stocks in a multispecies fishery cannot be targeted independent of one another; where there is insufficient data to measure a stock’s status relative to SDC [status determination criteria]; or when it is not feasible for fishermen to distinguish individual stocks among their catch.*” Following these guidelines, fishery managers in the U.S. sometimes group similar species together into complexes for assessment and management purposes. In other cases, the development of relatively inexpensive tools for rapid DNA sequencing has led to the discovery that stocks that are assessed and managed as a single species are in fact made up of several cryptic species; genetically distinct and often reproductively isolated populations that were previously classified as a single nominal species (Bickford et al., 2007).

The individual species within a multi-species fishery, whether unknown cryptic species or species grouped together for practical purposes, are likely to differ in terms of life-history parameters (e.g., growth, natural mortality, stock-recruitment relationship) and subsequently will differ in terms of their biological reference points such as maximum sustainable yield (MSY), and the equilibrium biomass and fishing mortality ( $B_{MSY}$  and  $F_{MSY}$  respectively) that correspond with MSY. This stock-specific variability is not explicitly accounted for when the fishery is managed as a single species. For example, stock assessments often treat a multi-species fishery as a single stock with common life-history characteristics, and therefore are expected to under-estimate the target exploitation rate for the more productive stocks within the complex and over-estimate it for those that are less productive.

The potential problems with assessing and managing a stock complex as if it were a single species have been generally recognized (Bizzarro et al., 2020; DeMartini, 2019; Dick & MacCall, 2010; Hicks et al., 2014). The additional scientific uncertainty associated with managing multiple species as a complex has often been accounted for by increasing the ‘uncertainty buffers’ that are used to alter the prescribed management advice [e.g., reduce a catch limit; Hicks et al. (2014)]. However, these uncertainty buffers have usually been developed in the context of single-species management and their performance for the species within a stock complex has rarely been quantified (DeMartini, 2019). Consequently, there is a need for a standardized framework for quantifying the performance of management approaches that manage several species within a stock complex as a single species, and comparing the risks to those associated with managing the individual species separately.

The purpose of this document is to provide a reproducible example of how the [openMSE](#) framework can be used to evaluate the performance of alternative management options for multispecies fisheries that are managed assuming a single stock. We use a case study from the Pacific Fishery Management Council’s (PFMC) Groundfish Fishery Management Plan (FMP) together with the PFMC’s uncertainty buffer approach to demonstrate the approach. However, the framework is designed to be flexible, and it should be clear from this document how it can be used to evaluate the performance of alternative assessment and management

models the used by the PFMC and other regional management councils.

## 1.1 Overview of the Case Study

The case study is based on the blue-deacon rockfish complex, managed by the PFMC and consisting of two species, blue (*Sebastes mystinus*) and deacon rockfish (*S. diaconus*). The stocks were thought to be a single species until morphometric and microsatellite genetic analysis by Frable et al. (2015) confirmed they were two genetically distinct stocks. Although now recognized as distinct species, because of the inability to distinguish historical landings by species and a lack of species-specific life-history information the two stocks are still assessed and managed as a complex (Bizzarro et al., 2020; Dick et al., 2017).

For simplicity and clarity, we restrict this demonstration of the framework to the case of 2 species in a multi-species fishery (e.g., cryptic species) being assessed with a single population dynamics model. However, the process outlined in this document can be easily be modified to model a complex with 3 or more species.

## 1.2 Overview of the PFMC's P\* Approach

The PFMC uses an approach known as P\* (p-star) to set the catch limits for the groundfish stocks. The P\* approach involves three steps. First, an overfishing limit (OFL) is calculated as the expected catch obtained from applying the maximum acceptable fishing mortality rate ( $F_{lim}$ ) to the estimated current stock biomass. For rockfish, the PFMC has defined  $F_{lim}$  as the fishing mortality rate corresponding with a 50% spawning potential ratio (SPR);  $F_{50\%}$ .

Next, an acceptable biological catch (ABC) limit is calculated by applying a multiplicative uncertainty buffer that reduces the OFL to account for scientific uncertainty, a term collectively referring to the underlying stochastic variability in the population dynamics (i.e., process error), observation error in the data collection process, and uncertainty due to the structure of the assessment model and error in the assumed and estimated parameters. The uncertainty buffer used to calculate the ABC from the OFL is determined using the P\* approach, where P\* is the percentile of the log-normal OFL distribution with standard deviation  $\sigma_{OFL}$ . P\* must never exceed 0.5 and is typically set to values of 0.4 or 0.45.

Finally, an annual catch limit (ACL) is calculated by applying a 40-10 harvest control rule (HCR) that reduces linearly the ACL from the ABC when the estimated relative spawning biomass ( $SB/SB_0$ ) is below 0.4, with ACL set to zero when  $\frac{SB}{SB_0} < 0.1$

The PFMC places groundfish stocks into 3 categories based on the quality of the data and assessment and specify minimum values for  $\sigma_{OFL}$  of 0.5 (Category 1), 1 (Category 2), and 2 (Category 3). Coupled with  $P^*=0.45$ , this results in an uncertainty buffer of 0.939, 0.882, and 0.778 for Categories 1 – 3 respectively; equivalent to setting the ABC 6.1%, 11.8% and 22.2% lower than the OFL.

Recently the PFMC adopted a new policy to account for additional uncertainty in the years between stock assessments by increasing  $\sigma_{OFL}$  for Category 1 and 2 stocks by 7.5% each year since the most recent assessment (Anon., 2019). The  $\sigma_{OFL}$  is reset to the baseline level when a full or update assessment is conducted.

### 1.3 Technical Details of the Framework

The framework has been developed with [openMSE](#), an open-source software package that includes the [MSEtool](#), [SAMtool](#), and [DLMtool](#) R packages.

The framework requires the most recent version of the openMSE R package:

```
install.packages('openMSE')
```

This will also install the most recent versions of MSEtool, SAMtool, and DLMtool:

```
packageVersion('MSEtool')
```

```
## [1] '3.2.909'
```

```
packageVersion('SAMtool')
```

```
## [1] '1.2.2'
```

```
packageVersion('DLMtool')
```

```
## [1] '6.0.3'
```

## 2 Generating Multi-Species Operating Models

Multi-stock operating models contain parameters describing the biological characteristics of the stocks within the multi-species fishery, the properties of the fleet(s) exploiting them, the observation processes for collecting the fishery data, and any implementation error of the prescribed management advice. This section demonstrates the process for creating multi-stock operating models within the openMSE framework.

The framework is flexible and the same approach can be used to generate multi-stock operating models with 3 or more stocks within the complex. Similarly, while we assume male and females within each species share the same biological parameters, if desired, users can follow this same approach and generate separate stock objects for the female and male components of the stock, including stocks that exhibit hermaphroditism where some males transform to females (or vice versa).

Our model assumed no spatial structure for the case studies. However, if sufficient information exists, users can specify the stock-specific movement patterns over an unlimited number of spatial areas, and control the spatial targeting behavior of the fishing fleet(s), including spatio-temporal closures to fishing.

To create an multi-stock operating model, we need to create **Stock**, **Fleet**, **Obs**, and **Imp** objects for each stock within the multi-species fishery.

### 2.1 Stock Objects

[Stock objects](#) contain the biological parameters for the modeled population. By default, the parameters in the **Stock** object are stored in the slots as upper and lower bounds of uniform distribution. In this example, we have populated the **Stock** objects assuming no uncertainty in the life-history parameters (i.e., all simulations

have the same value for each parameter). In more sophisticated analyses, it is common practice to generate multiple operating models that span the uncertainties in the fishery dynamics (e.g., range of values of  $M$ , etc). This can be done in two ways:

1. setting the range of a uniform distribution within a single operating model, e.g., `Stock@M <- c(0.1, 0.2)`; or
2. using the [Custom Parameters](#) feature to provide specific, often correlated, values for the parameters.

The life-history parameters used in this demonstration of the framework are based on those reported in the most recent stock assessment of the blue-deacon fishery (Dick et al., 2017). We have assumed some life-history parameters for the blue and deacon rockfish stocks in this analysis, and all parameter values presented here are for demonstration purposes only.

### 2.1.1 Blue rockfish stock

New stock objects are created with the `new` command. We create an empty stock object for the blue rockfish and populate the slots with the assumed biological parameters:

```
Blue <- new('Stock')
Blue@Name <- 'Blue'
Blue@M <- c(0.119, 0.119) # Natural mortality - all simulations same value
Blue@maxage <- 35
Blue@h <- c(0.64,0.64) # steepness
Blue@SRrel <- 1 # BH SRR
Blue@Linf <- c(355,355)
Blue@K <- c(0.15,0.15)
Blue@t0 <- c(-3.04, -3.04)
Blue@L50 <- c(222, 222)
Blue@L50_95 <- c(23,23) # increment between L50 and L95
Blue@Perr <- c(0.5,0.5) # sigma R for rec-devs
Blue@AC <- c(0,0) # auto-correlation in rec-devs
Blue@a <- 3.15E-5 # alpha in L-W relationship
Blue@b <- 2.88 # beta in L-W relationship
Blue@LenCV <- c(0.1,0.1) # CV of length-at-age
Blue@R0 <- 10000 # assumed unfished recruitment
# Spatial parameters
# (assumed fully mixed stock in two equal areas; ie one area)
# Models have 2 areas by default
Blue@Size_area_1 <- Blue@Frac_area_1 <- Blue@Prob_staying <- c(0.5,0.5)
# Time-varying parameters
# (assumed stationarity)
Blue@Msd <- Blue@Linfsd <- Blue@Ksd <- c(0,0)
# Discard mortality
# (assumed none)
```

```
Blue@Fdisc <- c(0,0)
# relative spawning biomass (depletion) at the end
# of the historical period (not used as we are providing actual F by year in cpars)
Blue@D <- c(0,0)
```

### 2.1.2 Deacon rockfish stock

The `Stock` object for the deacon rockfish is created in a similar manner. In this example, the deacon stock shares identical life-history parameters to the blue rockfish, except for the von Bertalanffy growth and the maturity-at-length parameters:

```
Deacon <- new('Stock')
Deacon@Name <- 'Deacon'
Deacon@M <- c(0.119, 0.119)
Deacon@maxage <- 35
Deacon@h <- c(0.64,0.64)
Deacon@SRrel <- 1
Deacon@Linf <- c(413, 413)
Deacon@K <- c(0.13,0.13)
Deacon@t0 <- c(-2.81, -2.81)
Deacon@L50 <- c(238, 238)
Deacon@L50_95 <- c(28,28)
Deacon@Perr <- c(0.5,0.5)
Deacon@AC <- c(0,0)
Deacon@a <- 3.15E-5
Deacon@b <- 2.88
Deacon@LenCV <- c(0.1,0.1)
Deacon@R0 <- 10000
Deacon@Size_area_1 <- Deacon@Frac_area_1 <- Deacon@Prob_staying <- c(0.5,0.5)
Deacon@Msd <- Deacon@Linfmsd <- Deacon@Ksd <- c(0,0)
Deacon@Fdisc <- c(0,0)
Deacon@D <- c(0,0)
```

## 2.2 Fleet Objects

The fleet dynamics are stored in a `Fleet` object. We assume a single fleet operating on the two fully-mixed stocks in the multi-species fishery and use the overall fleet characteristics estimated from the multiple fleets included in assessment (Dick et al., 2017). Multiple individual fleets can be modeled if fleet-specific management is required.

The `Fleet` object contains all information relating to the exploitation of the stocks:

```
BDFleet <- new('Fleet')
BDFleet@Name <- "Blue-Deacon Fleet"
```

```

BDFleet@nyears <- 117 # number of years since fishery began
BDFleet@CurrentYr <- 2016 # most recent year of data used in the assessment
BDFleet@L5 <- c(185, 185) # first length at 5% selection
BDFleet@LFS <- c(315, 315) # first length at full selection
BDFleet@Vmaxlen <- c(1,1) # vulnerability at maximum length (assumed asymptotic)

```

The selectivity-at-length parameters are based on the overall  $F$ -at-age estimated by the 2017 stock assessment for each year; i.e., summing the predicted  $F$ -at-age across the fleets and standardizing to a maximum value of 1. Time-varying selectivity (and retention) patterns can be specified using the [Custom Parameters](#) feature.

The `Fleet` object also requires the pattern in historical fishing effort. We calculate the apical  $F$  as the maximum overall  $F$ -at-age estimated by the 2017 stock assessment.

The blue-deacon assessment was conducted using Stock Synthesis 3 (SS3). The assessment files can be downloaded from the PFMC's [groundfish stock assessment website](#).

We use the `SS2OM` function to import the SS3 assessment into a single-stock, aggregated-fleet openMSE operating model:

```

SSdir <- 'Data/blue-deacon-rockfish-CA-SS' # path to SS3 output
OMdummy <- SS2OM(SSdir, nsim=2) # dummy operating model

```

And then enter the overall apical  $F$  by year from the assessment in the `EffLower` and `EffUpper` slots in the `Fleet` object (Figure 1):

```

Find <- OMdummy@cparams$Find[1,] # pattern in apical F estimated by SS3

# Both stocks have same pattern in F
BDFleet@EffYears <- seq(OMdummy@CurrentYr-OMdummy@nyears+1,
                       OMdummy@CurrentYr,
                       by=1)
BDFleet@EffLower <- Find # all simulations have same F-pattern
BDFleet@EffUpper <- Find

```

The remaining parameters in the `Fleet` object are populated with default values:

```

# no additional variability on catchability
BDFleet@Esd <- BDFleet@qinc <- BDFleet@qcv <- c(0,0)
# no retention curve (all selected fish are retained)
BDFleet@LR5 <- BDFleet@LFR <- c(0,0)
BDFleet@Rmaxlen <- c(1,1)
# Selectivity parameters are not relative to length-at-maturity
BDFleet@isRel <- FALSE
# No general discarding
BDFleet@DR <- c(0,0)
# F is distributed across areas proportional to vulnerable biomass
BDFleet@Spat_targ <- c(1,1)

```

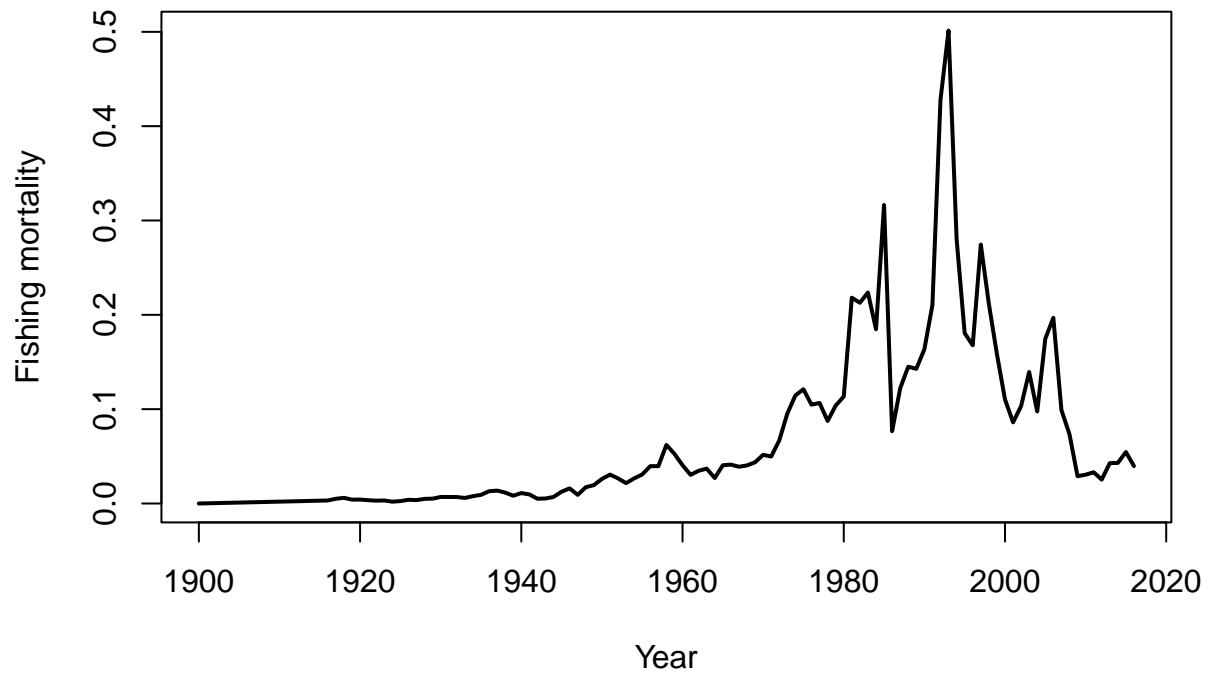


Figure 1: The pattern in historical fishing mortality for the two stocks in the blue-deacon fishery



## 2.3 Obs and Imp Objects

The observation parameters are stored in an [Obs](#) object. For demonstration purposes, we use the example `Generic_Obs` object from the `MSEtool` package for both stocks:

```
BObs <- Generic_Obs
DObs <- Generic_Obs
```

We also assume the management advice will be not have any implementation error by using the `Perfect_Imp` [Imp](#) object:

```
BImp <- Perfect_Imp
DImp <- Perfect_Imp
```

These assumptions can easily be modified by replacing the `Generic_Obs` and `Perfect_Imp` objects with other `Obs` and `Imp` objects included in `MSEtool` (e.g., see `avail('Obs')` and `avail('Imp')`) or customized by modifying the values in the slots in the `BObs`, `DObs`, `BImp`, and `DImp` objects (see [Obs Object](#) and [Imp Object](#) on the `openMSE` website for details).

## 2.4 Custom Parameters

We will use the [Custom Parameters](#) feature (referred to as *cpars*) to correlate the recruitment deviations of the two stocks and to set the catchability coefficient to 1 (i.e., use the apical *F* shown above for the historical spool-up period).

First, we create two empty lists to store the custom parameters:

```
Blue_cpars <- list()
Deacon_cpars <- list()
```

### 2.4.1 Correlated Recruitment Deviations

This step is only required if users wish to have a correlation structure in the recruitment deviations of stocks within the multi-species fishery. If no correlation structure is desired between the stocks, the recruitment deviations are calculated internally in the model and the following functions are not required.

We create a function to generate `nsim` samples of the annual recruitment deviations, where `nsim` is the number of simulations:

```
Generate_RecDevs <- function(nsim, Stock, nyears, proyears) {
  sigmaR <- runif(nsim, Stock@Perr[1], Stock@Perr[2])
  AC <- runif(nsim, Stock@AC[1], Stock@AC[2])
  procmu <- -0.5 * sigmaR^2 * (1 - AC)/sqrt(1 - AC^2)
  maxage <- Stock@maxage
  Perr_y <- array(rnorm((nyears + proyears+maxage) * nsim,
                       rep(procmu, nyears + proyears+maxage),
                       rep(sigmaR, nyears + proyears+maxage)),
```

```

        c(nsim, nyears + proyears+maxage))
# add auto-correlation (if any)
for (y in 2:(nyears + proyears+maxage))
  Perr_y[, y] <- AC * Perr_y[, y - 1] + Perr_y[, y] * (1 - AC * AC)^0.5
exp(Perr_y)
}

```

Then generate independent recruitment deviations for the blue and rockfish stocks with 50 simulations for a 50-year projection period:

```

nsim <- 50 # number of simulations
proyears <- 50 # number of projection years
Blue_cpars$Perr_y <- Generate_RecDevs(nsim=nsim, Stock=Blue,
                                     nyears=BDFleet@nyears, proyears=proyears)
Deacon_cpars$Perr_y <- Generate_RecDevs(nsim=nsim, Stock=Deacon,
                                       nyears=BDFleet@nyears, proyears=proyears)

```

To keep the run time of the simulations down, we are using `nsim=50`. In real applications it may be necessary to increase `nsim` to obtain stable results.

Finally, we generate a correlation structure between the two sets of annual recruitment deviations using the `cornode` function from the `mc2d` R package (Figure 2):

```

AddCorrelation <- function(cpars, target=0.5) {
  tt <- require(mc2d)
  if (!tt) stop('Package `mc2d` required for this function')

  Perr_y_1 <- cpars[[1]]$Perr_y
  Perr_y_2 <- cpars[[2]]$Perr_y
  nsim <- dim(Perr_y_1)[1]
  for (i in 1:nsim) {
    tt <- mc2d::cornode(cbind(Perr_y_1[i,],Perr_y_2[i,]), target=target)
    cpars[[1]]$Perr_y[i,] <- tt[,1]
    cpars[[2]]$Perr_y[i,] <- tt[,2]
  }
  cpars
}
cpars <- AddCorrelation(list(Blue_cpars, Deacon_cpars), target=0.5)

```

## 2.4.2 Catchability

Because we set the actual apical  $F$ -by-year in the `Fleet` object, rather than a general scale-free pattern in fishing effort, we need to tell the model to fix the catchability coefficient ( $q$ ) to 1 for all simulations:

```

cpars[[1]]$qs <- rep(1, nsim)
cpars[[2]]$qs <- rep(1, nsim)

```

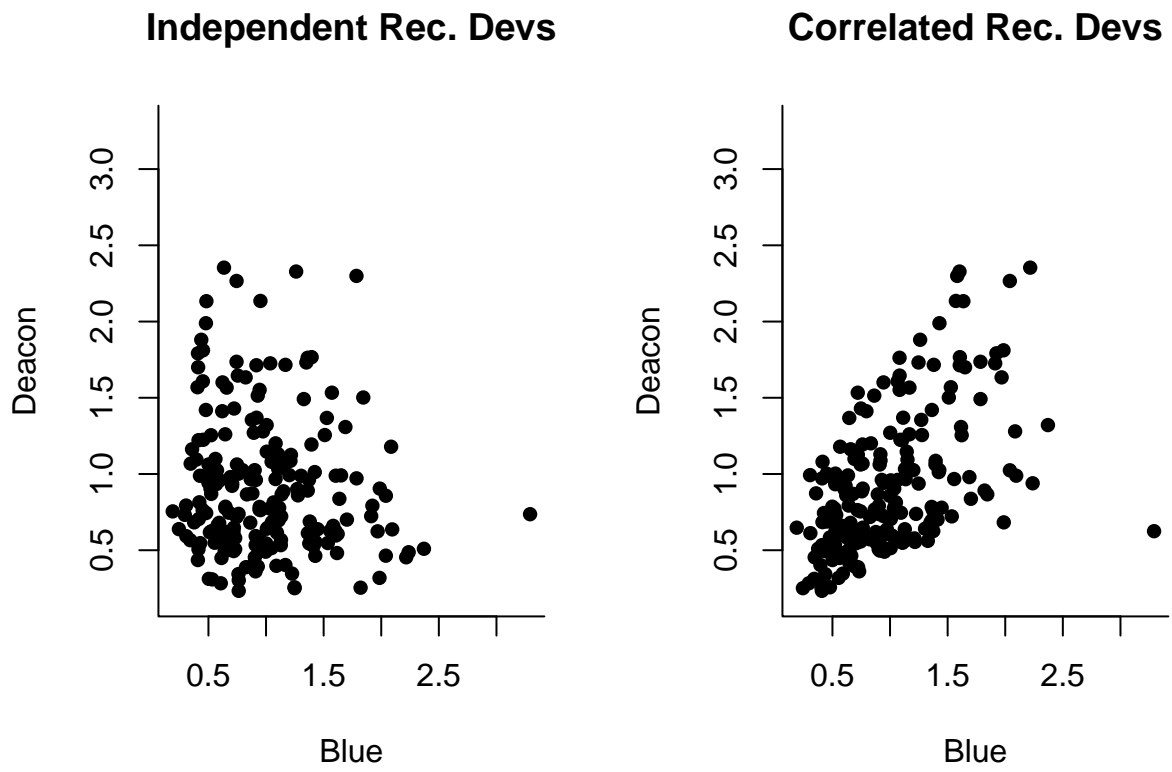


Figure 2: An example of the independent and correlated recruitment deviations.

## 2.5 Building the MOM Object

The components of the multi-stock operating model are stitched together into a object of class `MOM` (Multi-Operating Model):

```
BD_MOM <- new("MOM",
  Stocks=list(Blue, # blue Stock
             Deacon), # deacon Stock
  Fleets=list(list(BDFleet), # blue Fleet
             list(BDFleet)), # deacon Fleet
  Obs=list(list(BObs), # blue Obs
          list(DObs)), # deacon Obs
  Imps=list(list(BImp), # blue Imp
          list(DImp)), # deacon Imp
  CatchFrac=NULL,
  proyears=proyears, # number of projection years
  cpars=list( # cpars
    list(cpars[[1]]), # blue cpars
    list(cpars[[2]] # deacon cpars
  ),
  nsim=nsim, # number of simulations
  maxF=3,
  interval=1) # management procedure applied every year
```

## 3 Developing Management Procedures

In this section we demonstrate the development of example management procedures (MPs) that can be used to evaluate the PFMC's P\* approach to the simulated multispecies fishery. For comparison, we also include management procedures that provide stock-specific management advice (i.e., assessing and managing each species separately with the P\* approach).

### 3.0.1 Single-Species Stock Assessment of Multi-Species Stock

A stock assessment model estimates the vulnerable biomass, relative spawning biomass, and overfishing limit from the aggregated multi-species fishery data. Here we demonstrate an example MP that uses the statistical catch-at-age assessment model that uses Pope's approximation (`SCA_Pope`) from the `SAMtool` package.

In this demonstration of the framework, we calculate  $F_{50\%}$  using the mean life-history parameters from the two stocks within the multi-species fishery.

We create a generic MP function `SCA_template` that can be copied and modified to create other MPs.

Here `x` is simulation number, ranging from 1 to `nsim`, `Data` is a [Data object](#) containing the aggregated multi-species fishery data, `sigma`, and `pstar` are arguments specifying the uncertainty buffer, `interval` is

the assessment interval (the MP is run every year, but the SCA is only run in interval years), and `incbuffer` is a logical argument specifying whether the uncertainty buffer should be increased in the years between assessments.

```
SCA_template <- function(x, Data, sigma=0.5, pstar=0.45,
                        interval=4, incbuffer=TRUE, ...) {

  update_years <- seq(1, by=interval, to=200)
  yr_ind <- max(Data@Year)-Data@LHYear +1
  if (yr_ind %in%update_years) { # run assessment and update advice
    # run the SCA model
    run_sca <- SAMtool::SCA_Pope(x, Data)
    # predicted vulnerable biomass
    VB <- run_sca@VB[length(run_sca@VB)]
    # relative spawning biomass (depletion)
    Dep <- run_sca@SSB_SSB0[length(run_sca@SSB_SSB0)]
    # Flim (F50% SPR)
    ind <- which.min(abs(run_sca@forecast$per_recruit$SPR-0.5))
    Ulim <- run_sca@forecast$per_recruit$U[ind]
    Flim <- 1-exp(-Ulim)
    # Calculate OFL (Flim x VB)
    Vula <- run_sca@TMB_report$vvul
    Fage <- Flim * Vula
    Mage <- rep(Data@Mort[x], length(Fage))
    Zage <- Fage + Mage
    Nage <- run_sca@N_at_age[nrow(run_sca@N_at_age),]
    Ca <- Fage/Zage*(1-exp(-Zage))*Nage
    OFL <- sum(Ca*run_sca@obj$env$data$weight)
    # Apply P* uncertainty buffer
    buffer <- exp(stats::qnorm(pstar, 0, sigma))
    ABC <- OFL*buffer
    # Apply 40-10 HCR
    ACL <- HCR4010(Dep, ABC)
  } else {
    # non-assessment year
    # projection and apply buffer reduction
    lst_assess <- max(update_years[update_years<yr_ind])
    yr_since <- yr_ind - lst_assess
    run_sca <- Data@Misc[[x]]$assess
    Flim <- Data@Misc[[x]]$Flim
    p <- SAMtool::projection(run_sca, Ftarget = Flim, p_years=yr_since+1,
                            process_error = 0, obs_error=list(0,0))
    newDep <- p@VB[1,yr_since+1]/run_sca@VBO
    OFL <- p@Catch[1,yr_since+1]
  }
}
```

```

# increasing buffer
if (!incbuffer) {
  # constant buffer
  new_buffers <- rep(exp(stats::qnorm(pstar, 0, sigma)), interval+1)
} else {
  new_buffers <- get_buffer(1:(interval+1), sigma, pstar)
}
new_buffer <- new_buffers[yr_since+1]
ABC <- OFL*new_buffer
ACL <- HCR4010(newDep, ABC)
}
# create the management recommendation object
Rec <- new("Rec")
# store the ACL in the TAC slot
Rec@TAC <- ACL
# store the assessment results
Rec@Misc$assess <- run_sca
Rec@Misc$Flim <-Flim
# return Rec object
Rec
}
class(SCA_template) <- 'MP'

```

All MP functions must be assigned to class MP.

The 40-10 HCR used by the SCA MP is coded as:

```

HCR4010 <- function(Dep, ABC, b1=0.1, b2=0.4) {
  if (Dep<b1) return(0)
  if (Dep>b2) return(ABC)
  ABC * (-(b1*b2)/(Dep*(b2-b1))+(b2/(b2-b1)))
}

```

And the uncertainty buffer is updated in the years between assessments with:

```

get_buffer <- function(years, sigma, pstar){
  y <- 1:(length(years) - 1)
  r <- 0.075
  sigma_inc <- c(sigma, sigma * (1+(y)*r))
  buffer <- exp(stats::qnorm(pstar, 0, sigma_inc))
  round(buffer,3)
}

```

We can easily create additional MPs by copying SCA\_template and modifying the sigma, pstar, and interval arguments:

```

# combined stock assessment
MP_1_4_cs <- SCA_template

MP_2_4_cs <- SCA_template
formals(MP_2_4_cs)$sigma <- 1

MP_3_4_cs <- SCA_template
formals(MP_3_4_cs)$sigma <- 2

# constant buffer in years between assessments
MP_4_4_cs <- SCA_template
formals(MP_4_4_cs)$sigma <- 2
formals(MP_4_4_cs)$pstar <- 0.4
formals(MP_4_4_cs)$incbuffer <- FALSE

# 10-year interval
MP_1_10_cs <- SCA_template
formals(MP_1_10_cs)$interval <- 10

MP_2_10_cs <- SCA_template
formals(MP_2_10_cs)$interval <- 10
formals(MP_2_10_cs)$sigma <-1

MP_3_10_cs <- SCA_template
formals(MP_3_10_cs)$interval <- 10
formals(MP_3_10_cs)$sigma <-2

MP_4_10_cs <- SCA_template
formals(MP_4_10_cs)$interval <- 10
formals(MP_4_10_cs)$sigma <- 2
formals(MP_4_10_cs)$pstar <- 0.4
formals(MP_4_10_cs)$incbuffer <- FALSE

# assign all functions to class `MP`
class(MP_1_4_cs) <- class(MP_2_4_cs) <- class(MP_3_4_cs) <-
  class(MP_4_4_cs) <- class(MP_1_10_cs) <- class(MP_2_10_cs) <-
  class(MP_3_10_cs) <- class(MP_4_10_cs) <- 'MP'

```

### 3.0.2 Stock-Specific Management

Next, we create a template management procedure that use stock-specific data and provides stock-specific management advice.

```

SCA_template_ms <- function(x, DataList, sigma=0.5, pstar=0.45,
                           interval=4, incbuffer=TRUE, ...) {
  rec_out <- list()
  # number of stocks
  ns <- length(DataList)
  for (st in 1:ns) {
    Data <- DataList[[st]][[1]] # stock st, fleet 1
    rec_out[[st]] <- list(SCA_template(x, Data, sigma=sigma, pstar=pstar,
                                       interval=interval, incbuffer=incbuffer, ...))
  }
  rec_out
}
class(SCA_template_ms) <- 'MMP'

```

Here `DataList` is a list of stock-specific `Data` objects. This MP is of class `MMP` (multi-MP) and returns a list of management recommendations, one for each stock (and, if there are more than 1, each fleet) in the complex.

We can create the same set of MPs as those defined above by copying and modifying the arguments to `SCA_template_ms`:

```

# multi-stock assessment
MP_1_4_ms <- SCA_template_ms

MP_2_4_ms <- SCA_template_ms
formals(MP_2_4_ms)$sigma <- 1

MP_3_4_ms <- SCA_template_ms
formals(MP_3_4_ms)$sigma <- 2

# constant buffer in years between assessments
MP_4_4_ms <- SCA_template_ms
formals(MP_4_4_ms)$sigma <- 2
formals(MP_4_4_ms)$pstar <- 0.4
formals(MP_4_4_ms)$incbuffer <- FALSE

# 10-year interval
MP_1_10_ms <- SCA_template_ms
formals(MP_1_10_ms)$interval <- 10

MP_2_10_ms <- SCA_template_ms
formals(MP_2_10_ms)$interval <- 10
formals(MP_2_10_ms)$sigma <- 1

```



```

MP_3_10_ms <- SCA_template_ms
formals(MP_3_10_ms)$interval <- 10
formals(MP_3_10_ms)$sigma <-2

# constant buffer in years between assessments
MP_4_10_ms <- SCA_template_ms
formals(MP_4_10_ms)$interval <- 10
formals(MP_4_10_ms)$sigma <- 2
formals(MP_4_10_ms)$pstar <-0.4
formals(MP_4_10_ms)$incbuffer <- FALSE

# assign all functions to class `MMP`
class(MP_1_4_ms) <- class(MP_2_4_ms) <- class(MP_3_4_ms) <- class(MP_4_4_ms) <-
  class(MP_1_10_ms) <- class(MP_2_10_ms) <- class(MP_3_10_ms) <-
  class(MP_4_10_ms) <- 'MMP'

```

## 4 Closed-Loop Simulation Testing

Now that the multi-stock operating model has been created and we've defined a set of MPs, we can simulate the historical fishery and run the projections to evaluate the performance of the alternative uncertainty buffers.

### 4.1 Running the Closed-Loop Simulation

First we define the names of the 16 MPs we wish to test (8 MPs assuming a combined single stock (cs) and 8 MPs with stock-specific advice (ms) and 4 and 10 year assessment intervals):

```

MPs <- c('MP_1_4_cs', 'MP_2_4_cs', 'MP_3_4_cs', 'MP_4_4_cs',
        'MP_1_4_ms', 'MP_2_4_ms', 'MP_3_4_ms', 'MP_4_4_ms',
        'MP_1_10_cs', 'MP_2_10_cs', 'MP_3_10_cs', 'MP_4_10_cs',
        'MP_1_10_ms', 'MP_2_10_ms', 'MP_3_10_ms', 'MP_4_10_ms')

```

And then use the `multiMSE` function to simulate the historical fishery and run the forward projections:

```

BD_MMSE <- multiMSE(BD_MOM, MPs=MPs, parallel = FALSE, checkMPs=FALSE)

```

And save the MMSE object to disk:

```

saveRDS(BD_MMSE, 'BD_MMSE.rda')

```

Note that `multiMSE` has option to run using parallel processing (`parallel=TRUE`) but our experience has shown that it is often less memory intensive and quicker to run the multi-stock MSE using a single core rather than distributing over multiple cores in parallel.

It also possible to split the simulations between the historical spool-up period and the forward projections

using the `SimulateMOM` and `ProjectMOM` functions. This can be useful if the simulations for the spool-up period are time-consuming.

For example:

```
# Simulate the historical spool-up period
Hist <- SimulateMOM(BD_MOM)

# Save historical simulation object to disc (can be quite large)
saveRDS(Hist, 'BD_multihist.rda')

# Forward projections with different sets of MPs
MMSE1 <- ProjectMOM(Hist, MPs=MPlist1)
MMSE2 <- ProjectMOM(Hist, MPs=MPlist2)
```

## 4.2 Examining the MMSE Object

The performance of the alternative stock assessment methods, harvest control rules, and uncertainty buffers can be evaluated by examining the `MMSE` object returned by `ProjectMOM`.

Help information on the `MMSE` object is available in the usual way:

```
class?MMSE
```

The information most relevant for evaluating the performance of the MPs is stored in the `SSB`, `SB_SBMSY`, `F_FMSY`, and `Catch` slots.

For example, the spawning stock biomass ( $SB$ ) relative to  $SB_{MSY}$  for each stock in the complex is available in the `SB_SBMSY` slot:

```
dim(BD_MMSE@SB_SBMSY) # nsim, nstocks, nMPs, proyears

## [1] 50  2 16 50
```

Likewise, the fishing mortality relative to  $F_{MSY}$  is in the `F_FMSY` slot, for each stock and fleet in the complex:

```
dim(BD_MMSE@F_FMSY) # nsim, nstocks, nfleets, nMPs, proyears

## [1] 50  2  1 16 50
```

Catches (landings) and removals are stored in the same way for each stock and fleet:

```
dim(BD_MMSE@Catch) # nsim, nstocks, nfleets, nMPs, proyears

## [1] 50  2  1 16 50

dim(BD_MMSE@Removals) # nsim, nstocks, nfleets, nMPs, proyears

## [1] 50  2  1 16 50
```

Finally, the biological reference points are available in the `RefPoint` slot, which contains reference points calculated for each year (`ByYear`) and dynamic unfished reference points (`Dynamic_Unfished`).

For example:

```
# SSB_MSY - by simulation, stock, MP, and historical and projection year
dim(BD_MMSE@RefPoint$ByYear$SSBMSY)

## [1] 50 2 16 167
```

### 4.3 Calculating Performance of the MPs

Performance can be calculated in many different ways. The information in the `MMSE` object can be used to calculate performance metrics for the individual stocks within the complex, or combined across all stocks to calculate performance metrics for the stock complex as a whole. Here we provide some examples of calculating mean  $SB/SB_0$  and mean yield, as well as the probability that spawning biomass is above  $0.25SB_0$ .

Although the following code is specific to the MPs used in this example case study, the approach can be generalized to calculate any performance metrics from an `MMSE` object.

We use the `dplyr` and `ggplot2` libraries:

```
library(dplyr)
library(ggplot2)
```

First, we create a function to extract the relevant details from the MP names, and create a data frame with the uncertainty buffer (in the first year after assessment), the assessment interval, and the assessment type (combined stocks assessed together and by-stock assessments):

```
MP_details <- function(MP) {
  tt <- get(MP)
  buffer <- get_buffer(1, formals(tt)$sigma, formals(tt)$pstar)[1]
  interval <- strsplit(MP, "_")[[1]][3] %>% as.numeric()
  type <- strsplit(MP, "_")[[1]][4]
  if (type == 'cs') type <- 'Combined'
  if (type == 'ms') type <- 'By-Stock'
  data.frame(MP=MP, Buffer=buffer, Interval=interval, Type=type)
}

df_mp <- lapply(BD_MMSE@MPs[[1]], MP_details) %>% do.call('rbind', .)
head(df_mp)
```

```
##      MP Buffer Interval  Type
## 1 MP_1_4_cs 0.939      4 Combined
## 2 MP_2_4_cs 0.882      4 Combined
## 3 MP_3_4_cs 0.778      4 Combined
## 4 MP_4_4_cs 0.602      4 Combined
## 5 MP_1_4_ms 0.939      4 By-Stock
## 6 MP_2_4_ms 0.882      4 By-Stock
```

Next, we create a function to extract the relevant information from the MMSE object and return the information in a data frame. In this case we are interested in the relative spawning biomass ( $SB/SB_0$ ) and the yield by simulation, MP, and year, for each stock in the multispecies model, and for the combined stocks (summing the stocks together):

```
extract_data <- function(MMSE) {
  nyears <- MMSE@nyears
  proyears <- MMSE@proyears
  ns <- length(MMSE@Stocks)
  nsim <- MMSE@nsim
  nMPs <- MMSE@nMPs
  # projection years
  p_yrs <- (nyears+1):(nyears+proyears)
  # Calculate SB/SBO and Yield for each stock
  SBO_C <- SB_C <- relSB <- Yield <- list()
  for (st in 1:ns) {
    # unfished spawning biomass by sim and year for st
    SBO_C[[st]] <- MMSE@RefPoint$ByYear$SSBO[,st,,p_yrs]
    # spawning biomass by sim and year for st
    SB_C[[st]] <- MMSE@SSB[,st,,]
    # relative spawning biomass
    relSB[[st]] <- SB_C[[st]]/SBO_C[[st]]
    # yield relative to MSY by sim and year for st
    Yield[[st]] <- MMSE@Removals[,st,1,,]/MMSE@RefPoint$ByYear$MSY[,st,,p_yrs]
  }
  df_list <- list()
  for (st in 1:ns) {
    df_list[[st]] <- data.frame(sim=1:nsim,
                                MP=rep(MMSE@MPs[[1]], each=nsim),
                                Year=rep(1:proyears, each=nsim*nMPs),
                                relSB=as.vector(relSB[[st]]),
                                Yield=as.vector(Yield[[st]]),
                                Stock=MMSE@Stocks[[st]]@Name)
  }
  # Calculate SB/SBO and Yield for combined stocks
  SBO_C <- Reduce(`+`, SBO_C)
  SB_C <- Reduce(`+`, SB_C)
  relSB[[ns+1]] <- SB_C/SBO_C
  Yield[[ns+1]] <- Reduce(`+`, Yield)
  df_list[[ns+1]] <- data.frame(sim=1:nsim,
                                MP=rep(MMSE@MPs[[1]], each=nsim),
                                Year=rep(1:proyears, each=nsim*nMPs),
                                relSB=as.vector(relSB[[ns+1]]),
                                Yield=as.vector(Yield[[ns+1]]),
```

```

      Stock='Combined Stocks')

df_all <- do.call('rbind', df_list)
df_mp <- lapply(MMSE@MPs[[1]], MP_details) %>% do.call('rbind', .)
DF <- left_join(df_mp, df_all, by='MP')
stock_names <- DF$Stock %>% unique()
type_names <- DF$Type %>% unique()
DF$Stock <- factor(DF$Stock, levels=stock_names,
                  ordered = TRUE)
DF$Type <- factor(DF$Type, levels=type_names,
                 ordered = TRUE)

DF
}

DF <- extract_data(BD_MMSE)
head(DF)

```

```

##           MP Buffer Interval      Type sim Year      relSB      Yield Stock
## 1 MP_1_4_cs  0.939          4 Combined   1    1 0.2899069 0.59846557 Blue
## 2 MP_1_4_cs  0.939          4 Combined   2    1 0.3993369 0.41769571 Blue
## 3 MP_1_4_cs  0.939          4 Combined   3    1 0.2835697 0.77970676 Blue
## 4 MP_1_4_cs  0.939          4 Combined   4    1 0.3098269 0.06480535 Blue
## 5 MP_1_4_cs  0.939          4 Combined   5    1 0.2818066 0.07132160 Blue
## 6 MP_1_4_cs  0.939          4 Combined   6    1 0.3348874 0.78705116 Blue

```

Note that our models are scaleless (we used an arbitrary R0) so the yield metrics are calculated relative to MSY.

Now that we have the relevant information in a data frame, we can create plots to summarize the data.

#### 4.3.1 Mean Relative Spawning Biomass

```

DF_mean <- DF %>% group_by(MP, Year, Stock, Buffer, Interval, Type) %>%
  summarize(MeanSB=mean(relSB), MeanY=mean(Yield), .groups = 'keep')

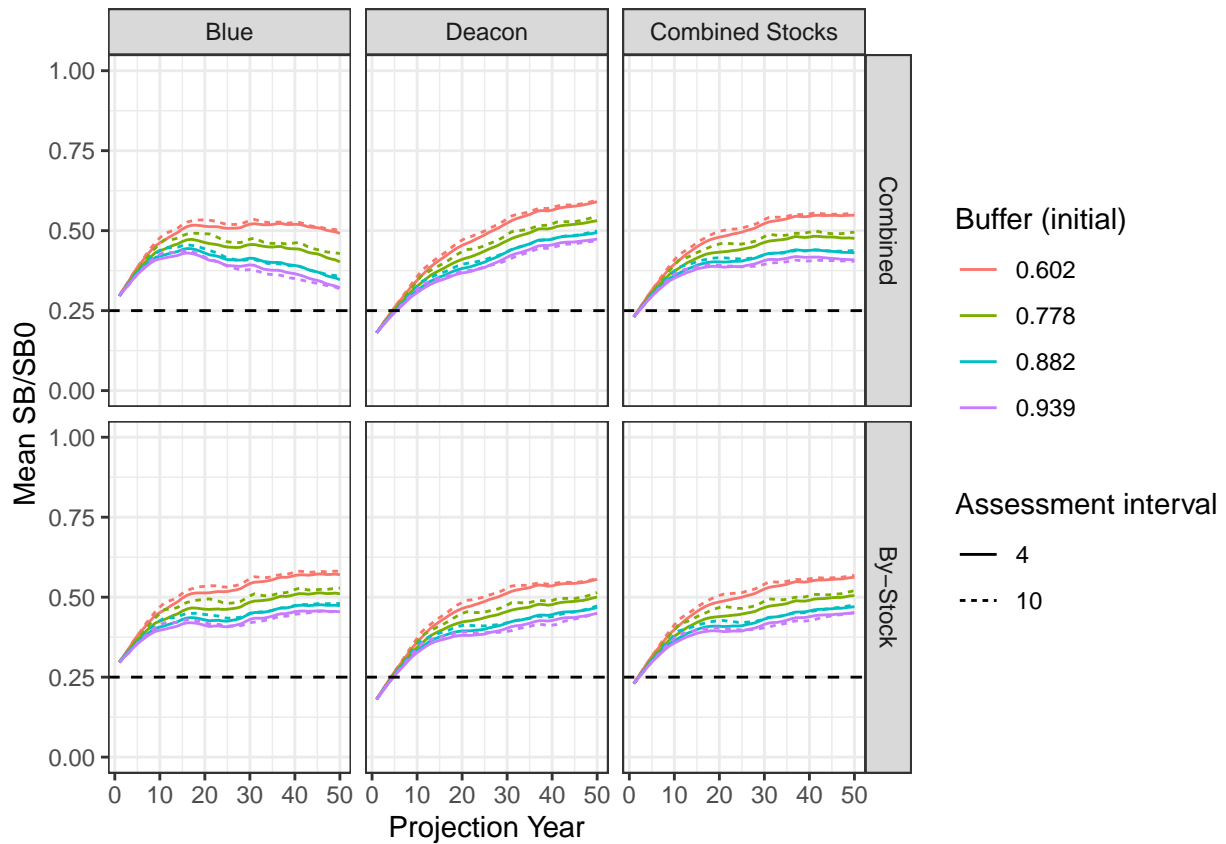
ggplot(DF_mean,
       aes(x=Year, y=MeanSB, color=as.factor(Buffer),
           linetype=as.factor(Interval))) +
  facet_grid(Type~Stock) +
  geom_line() +
  theme_bw() +
  expand_limits(y=c(0,1)) +
  geom_hline(yintercept = 0.25, linetype=2) +
  labs(x='Projection Year',

```

```

y='Mean SB/SB0',
color='Buffer (initial)',
linetype='Assessment interval')

```

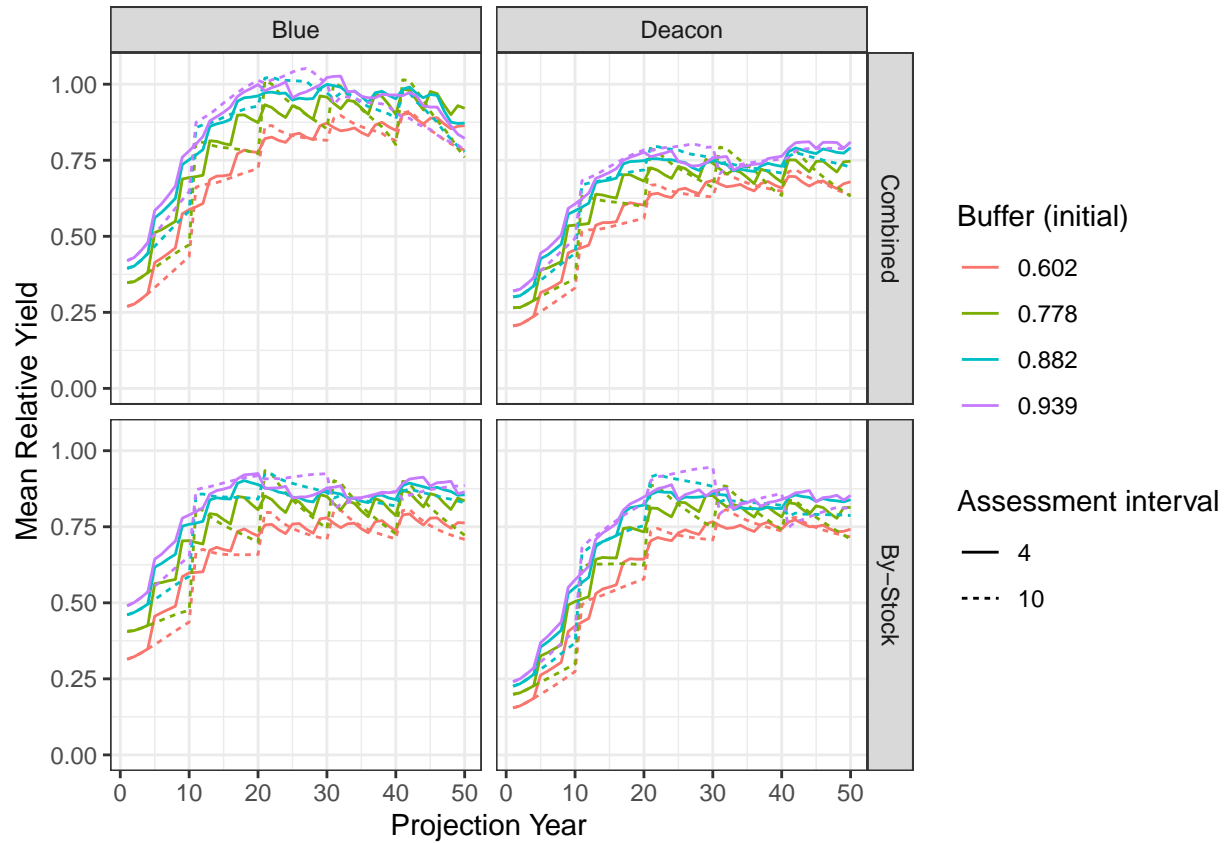


#### 4.3.2 Mean Relative Yield for each Stock

```

ggplot(DF_mean %>% filter(Stock != 'Combined Stocks'),
  aes(x=Year, y=MeanY, color=as.factor(Buffer),
    linetype=as.factor(Interval))) +
  facet_grid(Type~Stock) +
  geom_line() +
  theme_bw() +
  expand_limits(y=c(0,1)) +
  labs(x='Projection Year',
    y='Mean Relative Yield',
    color='Buffer (initial)',
    linetype='Assessment interval')

```

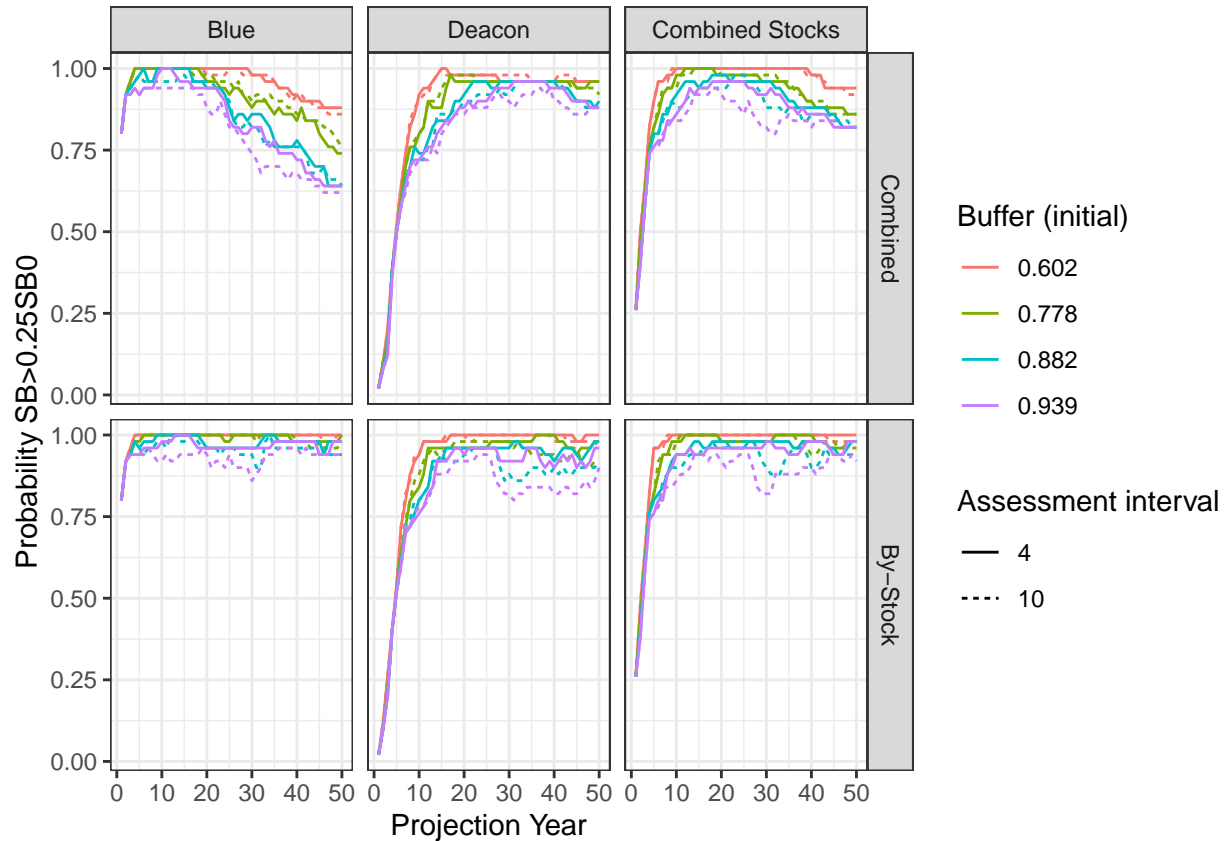


### 4.3.3 Probability $SB > 0.25SB_0$ in each year

```
DF_prob <- DF %>% group_by(MP, Year, Stock, Buffer, Interval, Type) %>%
  summarize(Prob=mean(relSB>0.25), .groups = 'keep')

ggplot(DF_prob, aes(x=Year, y=Prob, color=as.factor(Buffer),
                    linetype=as.factor(Interval))) +
  facet_grid(Type~Stock) +

  geom_line() +
  theme_bw() +
  labs(x='Projection Year',
       y='Probability  $SB > 0.25SB_0$ ',
       color='Buffer (initial)',
       linetype='Assessment interval')
```



#### 4.3.4 Trade-off plots of Prob $SB > 0.25SB_0$ and Yield in last 10 years

```

yrs <- 41:50
DF_prob2 <- DF %>% group_by(MP, Stock, Buffer, Interval, Type) %>%
  filter(Year %in% yrs) %>%
  summarize(Prob=mean(relSB>0.25),
            meanY=mean(Yield),
            lowY=quantile(Yield,0.1),
            highY=quantile(Yield,0.9),
            .groups = 'keep')

ggplot(DF_prob2 %>% filter(Stock != 'Combined Stocks'),
       aes(x=Prob, y=meanY, color=as.factor(Buffer),
           shape=as.factor(Interval))) +
  facet_grid(Type~Stock) +
  geom_point(size=1.5) +
  geom_errorbar(aes(ymin=lowY, ymax=highY)) +
  xlim(0.5,1) +
  theme_bw() +
  labs(x='Prob. SB>0.25SB0',

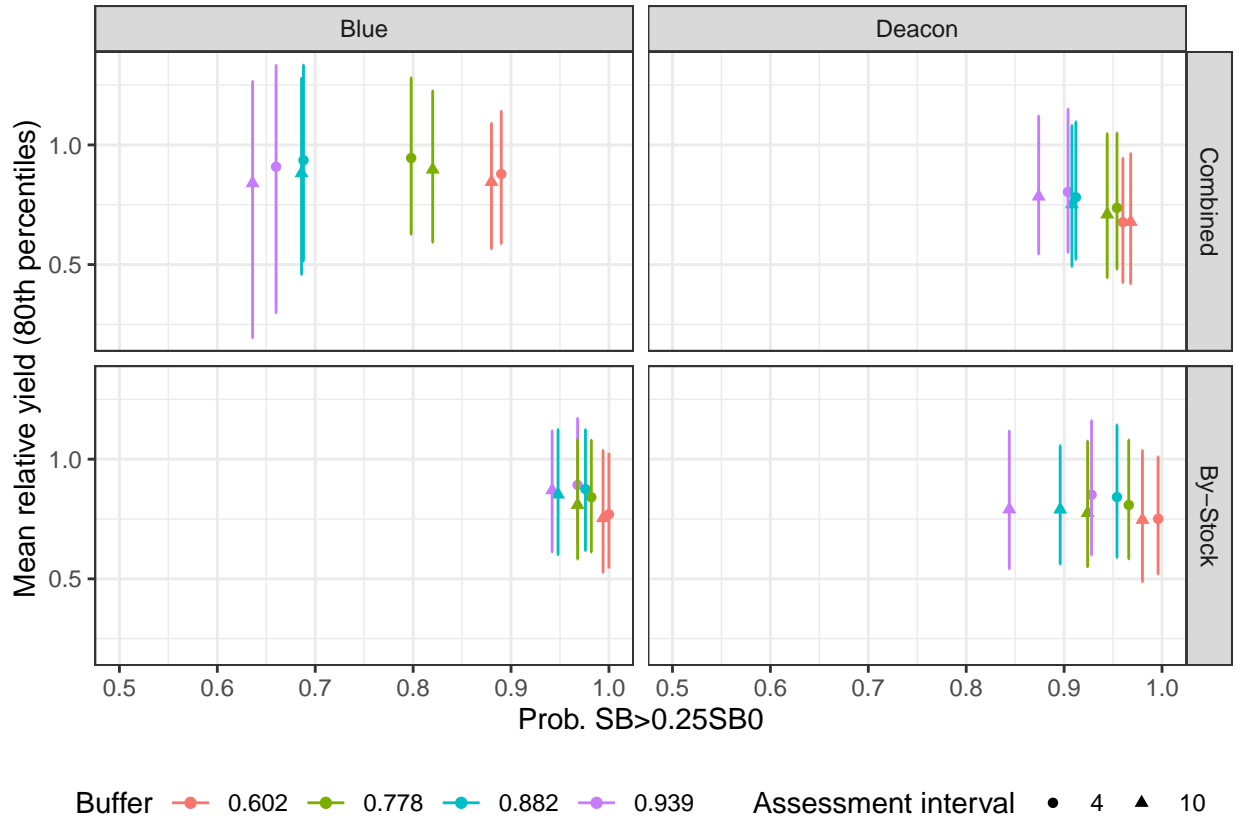
```



```

y='Mean relative yield (80th percentiles)',
color="Buffer",
shape='Assessment interval') +
theme(legend.position = 'bottom')

```



## 5 Acknowledgments

We thank the Resources Legacy Fund (RLF) for funding support for this work, and E.J Dick and John Field for helpful comments and feedback throughout the project.

## References

- Anon. (2019). *Scientific and statistical committee report on new methodology informing sigma values* (Supplemental SSC Report 1 Agenda Item G.3.a). Pacific Fishery Management Council. <https://www.pcouncil.org/documents/2019/03/agenda-item-g-3-a-supplemental-ssc-report-1.pdf/>
- Bickford, D., Lohman, D. J., Sodhi, N. S., Ng, P. K. L., Meier, R., Winker, K., Ingram, K. K., & Das, I. (2007). Cryptic species as a window on diversity and conservation. *Trends in Ecology & Evolution*, 22(3), 148–155. <https://doi.org/10.1016/j.tree.2006.11.004>

- Bizzarro, J. J., Gilbert-Horvath, E. A., Dick, E. J., Berger, A. M., Schmidt, K. T., Pearson, D., Petersen, C., Kautzi, L. A., Miller, R. R., Field, J. C., & Garza, J. C. (2020). Genetic identification of blue rockfish (*Sebastes mystinus*) and deacon rockfish (*S. diaconus*) to enable life history analyses for stock assessment. *Fishery Bulletin*, 118(1), 37–55. <https://spo.nmfs.noaa.gov/content/fishery-bulletin/genetic-identification-blue-rockfish-sebastes-mystinus-and-deacon-rockfish>
- DeMartini, E. E. (2019). Hazards of managing disparate species as a pooled complex: A general problem illustrated by two contrasting examples from Hawaii. *Fish and Fisheries*, 20(6), 1246–1259. <https://doi.org/10.1111/faf.12404>
- Dick, E. J., Berger, A., Bizzarro, J. J., Bosley, K., Cope, J., Field, J., Gilbert-Horvath, L., Grunloh, N., Ivens-Duran, M., Miller, R., Privitera-Johnson, K., & Rodomsky, B.T. (2017). *The Combined Status of Blue and Deacon Rockfishes in U.S. Waters off California and Oregon in 2017*. Pacific Fishery Management Council, Portland, OR. <http://www.pcouncil.org/groundfish/stock-assessments/>
- Dick, E. J., & MacCall, A. D. (2010). *Estimates of sustainable yield for 50 data-poor stocks in the Pacific Coast Groundfish Fishery Management Plan*. National Oceanic and Atmospheric Administration. <https://repository.library.noaa.gov/view/noaa/3737>
- Frable, B. W., Wagman, D. W., Frierson, T. N., Aguilar, A., & Sidlauskas, B. L. (2015). A new species of *Sebastes* (Scorpaeniformes: Sebastidae) from the northeastern Pacific, with a redescription of the blue rockfish, *S. mystinus* (Jordan and Gilbert, 1881). *Fishery Bulletin*, 113(4), 355–377. <https://doi.org/10.7755/FB.113.4.1>
- Hicks, A. C., Wetzel, C., & Harms, J. (2014). *The status of rougheye rockfish (*Sebastes aleutianus*) and blackspotted rockfish (*S. melanostictus*) as a complex along the U.S. West Coast in 2013* (p. 269). Northwest Fisheries Science Center. <https://www.pcouncil.org/documents/2014/01/the-status-of-rougheye-rockfish-sebastes-aleutianus-and-blackspotted-rockfish-s-melanostictus-as-a-complex-along-the-u-s-west-coast-in-2013-published-january-6-2014.pdf/>