```
<<InitialStuff, echo=FALSE, results=hide>>=
library(ggplot2)
library(FLCore)

data(nsher)

options(width = 80)
@
```

Using `FLBRP`

Laurence Kell

19. January 2011

Table of Contents

Introduction

Biological reference points are important elements in scientific advice, where they are commonly used to specify targets and limits. For example maximum sustainable yield (MSY) provides a target for the exploitation rate or fishing mortality (F_{MSY}) and for the associated biomass (B_{MSY}). Due to uncertainty it is also often necessary to set limits since a fishing mortality of F_{MSY} may in some cases actually drive the stock below $B_{MSY.}$ The limit could be a biomass level below B_{MSY} which triggers a rebuilding plan if the stock falls below it.

Reference points may be empirically or model based. Model based reference points may either be derived directly from a stock assessment model, as in the case of a biomass dynamic model where B_{MSY} is a function of carrying capacity (K) and r, or calculated based upon model outputs. FLBRP is intended to be used in the latter case where an age based model is used as the input.

Proxies for MSY such as $F_{0.1}$, are commonly derived from equilibrium values, which in the case of F_{MSY} and may include a stock-recruitment relationship. The approach is based upon an age-structured equilibrium model that combined SSB- and yield-per-recruit, and stock–recruitment analyses partial fishing mortality-at-age (F_a), natural mortality-at-age (M_a), proportion mature-at-age (Q_a), and weight-at-age (W_a) data, with a stock–recruitment relationship (SRR), where a is age, n the plus group, and r the age-at-recruitment. Spawners-per-recruit (S/R) is given as a function of F by

$$\text{S/R} = \sum_{a=r}^{n-1} \mathrm{e}^{-\sum_{i=r}^{a-1} F_i+M_i} W_a Q_a + \mathrm{e}^{-\sum_{i=r}^{n-1} F_i+M_i} \frac{W_n Q_n}{1-\mathrm{e}^{-F_n+M_n}} , \tag{1}$$

where the second term is the plus group, i.e. a summation of all ages from the last age to infinity. Yield is derived from the yield-per-recruit ratio (Y/R):

$$\text{Y/R} = \sum_{a=r}^{n-1} \mathrm{e}^{-\sum_{i=r}^{a-1} F_i+M_i} W_i \frac{F_i}{F_i+M_i}(1-\mathrm{e}^{-F_i-M_i}) + \mathrm{e}^{-\sum_{i=r}^{n-1} F_i+M_i} W_n \frac{F_n}{F_n+M_n} \tag{2}$$

Spawning-stock biomass (S) and yield (Y) can be derived from the spawner-per-recruit ratio by re-arranging the stock–recruitment model so that recruitment is a function of S/R (Sissenwine and Shepherd, 1987). A compensatory (Beverton and Holt, 1957) stock–recruitment relationship was fitted. Recruitment, R, can be derived from the spawner-per-recruit ratio, i.e. for Beverton and Holt:

$$R = \alpha - \frac{\beta}{\text{S/R}} , \tag{3}$$

and for Ricker:

$$R = \ln(\alpha \text{S/R}) \frac{\beta}{\text{S/R}} . \tag{4}$$

`FLBRP` **Class**

Slots

The FLBRP class holds slots corresponding to data and assumptions about biological parameters, selection patterns and economic quantities. These slots are time invariant and so only have length of 1 along the second year dimension of FLQuant. There are also slots for specifying the stock recruitment relationship, recoding time series of historic observations and saving the estimates of equilibrium values for different effort or F levels. These slots can be inspected, changed or manipulated.

.General Information

name	character: i.e. the stock that the data came from
desc	character: a description that helps in remembering where the data came from
range	numeric: the range of years and ages in the data, can be used to truncate results of methods without changing the data

3 Biological Parameters

The following slots are average values and can be derived (e.g. from "FLStock") from other classes

Weights-at-age

stock.wt	FLQuant: stock
landings.wt	FLQuant: landings
discards.wt	FLQuant: discards
bycatch.wt	FLQuant: byactch

catch.wt is not an actual slot but is calculated as a weighted sum of landings and discards weights.

m	FLQuant: Natural mortality
mat	FLQuant: Proportion mature-at-age
availability	FLQuant: proportion of stock by area
harvest.spwn	FLQuant: proproption of fishing mortality before spawning
m.spwn	FLQuant: proproption of natural mortality before spawning

Stock Recruitment relationship

The stock recruitment relationship is specified in two slots

model	FLFormula specifying the functional form
params	FLPar specifying the parameters

these can be accessed (i.e. inspected or modified) directly or set when creating object from another FLR object such as `FLSR` or passed to an method such as `as.FLStock()` .

Fishery parameters

Selection pattern is specified for landings and discards

landings.sel	FLQuant: selection pattern at age of lanhded numbers, scaled by Fbar range
discards.sel	FLQuant: selection pattern at age of discarded numbers, scaled by Fbar range

catch.sel is not an actual slot but is calculated as the sum of landings and discards selection patterns.

bycatch mortality	FLQuant: mortality-at-age due to fishing by other than target fisheries
fbar	FLQuant: mean equilibrium fishing mortlaity

Economic parameters

price	FLQuant: price per unit wieght-at-age
fcost	FLQuant: fixed costs
vcost	FLQuant: variable costs

Historical time series

Historic observations slots, these are the observed values, e.g. derived from the "FLStock" object used to creat "FLBRP" object.

fbar.obs	FLQuant time series of historic mean fishing mortality
.obs	FLQuant time series of historic landing biomass
.obs	FLQuant time series of historic discard biomass
.obs	FLQuant time series of recruitment
.obs	FLQuant time series of spawning stock biomass
.obs	FLQuant time series of profit

catch.obs: is not an actual slot but is calculated as the sum of landings and discards, while yield.obs: synonymous with landings.obs

Reference points

refpts	refpts: biological and economic reference points, that for each refer-

	ence point contains the corresponding values of harvest, yield, rec, ssb, biomass, revenue, cost and profit.

By default the main reference points $F_{0.1}$, F_{MAX}, SPR30%, MSY and MEY are estimated.

Methods

As well as slots the class contains methods that derive quantities of interest from the slots, these might require operations on several slots, for example ssb() will calculate the SSB from the sum of the product of number, mass and maturity-at-age with a year/season, while catch will return the sum of discards and landings.

The FLQuant returned by these methods are time invariant and so only have length of 1 along the second year dimension.

yield	FLQuant equilibrium estimate of yield for given effort
catch	FLQuant equilibrium estimate of yield for given effort
landings	FLQuant equilibrium estimate of yield for given effort
discards	FLQuant equilibrium estimate of yield for given effort

stock.n:	FLQuant equilibrium estimate of catch numbers at age for given effort
landings.n:	FLQuant mean selection pattern
discards.n:	FLQuant equilibrium estimate of fishing mortality-at-age for given effort

rec	FLQuant equilibrium estimate of recruitment for given effort
ssb	FLQuant equilibrium estimate of spawning stock biomass for given effort
stock	FLQuant equilibrium estimate of biomass for given effort

revenue	FLQuant equilibrium estimate of revenue for given effort
costs	FLQuant equilibrium estimate of costs for given effort
profit	FLQuant equilibrium estimate of profi for given effort

spr	spawner per recruit e.g. ssb/rec
ypr	yield per recruit yield/rec
spr0	SSB per recruit at F=0

FLBRP:	creates new FLBPR object

is.FLBRP:	checks object of type FLBRP
computeRefpts	computes reference points and return array
brp	creates runs both equilibrium and computeRefpts
setPlusGroup	creates new plusgroup whose age is greater than current plus group

Examples

Creating an object

The simplest way to create a new instance, i.e. object of a class, is from an existing object of an appropriate class, e.g. from an FLStock.

```
#### Creating an new FLBRP object from an FLStock

## load package
library(FLBRP)

## Use example FLStock object from FLCore
data(ple4)

## Create the corresponding FLBRP object
pleBrp<-FLBRP(ple4)

## Inspect the object using summary to get a quick overview
summary(t.)
```

Chunk 1:

“Bespoke” objects can be created using the FLBRP creator.

```
FLBRP(object, sr, model=formula(rec~a), params=FLPar(1,
params='a'),fbar=seq(0, 4, 0.04), nyears=3, biol.nyears=nyears,
fbar.nyears=nyears, sel.nyears=fbar.nyears, na.rm=TRUE,
mean='arithmetic', ...)
```

This automatically creates an FLBRP object from an FLStock, there are also various arguments that allow the slots to be calculated.

```
#### Creation from an FLStock II
catch.sel(pleBrp)

pleBrp<-FLBRP(ple4, nyrs = 10)

catch.sel(pleBrp)
```

Chunk 2:

The slots are now all created and filled with data

```
#### Creation from an FLStock III

getSlots("FLBRP")

## Selection Patterns
catch.sel(pleBrp)
xyplot(data~age,data=catch.sel(pleBrp),type="l")

discards.sel( pleBrp)
landings.sel( pleBrp)
bycatch.harvest(pleBrp)

## Mass-at-age
stock.wt( pleBrp)
catch.wt( pleBrp)
discards.wt(pleBrp)
bycatch.wt( pleBrp)

xyplot(data~age,groups=qname,data=FLQuants(swt=stock.wt(pleBrp),cwt=catch.wt
(pleBrp)),type="l")

## Biological parameters
m( pleBrp)
mat(pleBrp)

xyplot(data~age|
qname,data=FLQuants(sel=catch.sel(pleBrp),dsel=discards.sel(pleBrp),
                    swt=stock.wt(pleBrp), cwt =catch.wt(pleBrp),
                    mat=mat(     pleBrp), m   =m(       pleBrp))
,type="l",scale="free")
```

Chunk 3:

Although no catch.wt slot exists, catch.wt(pleBrp) return the right data; this is because catch.wt is the weight sum i.e. sum of numbers*weight)/(sum of number), of landings and discards and so can be calculated from other slots. This shows the benefit of the design of object orientated programming. The delatils of implementation are hidden from the user who has a common interface to quanties of interest. Making catch.wt a method, so that the values are calculated from other values, saves memory required for storage and also ensures that all quantities are consistent with each other.

There are also slots corresponding to the historic observations, it may also be used to extract objects from an FLBRP object for other analysis.

```
## historic observations
fbar.obs(    pleBrp)
yield.obs(   pleBrp)
landings.obs(pleBrp)
discards.obs(pleBrp)
rec.obs(     pleBrp)
ssb.obs(     pleBrp)
profit.obs(  pleBrp)
```

Chunk 4:

The FLBRP object provides a coherent structure for storing data and analysing it, but to take advantage of other packages and to do one off analyses often it can be important to extract parts of it into other format. A useful data type in R is the data.frame, this is like a database or SAS data set where the various variable are stored by column and the observations by row. First we create a list of FlQuant objects (i.e. FLQuants) and then corece it into a data.frame. By default the data.frame has a column idicating what the data represent, although other formats may be more appropriate depending on what is the intention of the analyst, for example the model.frame function creates a data.frame where the data are in columns.

```
obs<-FLQuants(fbar    =fbar.obs(    pleBrp),
              yield   =yield.obs(   pleBrp),
              landings=landings.obs(pleBrp),
              discards=discards.obs(pleBrp),
              rec     =rec.obs(     pleBrp),
              ssb     =ssb.obs(     pleBrp))

## coercion into a data.frame
head(obs.df<-as.data.frame(obs))

ggplot(obs.df)+geom_line(aes(year,data))+facet_wrap(~qname,scale="free")

## coercion into a model.frame
head(obs.mf<-model.frame(obs))

cor(obs.mf[,c("fbar","yield","landings","discards","rec","ssb")])
```

Chunk 5:

as well as extracting parts of an object, you can also extract all the slots at once, and even export it from R

v

```
brp.df<-as.data.frame(pleBrp)
head(brp.df)


dim(brp.df)
write.table(brp.df,save= "c:/temp/pleBRP.txt")
```

Chunk 6:

The easiest way often to inspect an object is to plot it, plots are of various types, summaries of the equilibrium curves with the various reference points or advice plots based on the Kobe plots. For now we will just look at the basic plots and return in more detail later. As well as filing up the data slots, calculations can be done to estimate equilibrium values and reference points, this is done using either computeRefpts or brp. The later just calculates the reference points, the former will calculate both reference points and equilibrium values, it is useful therefore for updating an entire FLBRP object.

```
plot(pleBrp)

pleBrp<-brp(pleBrp)

plot(pleBrp)
```

Chunk 7:

Reference points

The main reference points estimated are the biological reference points $F_{0.1}$, F_{MAX}, SPR30%, MSY and the the economic reference point MEY. Although a variety of reference points can also be set up corresponding to values of harvest, yield, rec, ssb, biomass, revenue, cost and profit or ratios of these quantities. For example virgin biomass can be obtained from setting harvest to 0 and then calculating the corresponding level of biomass, while F_{crash} can be estimated by setting biomass to 0 and calculating the corresponding harvest value and the break-even point (where costs=revenue) by setting profit=0.

The slot 'fbar' has a slightly different structure. This is a vector containing the values of fbar that the equilibrium values are calculated for.

```
####
fbar(pleBrp)
```

Chunk 8:

Please ignore the fact that the second dimension is labelled 'year'. This is only due to the restrictions placed on labelling dimension of FLQuant objects. The default range of 'fbar' values is from 0 to 4 in steps of 0.04. A user defined range can be specified when the object is created e.g.

```
####
pleBrp2 <- FLBRP(ple4,fbar=seq(from = 0, to = 5, by = 0.5))
fbar(pleBrp2)
```

Chunk 9:

Fitting

Once an FLBRP object has been created then equilibrium quantities at age and time series can be estimated. We haven't specified a stock-recruitment relationship yet. The default is a mean of value of 1. Specifying alternative relationships is covered below.

Estimates of equilibrium values at each of the values of Fbar in the 'fbar' slot.

```
####
## Fishing mortality
harvest(pleBrp)

## check that it worked!
sweep(harvest(pleBrp),1,catch.sel(pleBrp),"/")
apply(harvest(pleBrp)[range(pleBrp)[["minfbar"]]:range(pleBrp)
[["maxfbar"]],],2,mean)

## other estimated quantities-at-age
stock.n(pleBrp)
landings.n(pleBrp)
discards.n(pleBrp)

yield(pleBrp)
rec(pleBrp)
ssb(pleBrp)
```

Chunk 10:

Reference Points

As well as calculating equilibrium quantities, you can also calculate reference points based upon these. For example, the yield and spawner per recruit reference points F0.1, Fmax.

There are two main fitting functions:

- computeRefpts - calculates reference points and returns them as an array
- brp - also calculates reference points and returns an FLBRP object with the reference points in.

```
#### estimate reference points
computeRefpts(pleBrp)

## inspect reference points
refpts(pleBrp)
```

Chunk 11:

Fmsy is the same as Fmax, since the default assumed stock recruitment relationship is mean recruitment. You can plot the reference points and expected quantities over the range of fbar values.

Defining reference points

There are three ways of defining reference points

- By name
- Specifying a value

- Specifying a ratio

By name

The reference points corresponding to "f0.1", "fmax", "msy" and "mey" are calculated by default.

Spawner per recruit ratios can also be calculated using the name in the form spr.xx, where xx coresponds to the percentage of virgin e.g. spr.30 would be 30% of virgin.

```
####
computeRefpts(pleBrp)

You can use others e.g. crash

refpts(pleBrp)<-refpts(as.numeric(NA),refpt=c("crash"))
computeRefpts(pleBrp)
```

Chunk 12:

Specifying a value

Alternatively a single value can be specified (all other ones=NA) and then the corresponding values for all other quantities will be calculated.

```
####
refpts(pleBrp)<-refpts(c(.5,NA,NA,NA,NA,NA,NA,NA),refpt=c("mybrp"))
# Check the set up
refpts(pleBrp)
# And calculate
computeRefpts(pleBrp)
```

Chunk 13:

Specifying a ratio

Values corresponding to spawner and yield per recruit ratios can be calculated if their ratios are specified (al other ones= NA).

e.g. SPR=0.5

```
#### First clean up the old refpts by setting to NA
refpts(pleBrp)[] <- NA
# Add your new one
refpts(pleBrp)[,c("ssb","rec")]<-c(.5,1)
refpts(pleBrp)
computeRefpts(pleBrp)
```

Chunk 14:

It can be useful to plot this against the historical data. First we will modify the stock-recruitment relationship in the FLBRP object so that the mean value is the mean of the historical recruitment.

The parameters of the stock-recruitment relationship are held in the 'sr.params' slot. This can be easily modified. For more information on how to specify alternative stock-recruitment relationships, see below.

```
#### Look at current value of the mean stock-recruitment relationship
pleBrp<-FLBRP(ple4)

#### Look at current value of the mean stock-recruitment
relationship
params(pleBrp)
# Now set the recruitment to the mean value
params(pleBrp)["a"]<-mean(pleBrp@rec.obs)
# Redo the brp estimation
pleBrp<-brp(pleBrp)
# And plot the results with the historical observations by
setting the obs flag to TRUE
plot(brp(pleBrp),obs=TRUE)
```

Chunk 15:

Stock Recruitment Relationships

It is possible to include alternative stock recruitment assumptions. Details of the stock-recruitment relationship are found in the 'sr.model' and 'sr.params' slots:

```
# The SR model
model(pleBrp)

# And the current parameter values
params(pleBrp)
```

Chunk 16:

The default stock-recruitment relationship is mean recruitment. It is straightforward to change the value as we saw above

```
####
params(pleBrp)[["a",1]] <-exp(mean(log(rec.obs(pleBrp))))
pleBrp <-brp(pleBrp)
plot(pleBrp)
```

Chunk 17:

It is sometimes preferable to use a fitted stock-recruitment relationship. First, take a look at the historical data.

```
#### ## Stock recruitment relationships
plot(ssb.obs(pleBrp),rec.obs(pleBrp),xlim=c(0,max(ssb.obs(pleBrp))),ylim=c(0
,max(rec.obs(pleBrp))))
lines(lowess(ssb.obs(pleBrp),rec.obs(pleBrp)))
```

Chunk 18:

Now fit a Ricker relationship using FLSR:

```
##### Stock recruitment relationships II
## Ricker stock recruitment relationship
pleSR       <-as.FLSR(ple4)
model(pleSR)<-ricker()

## fit
pleSR <-fmle(pleSR)

plot(pleSR)
```

Chunk 19:

Th FLSR object can then be passed directly into the FLBRP creator:

```
####
pleBrp.rk <-FLBRP(ple4,sr=pleSR)

# See how the SR slots have been set
model(pleBrp.rk)
params(pleBrp.rk)

# And re-estimate values
pleBrp.rk <-brp(pleBrp.rk)
plot(pleBrp.rk)
```

Chunk 20:

Compare reference points:

```
###### reference points
refpts(pleBrp)
refpts(pleBrp.rk)
```

Chunk 21:

Plot the surplus production curves:

```
## surplus production curves
plot( ssb(pleBrp)/1000,yield(pleBrp)/1000,type="l",ylim=c(0,150))
lines(ssb(pleBrp.rk)/1000, yield(pleBrp.rk)/1000,type="l",col="red")
```

Chunk 22:

Economic reference points

Requires prices and fixed and variable costs

```
####
price(pleBrp)[]<-c(1,1,2,2,rep(3,6))
fcost(pleBrp)[]<-200000
vcost(pleBrp)[]<-50000

refpts(pleBrp)<-refpts()
computeRefpts(pleBrp)
```

Chunk 23:

Monte Carlo Simulation

Here we see how the reference points can vary with uncertainty, for example, in the natural mortality.

```
####
data(ple4)
pleBrp<-FLBRP(ple4)

# Expand along the iteration dimension
pleBrp <-propagate(pleBrp,iter=100)
# Put some noise on the natural mortality
m(pleBrp) <-m(pleBrp)*rlnorm(prod(dim(m(pleBrp))),0,0.3)
# Calculate reference points
refpts(pleBrp)<-computeRefpts(pleBrp)
# Remove the economic refpts
refpts(pleBrp) <- refpts(pleBrp)[-5,]
# 'ave a butchers
refpts(pleBrp)

# Turn refpts into a dataframe (makes plotting easier)
data<-cbind(expand.grid(dimnames(refpts(pleBrp)
[,"harvest",])),val=c(refpts(pleBrp)[,"harvest",]))[,-2]

# Plots
## default
plot(refpts(pleBrp)[1:4,1:5])

## bespoke
histogram( ~ val | refpt,
data = data,
xlab = "Fishing Mortality", type = "density",
panel = function(x, ...) {
panel.histogram(x, ...)
panel.mathdensity(dmath = dnorm, col = "black",
args = list(mean=mean(x),sd=sd(x)))},
scale="free", xlim=c(0,.5))
```

Chunk 24:

More Plots

The main plot is of the equilibrium curves, showing the reference points and if desired the observed historic values. It is always good to predict the observed and predicted values, since if they are very different then something is wrong with the assumptions and should be checked. There are also default plots for the reference points.

```
kobe(pleBrp.rk)
```

Chunk 25:

Advice plots in the form of Kobe phase plots are also available, the Kobe plot has axes correspond to fishing mortality relative to F_{MSY} and SSB relative to B_{MSY} The green quadrant corresponds to the stock being above B_{MSY} and harvesting at less than $F_{MSY.}$ While the red quadrant to the stock being below B_{MSY} and harvesting at greater than F_{MSY} and the yellow quadrants to the stock being either below B_{MSY} or harvesting at greater than F_{MSY}. The black line corresponds to the median of historical estimates, the grey to the median of projected estimates and the points to individual realisations at the end of the projection period (2020), white correspond to the 50th bi-variate percentile. Individual panels correspond to the different TAC levels (20,000 tonnes starting in the top left hand panel then increasing TACs going left to right across columns).

The Kobe plots can then be used to show the probabilities of restoring the stock to be within management targets over time given the different management.

1 References

Pilling, G.M., Kell, Hutton, T., Bromley, P.J., Tidd A.N. and Bolle L.J., 2008. Can economic and biological management objectives be achieved by the use of MSY-based reference points? A North Sea plaice (Pleuronectes platessa) and sole (Solea solea) case study. ICES Journal of Marine Science: Journal du Conseil 2008 65(6):1069-1080.