Monkfish 27.8c9a: Short-Cut MSE approach for robustness tests of harvest control rules in sex-structured models

Henning Winker (GFCM)

08 March, 2025

Contents

1 Short-Cut MSE for Harvest Control Rule (HCR) 1
1.1 Glossary 2

2 Build FLStock 2
2.1 Plot SS3 Stock Dynamics 4
2.2 Consistency checks using backtesting 7
2.3 Estimate candidate reference points 7

3 Tuning grid 9

4 Set up short-cut MSE 13
4.1 Setting up harvest control rules 14

5 Performance Evaluation with adjustment for sex-tructured stocks 26
5.1 Define Metrics for performance evaluation of sex-structured 26
5.2 Long-term last 25 years (75 years) 26
5.3 PLOT performance 27

Load packages

```
# Load
library(ggplot2)
library(FLCore)
library(ggplotFL)
library(mse)
library(FLRef)
library(ggpubr)
library(mseviz)
```

1 Short-Cut MSE for Harvest Control Rule (HCR)

In contrast to a full Management Strategy Evaluation (MSE) simulation design (Punt et al. 2017), the MSE 'shortcut' approach, omits the step of the annual updating of the estimation model (assessment) in the feedback control. Instead, it emulates an annual update of the benchmark assessment model by passing outcomes (SSB and F) from the 'true' age-structured dynamics from the operating model (OM) with assessment error to the harvest control rule (HCR) and catch implementation system.

The HCRs were implemented using a simulated feedback control loop between the implementation system and the operating model, where the implementation system translates the emulated assessment outcome via the HRC into the Total Allowable Catch (TAC) advice. The feedback control loop between the implementation system and the OM allows accounting for the lag between the last of year data used in the assessment and the implementation year of catch advice (C_{adv}).

For blackspot seabream, the implementation system of the harvest control rule is based on the assumption that advice is given for year $y+1$ based on an assessment completed in year y, which is fitted to data up until last data year $y-1$. Therefore implementation of the derived C_{adv} through HCR requires projection of the stock dynamics by way of a short-term forecast. To do this, numbers-at-age were projected through the year of assessment. Status quo recruitment, M_a, w_a and mat_a were set as the mean of the last 3 years. A projection based on a fixed fishing mortality-at-age to the last year ($y-1$) in the assessment is then made through to the implementation year ($y+1$).

The limitations of the MSE short-cut approach are that it cannot fully account for uncertainties resulting from imperfect sampling of the full age-structure (e.g. poorly sampled recruits), observation error, misspecified model assumptions and selectivity. On the other hand, the short-cut MSE approach is straight-forward to implement (`FLR`) and reduced complexity and computation time when the focus is predominantly optimizing HCRs for setting quotas on the premises that a benchmark assessment form the basis for the advice.

Here, the MSE short-cut approach is implemented using the tools available in the Fisheries Library for R (FLR; Kell et al., 2007; https://flr-project.org/)

1.1 Glossary

The following glossary summarizes key HCR parameters and associated target and limit reference points that are considered for tuning the candidate HCRs to optimise the trade-offs between maximising fishing opportunity and risk:

- F_{MSY}: target reference point for fishing mortality at Fmsy (or its proxy), (e.g. F_{B35})
- B_{MSY}: the average biomass around which the biomass fluctuated when fishing at F_{MSY} or its proxy (e.g. B_{35})
- B_{lim}: a deterministic biomass limit reference point below which a stock is considered to have reduced reproductive capacity. Here B_{lim} was set to $0.25B_{tgt}$
- B_{pa}: a precautionary biomass reference point set with high probability that biomass is above B_{lim}, which acts as a safety margin below which the risk of reduced reproductive capacity is increasing. When the biomass is estimated to be above Bpa, the stock is considered to be wthin safe biological limits in terms of its reproductive capacity.
- C_{adv}: advised catch as output of the management procedure
- $B_{trigger}$: biomass trigger point of the HCR, specified as change point of biomass below which fishing mortality reduced relative to Ftgt. Btrigger is typically specified as ratio to B_{MSY}.

2 Build FLStock

SS3 outputs are loaded with the `readFLSss3()` into an `FLStock` object. The folder that contains the model outputs has to be specified.

In the following, the area outside is evaluated first.

```
Loading objects:
  stk
  sr
  out
```

```
run = "mon.27.8c9a"
stk = window(ss3om::readFLSss3(file.path("ss3mods",run),wtatage = TRUE))
```

```r
# Fill NAs
stk@m.spwn[] = 0
stk@harvest.spwn[] = 0
sr = ss3om::readFLSRss3(file.path("ss3mods",run))
stk@name = run
stk@desc = "2024, ICES, SS3"
out = ss3om::readOutputss3(file.path("ss3mods",run))
```

```r
range(stk)
```

```
        min       max plusgroup   minyear   maxyear   minfbar   maxfbar
          0        30        30      1980      2023         2        15
```

```r
plot(stk,metrics=list(SSB=function(x)unitSums(ssb(x)[,,,1]),
                      F=function(x)fbar(x),Catch=function(x)catch(x),Rec=function(x)unitSums(rec(x))))+
  theme_bw()+ylab("F")+xlab("Year")+facet_wrap(~qname,scales="free_y")
```

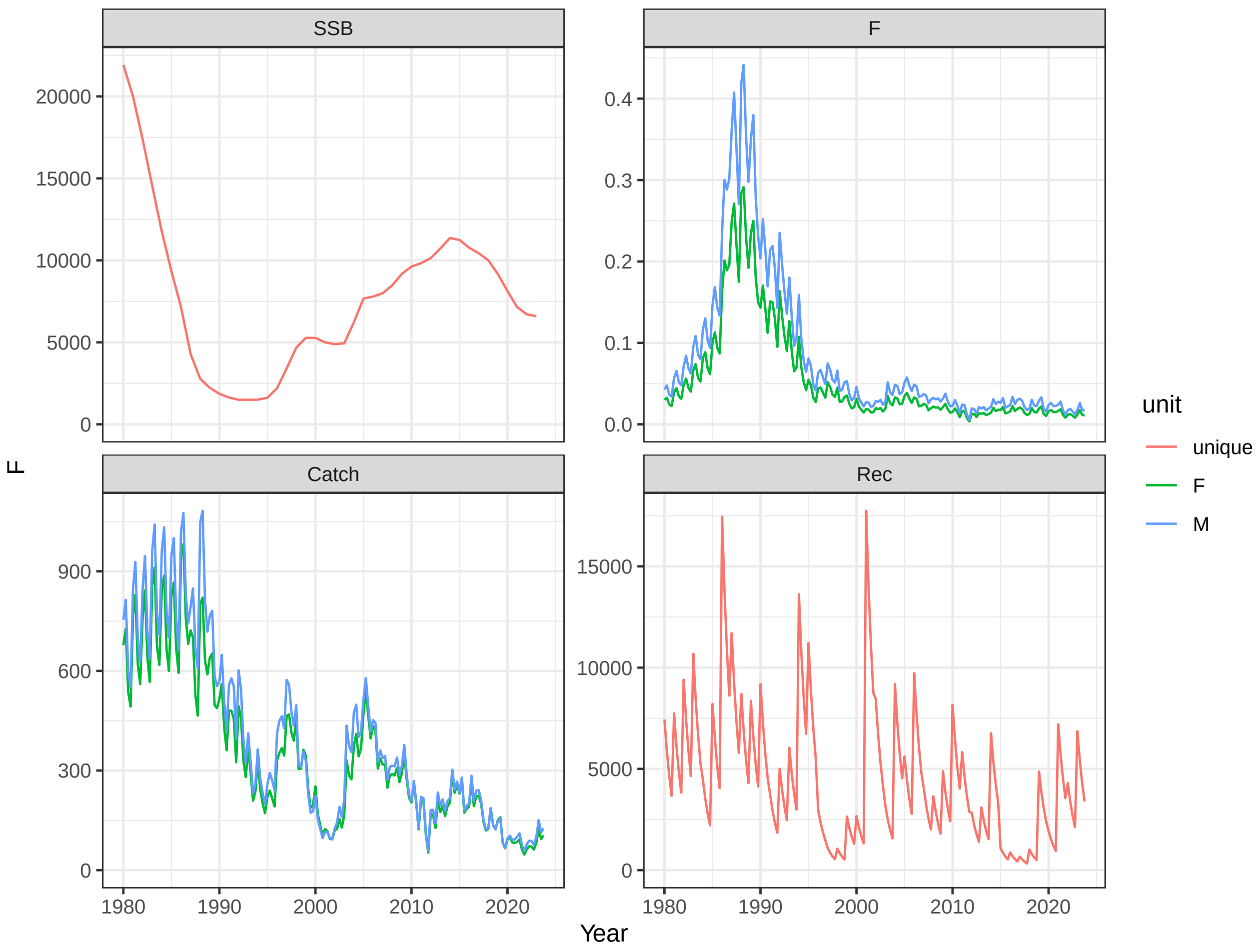

Figure 1: Seasonal stock trajectories

2.1 Plot SS3 Stock Dynamics

```
plotdyn(stk)
```

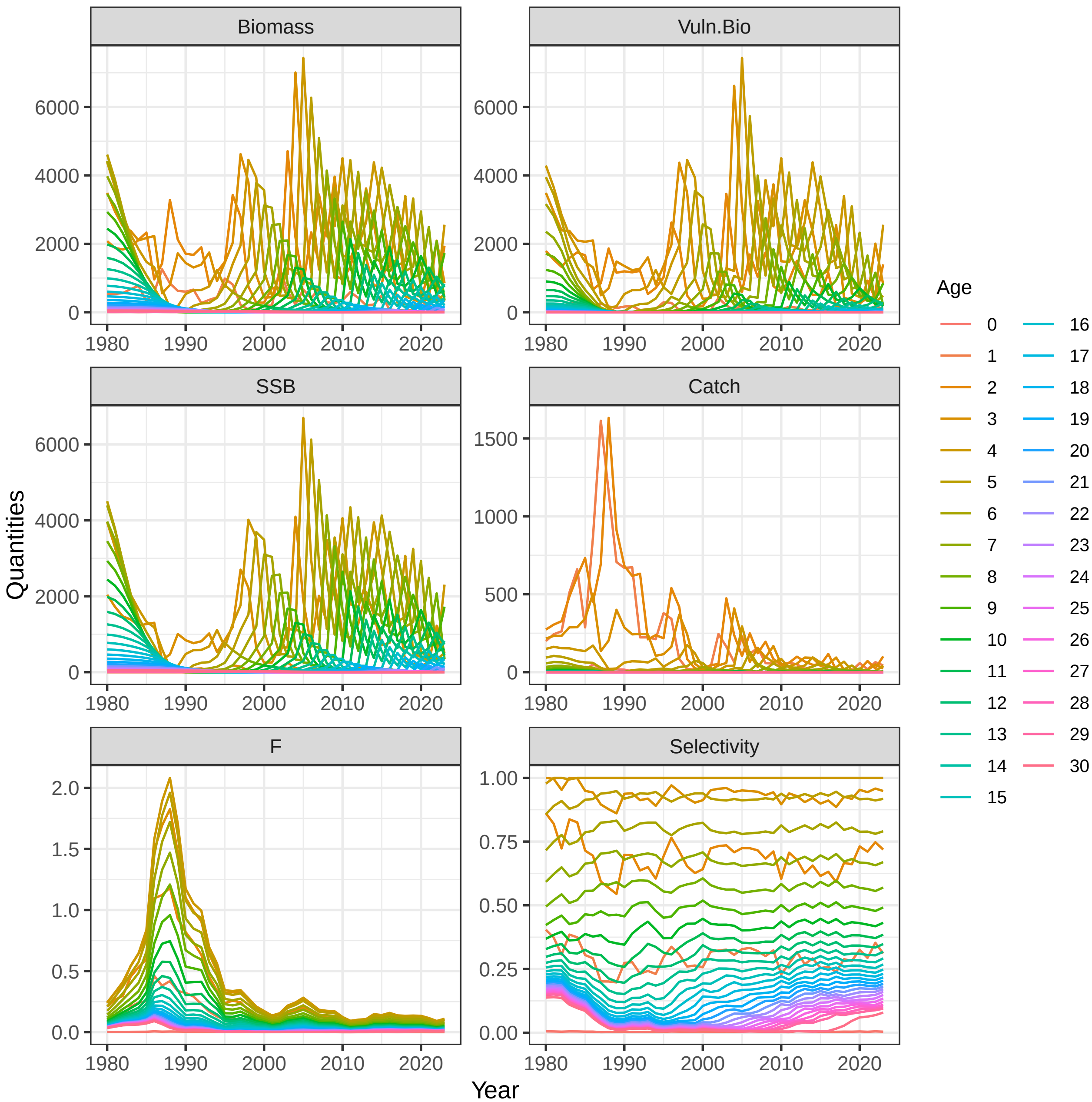

Figure 2: Stock assessment trajectories at age

```
plotbioyr(stk)+
  ggtitle(paste0(stk@name))
```

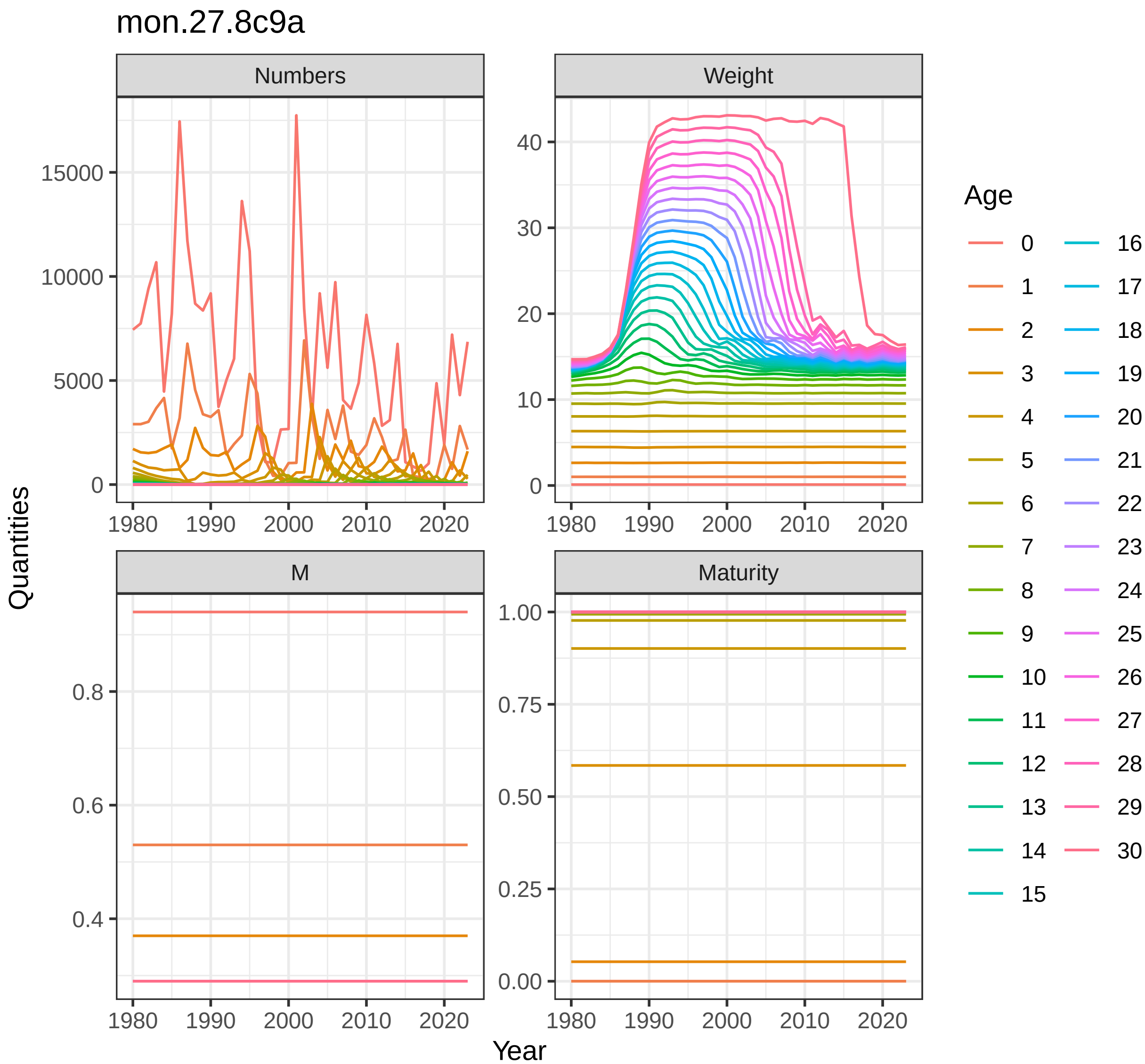

Figure 3: Stock biology trajectories at age

```r
plotbioage(stk)+theme(legend.position = "none")
```

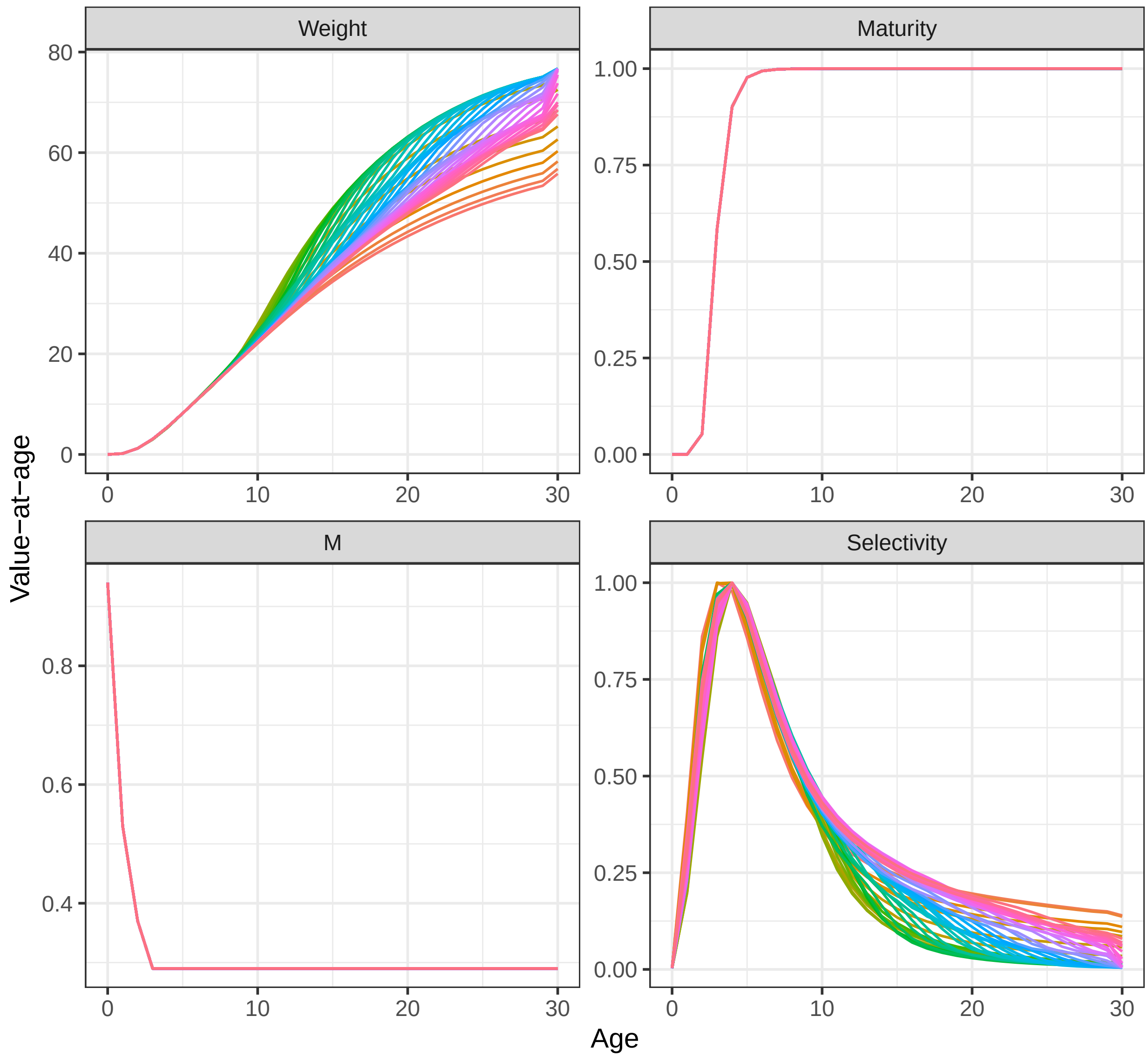

Figure 4: Annual stock quanties at age

2.2 Consistency checks using backtesting

Set seed

```
set.seed(123)
```

Get bias adjusted recruitment deviations from ss3 model

Simplify to annual sex-structured model

```
if(dims(stk)$season>1){
stka = simplify(stk,'season',weighted = TRUE,harvest=TRUE)

discards.wt(stka) = stock.wt(stka)
stka@discards  =  computeDiscards(stka)
# Make annual sra
sra = sr
params(sra) = FLPar(an(sr@params[,1]),params=rownames(sr@params))
} else {
sra = sr
stka = stk

}
```

```
yrs = an(dimnames(stk)$year)
recruit = out$recruit[out$recruit$Yr%in%yrs,]
dms <- list(year = yrs)
sigR =  mean(an(out$sigma_R_info[1:2,"SD_of_devs_over_sigma_R"])) # Realised sigR
residuals <- FLQuant(exp(recruit$dev - 0.5 * recruit$biasadjuster *sigR^2),
        dimnames = c(age = 0, dms), units = "")
recs = FLQuant(recruit$pred_recr, dimnames = c(age = 0, dms), units = "")


if (dims(stk)$unit == 2) recs <- expand(recs, unit = c("F", "M"))
```

```
if (dims(stka)$unit == 2)
yrs = an(dimnames(stka)$year)
testC = fwd(stka,sr=recs[,ac(yrs[-1])],
  control=fwdControl(year=yrs[-1], value=(unitSums(catch(stka)[, ac(yrs[-1])])),
  quant="catch"))

testF = fwd(stka, sr=recs[,ac(yrs[-1])],
  control=fwdControl(year=yrs[-1], value=unitMeans(fbar(stka)[, ac(yrs[-1])]),
  quant="fbar"))
```

```
plot(window(FLStocks(ss3om=stka,backtestC=testC,backtestF=testF)))+theme_bw()+facet_wrap(~qname,scale="
```

Note that minor deviations are likely due to difficulties in precisely adjusting the rec devs with bias correction.

2.3 Estimate candidate reference points

```
# Extract pars
s = params(sra)[[1]]
R0 = params(sra)[[2]]
B0 = params(sra)[[3]]
# Main recdevs
recyrs = recruit$Yr[recruit$era =="Main"]
```

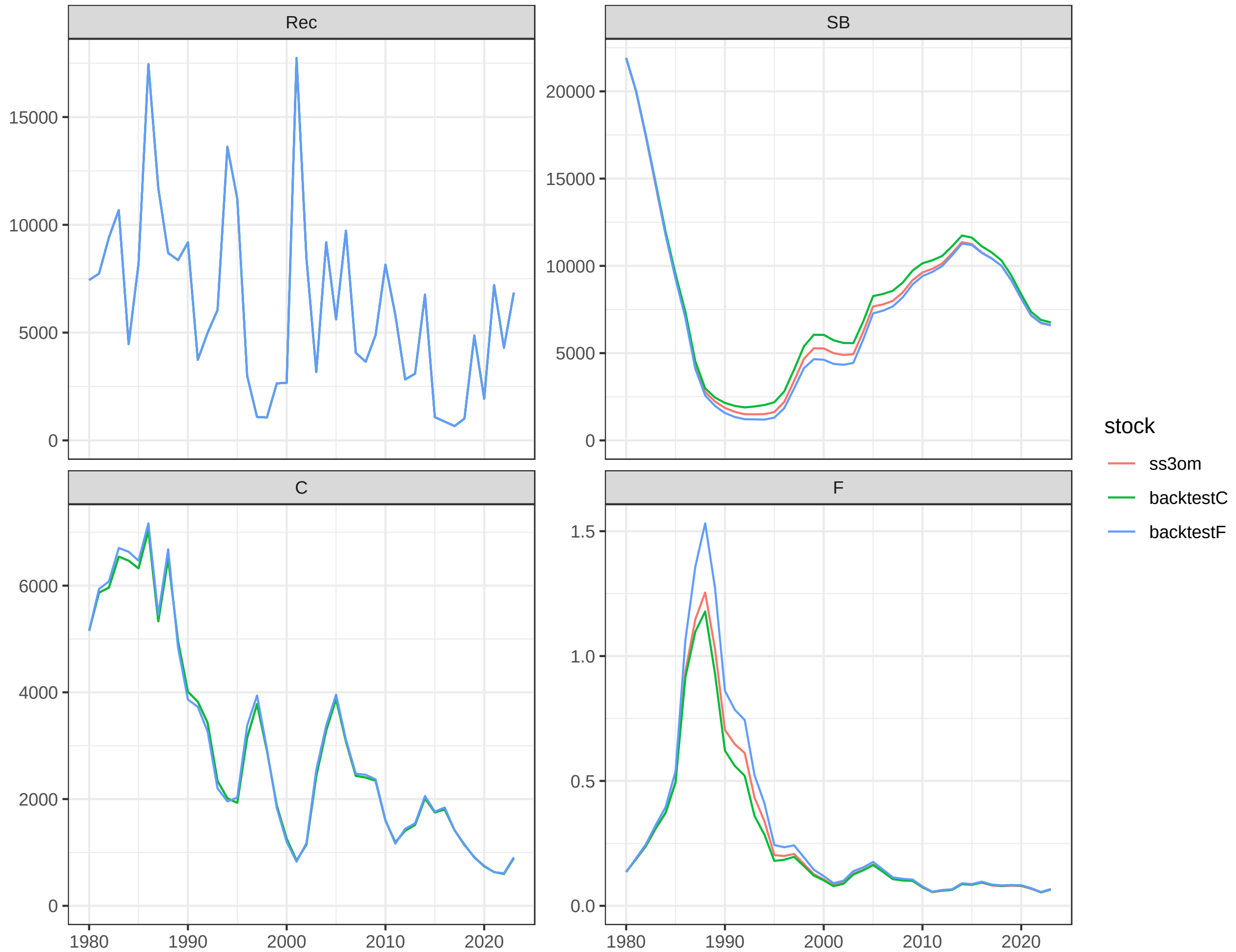

Figure 5: Comparison of stock trajectories from ss3om and a backtest under the same Fbar

```r
maindevs = unitSums(residuals[,ac(recyrs)])
rho = cor(maindevs [,-1],maindevs [,-length(maindevs)])
sigmaR = out$sigma_R_in

rho
   [1] 0.4441913
sigmaR
   [1] 0.8

# MSY refpts
Bmsy <- out$derived_quants$Value[out$derived_quants$Label=="SSB_MSY"]
Fmsy <- out$derived_quants$Value[out$derived_quants$Label=="annF_MSY"]
MSY <- out$derived_quants$Value[out$derived_quants$Label=="Dead_Catch_MSY"]
out$derived_quants$Value[out$derived_quants$Label=="B_MSY/SSB_unfished"]
   [1] 0.197853
# Short cut devs
ay = out$endyr # assessment year
SSBcv <- out$derived_quants$StdDev[out$derived_quants$Label==paste0("SSB_",ay)]/
  out$derived_quants$Value[out$derived_quants$Label==paste0("SSB_",ay)]

Fcv <- out$derived_quants$StdDev[out$derived_quants$Label==paste0("F_",ay)]/
  out$derived_quants$Value[out$derived_quants$Label==paste0("F_",ay)]
```

3 Tuning grid

Specify EQSIM outputs

```r
Blim=2936
Bpa=4373
Btri.eq = 4373
Fmsy.eq =0.198
Fp05.eq =0.25
```

Note that in this case the “true” F_{MSY} as the property of the model is smaller than the F_{MSY} derived from EQsim. This may be explained by the presence of the harvest control rule resulting in effective taking place, on average, below Btrigger.

```r
Fmsy # SS3
   [1] 0.181997
Fmsy.eq
   [1] 0.198
```

Function to find B for F at equilibrium

```r
fwdB4F = function(stock,sr,Fs=0.2,nfy=100){

    if (class(stock) == "FLStockR") {

        stock = as(stock, "FLStock")
    }
    fyrs = (dims(stock)$maxyear + 1):(dims(stock)$maxyear + nfy)
    nfy = length(fyrs)
    stkf = stf(stock, nfy)
```

```
    bx = do.call(c, lapply(an(Fs),function(x){
    ictrl = fwdControl(data.frame(year = fyrs, quant =  "fbar", value = x))
    out = fwd(stkf, sr = sr, control = ictrl)
    an(tail(unitSums(ssb(out))))
    }))

    data.frame(F=an(Fs),B=bx)
}
```

```
Fx = c(rev(seq(0.6,1,0.025)))
Ftgt = FLPar(c(Fmsy,Fmsy.eq,Fx*Fmsy),params=c("Fmsy.om","Fmsy.eq",paste0(Fx,"Fmsy")))
Ftgt
   An object of class "FLPar"
   params
     Fmsy.om    Fmsy.eq      1Fmsy 0.975Fmsy  0.95Fmsy 0.925Fmsy   0.9Fmsy 0.875Fmsy
       0.182      0.198      0.182      0.177      0.173      0.168      0.164      0.159
    0.85Fmsy 0.825Fmsy   0.8Fmsy 0.775Fmsy  0.75Fmsy 0.725Fmsy   0.7Fmsy 0.675Fmsy
       0.155      0.150      0.146      0.141      0.136      0.132      0.127      0.123
    0.65Fmsy 0.625Fmsy   0.6Fmsy
       0.118      0.114      0.109
   units:  NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
Bftgt = fwdB4F(stka,sra,Fs=Ftgt,nfy=200)

Bftgt
```

```
Ftgt.tune = FLPar(c(Bftgt$F,Bftgt$B),params=c("Fmsy.om","Fmsy.eq",paste0(Fx,"Fmsy"),
                                              "Bmsy.om","Bmsy.eq",paste0("B",Fx,"ftgt")))
```

```
np = an(nrow(Ftgt.tune))

df = data.frame(
Tune = rownames(Ftgt.tune)[1:(np/2)],
Ftgt = round(an(Ftgt.tune)[1:(np/2)],3),
Btrigger = round(Btri.eq,1),
Btgt = round(an(Ftgt.tune)[(np/2+1):np],1),
"xB0" =  round(an(Ftgt.tune)[(np/2+1):np]/B0,3))
df$Btrigger[1] = 0
df$xB0 = round(df$xB0,2)
```

```
knitr::kable(df,"pipe",
      align ="lcccccc",
      caption="Option: Initial tuning grid with EQSIM Btrigger based on the true Fmsy.om tuning with Fms
```

Table 1: Option: Initial tuning grid with EQSIM Btrigger based on the true Fmsy.om tuning with Fmsy.eq for reference.

Tune	Ftgt	Btrigger	Btgt	xB0
Fmsy.om	0.182	0	5973.2	0.21
Fmsy.eq	0.198	4373	5186.1	0.18
1Fmsy	0.182	4373	5973.2	0.21
0.975Fmsy	0.177	4373	6217.3	0.22
0.95Fmsy	0.173	4373	6471.1	0.22
0.925Fmsy	0.168	4373	6734.9	0.23
0.9Fmsy	0.164	4373	7009.2	0.24
0.875Fmsy	0.159	4373	7294.5	0.25
0.85Fmsy	0.155	4373	7591.0	0.26
0.825Fmsy	0.150	4373	7899.3	0.27
0.8Fmsy	0.146	4373	8219.9	0.28
0.775Fmsy	0.141	4373	8553.2	0.30
0.75Fmsy	0.136	4373	8899.7	0.31
0.725Fmsy	0.132	4373	9260.0	0.32
0.7Fmsy	0.127	4373	9634.6	0.33
0.675Fmsy	0.123	4373	10024.0	0.35
0.65Fmsy	0.118	4373	10428.8	0.36
0.625Fmsy	0.114	4373	10849.7	0.38
0.6Fmsy	0.109	4373	11287.3	0.39

```
refpts = FLPar(Fmsy=Fmsy,Fmsy.eq=Fmsy.eq,Bmsy=Ftgt.tune["Bmsy.om"],Bmsy.eq=Ftgt.tune["Bmsy.eq"],Blim = B

refpts
   An object of class "FLPar"
   params
       Fmsy  Fmsy.eq     Bmsy  Bmsy.eq     Blim      Bpa Btrigger       B0
   1.82e-01 1.98e-01 5.97e+03 5.19e+03 2.94e+03 4.37e+03 4.37e+03 2.89e+04
         R0
   7.44e+03
   units:  NA
```

Create `FLStockR` with `@refpts`

```
stkr = FLStockR(stka)

stkr@refpts = refpts
```

Summarize short-cut params

```
spars = FLPar(s=s,sigmaR=sigmaR, rho=rho,Fcv=Fcv,SSBcv=SSBcv)
```

```
plotAdvice(stkr)
```

```
save(stk,stka,stkr,refpts,sr,sra,spars,file="rdata/om.ss3ref.mon.rdata")
```

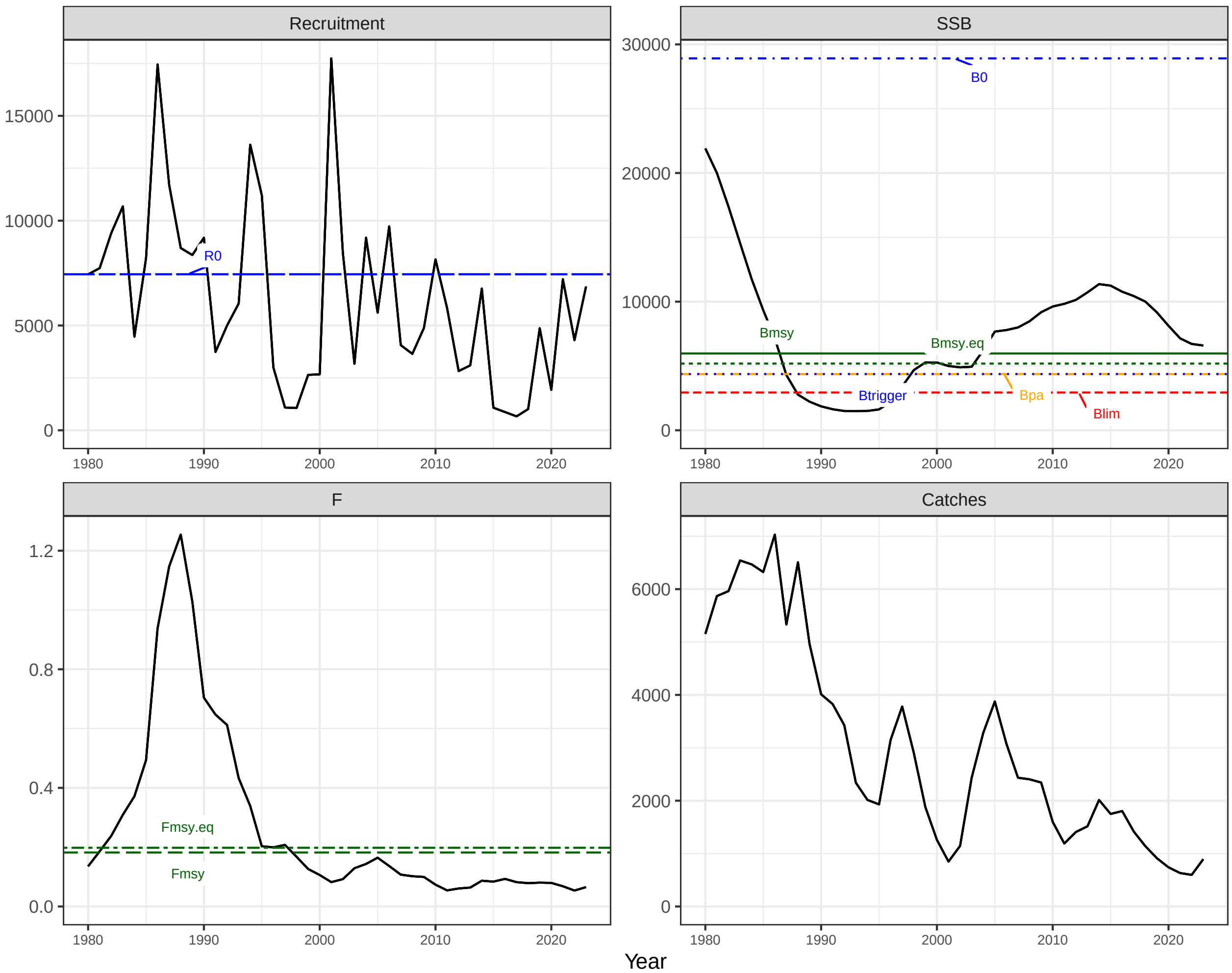

Figure 6: Status Advice plot showing stock trajectories of Recruitment, SSB, F, recruitment and $Yield$

4 Set up short-cut MSE

Load OM conditioned to Stock Synthesis

```
load("rdata/om.ss3ref.mon.rdata",verbose=T)
   Loading objects:
     stk
     stka
     stkr
     refpts
     sr
     sra
     spars
```

Next set up the MSE horizon

```
# data year
dy <- dims(stka)$maxyear
# FINAL year
fy <- dy+50
# assessment year
ay = dy+1
# intermediate years
iy = ay
```

For illustration the number of iterations are reduced to 100.

```
# NUMBER iterations
it <- 1000
```

Subset the 1000 simulated stock iterations to the first 100

```
stki = propagate(stka,it)
```

Generate recruitment

```
srdevs <- rlnormar1(n=it,sdlog=spars["sigmaR"], rho=spars["rho"], years=seq(dy, fy))
# Sex-structured
if (dims(stki)$unit == 2) srdevs  <- expand(srdevs , unit = c("F", "M"))
```

Now construct the `FLom` object from the `mse` package by passing on `FLStock`,`refpts`, `sr` and the method used for foward projections.

```
om <- FLom(stock=stki, refpts=refpts,
           sr=sra, projection=mseCtrl(method=fwd.om),deviances=srdevs)

class(om)
   [1] "FLom"
   attr(,"package")
   [1] "mse"
```

Next add the structure for the future years: average of last 3 years

```
om <- fwdWindow(om, end=fy)
```

Next, a so called observation error is constructed. In the case of the short-cut MSE, it simply holds the "perfect" stock information. For a full MSE with inbuilt estimation model it would also generate the observations with errors, such a catch-at-age and survey numbers at age for SAM or a4a, or biomass surveys indices and catches for SPicT or JABBA.

```
oem <- FLoem(
  observations=list(stk=stock(om)),
  method=perfect.oem
)
```

However, there is increasing realisation that the assessment estimates are imperfect. Therefore, ICES has implemented procedures to add uncertainty about the key quantities F and SSB, where the error on F is specified by a the random error term Fcv and a first order autocorrelation parameter $Fphi$ and the precision of SSB can specified by $SSBcv$

Short-cut deviations

```
sdevs <- shortcut_devs(om, Fcv=spars["Fcv"], Fphi=0.432, SSBcv=spars["SSBcv"])
```

Finally, the implementation error module `iem` is setup. In this case, with a random catch implementation error of 10%.

```
iem <- FLiem(method=noise.iem,
  args=list(noise=rlnorm(it, rec(om) %=% 0, 0.1)))
```

```
save(om, oem,sdevs,iem, file="rdata/flom.mon.rda")
```

4.1 Setting up harvest control rules

This can be effectively implemented ICES advice rule hockey-stick by setting the $B_{trigger}$ to zero, using the `icesControl` function. This function can also take the `SSBdevs` and `Fdevs` that implement the deviations from the SSB and F with aim to account for assessment errors.

Adjusted mse controls for sex-structured stocks

```

shortcut.sa2 <- function(stk, idx, SSBdevs=unitSums(ssb(stk)) %=% 1, args, tracking, ...) {
  # DIMS
  y0 <- args$y0
  dy <- args$dy
  ay <- args$ay
  it <- args$it

  # SUBSET oem stock
  stk <- window(stk, end=dy)

  ind <- FLQuants(
    # SSB + devs
    #><> add unitSums
    ssb=unitSums(ssb(stk)) * window(SSBdevs, start=y0, end=dy))

  track(tracking, "conv.est", ac(ay)) <- 1

  list(stk=stk, ind=ind, tracking=tracking)
}



tac.is2 <- function(stk, ctrl, args, output="catch", recyrs=-2,
  Fdevs=unitMeans(fbar(fut)) %=% 1, dtaclow=NA, dtacupp=NA, fmin=0, reuse=TRUE,
  initac=metrics(stk, output)[, ac(iy - 1)], tracking) {
```

```r
# EXTRACT args
spread(args)


# SET control years
cys <- seq(ay + management_lag, ay + management_lag + frq - 1)

# PREPARE stk for cys, biology as in last nsqy years
fut <- fwdWindow(stk, end=cys[length(cys)], nsq=nsqy)

# PARSE recyrs if numeric
id <- dimnames(stk)$year

# COERCE to list
if(!is.list(recyrs)) {
  recyrs <- list(recyrs)
}

# PARSE list
for(i in recyrs) {
  if(is(i, 'character')) {
    id <- id[!id %in% i]
  } else if(all(i < 0)) {
    if(length(i) == 1)
      id <- rev(rev(id)[-seq(abs(i))])
    else
      id <- rev(rev(id)[i])
  } else if(all(i > 0)) {
    id <- rev(rev(id)[seq(abs(i))])
  }
}

# SET years to use
recyrs <- id

# CHECK recyrs
if(!all(recyrs %in% dimnames(stk)$year)) {
  stop("'recyrs' cannot be found in input stk")
}

# TODO: OTHER rec options


# SET GM recruitment from past
#><> add unitSums()
gmnrec <- exp(yearMeans(log(unitSums(rec(stk))[, recyrs])))

# SETUP SRR
srr <- predictModel(model=rec~a, params=FLPar(a=gmnrec))

# STORE geomeanrec value
track(tracking, "gmrec.isys", ay + management_lag) <- gmnrec
```

```r
# ADD F deviances for 1 year

# reuse = TRUE
if(isTRUE(reuse) | toupper(reuse) == 'F') {
  ftar <- rep(c(ctrl[1,]$value * Fdevs[, ac(cys[1])]), length(cys))
# reuse = FALSE
} else {
  ftar <- c(ctrl$value * Fdevs[, ac(cys)])
}

# TRACK Ftarget
track(tracking, "fbar.isys", cys) <- ftar

# FORECAST for iyrs and my IF mlag > 0,
if(management_lag > 0) {

  # SET F for intermediate year #><> added unitMeans
  #><> add unitMeans()

  fsq <- unitMeans(fbar(stk))[, ac(dy)]

  # TODO: ADD TAC option

  # CONSTRUCT fwd control
  fctrl <- fwdControl(
    # ay as intermediate with Fsq TODO: Other options
    list(year=seq(ay - data_lag + 1, length=management_lag),
      quant="fbar", value=rep(c(fsq), management_lag)),
    # target
    list(year=cys, quant="fbar", value=c(ftar))
  )

# else only for my
} else {
  fctrl <- fwdControl(
    list(year=ay + management_lag, quant="fbar", value=ftar))
}

# RUN STF ffwd
fut <- ffwd(fut, sr=srr, control=fctrl)

# ID iters where hcr set met trigger and F > fmin
id <- c(tracking[[1]]["decision.hcr", ac(ay)] > 2) &
  c(unitMeans(fbar(fut))[, ac(ay + management_lag)] > fmin)

# EXTRACT catches
if(isTRUE(reuse) | toupper(reuse) == "C") {
  TAC <- expand(unitSums(catch(fut))[, ac(cys)[1]], year=seq(length(cys)))
} else {
  TAC <- unitSums(catch(fut))[, ac(cys)]
}

# GET TAC dy / ay - 1
```

```
  if(ay == iy)
    prev_tac <- rep(c(initac), length=args$it)
  else
    prev_tac <- c(tracking[[1]]["isys", ac(ay)])

  # APPLY upper and lower TAC limit, if not NA and only for id iters
  if(!is.na(dtacupp)) {
    iter(TAC, id) <- pmin(c(iter(TAC, id)), prev_tac[id] * dtacupp)
  }
  if(!is.na(dtaclow)) {
    iter(TAC, id) <- pmax(c(iter(TAC, id)), prev_tac[id] * dtaclow)
  }

  # CONSTRUCT fwdControl
  # TODO: USE frq here
  ctrl <- fwdControl(lapply(seq(length(cys)), function(x)
    list(year=cys[x], quant=output, value=TAC[,x])))

  return(list(ctrl=ctrl, tracking=tracking))
}
```

```
arule <- mpCtrl(list(
  # (est)imation method: shortcut.sa + SSB deviances
  est = mseCtrl(method=shortcut.sa2,
    args=list(SSBdevs=sdevs$SSB)),

  # hcr: hockeystick (fbar ~ ssb | lim, trigger, target, min)
  hcr = mseCtrl(method=hockeystick.hcr,
    args=list(lim=0, trigger=Btri.eq, target=Fmsy.eq,
    min=0, metric="ssb", output="fbar")),

  # (i)mplementation (sys)tem: tac.is (C ~ F) + F deviance
  isys = mseCtrl(method=tac.is2,
    args=list(recyrs=-2, fmin=0, Fdevs=sdevs$F))
  ))
```

This rule can now be run passing on the `om`, `oem` and `arule` and an additional argument to set the implementation year to 2024.

Note that the default setting assumes 1 year lag between data (reference) year and assessment (reporting) year and that the TAC is implemented the next year. In case the assessment is conducted in 2023, based on data from 2022 and the TAC is implemented for 2024.

```
  Loading objects:
    stks
```

```
mseargs <- list(iy=dy, fy=fy, data_lag=1, management_lag=1, frq=1)

system.time(
run <- mp(om, oem=oem, ctrl=arule, args=mseargs,verbose=T)
)

mp.eqsim  = run@om@stock
```

```
# make FLStocks from om until 2024 (implementation) and the run
```

```
stk.eqsim = FLStocks(stock=window(om@stock,end=dy),
              fixedFmsy=mp.eqsim)

plot(stk.eqsim )+facet_wrap(~qname,scales="free")+
  theme_bw()+
   geom_vline(xintercept = c(dy),linetype=2,col=1)+
 geom_vline(xintercept = c(fy),linetype=2,col=4)
```

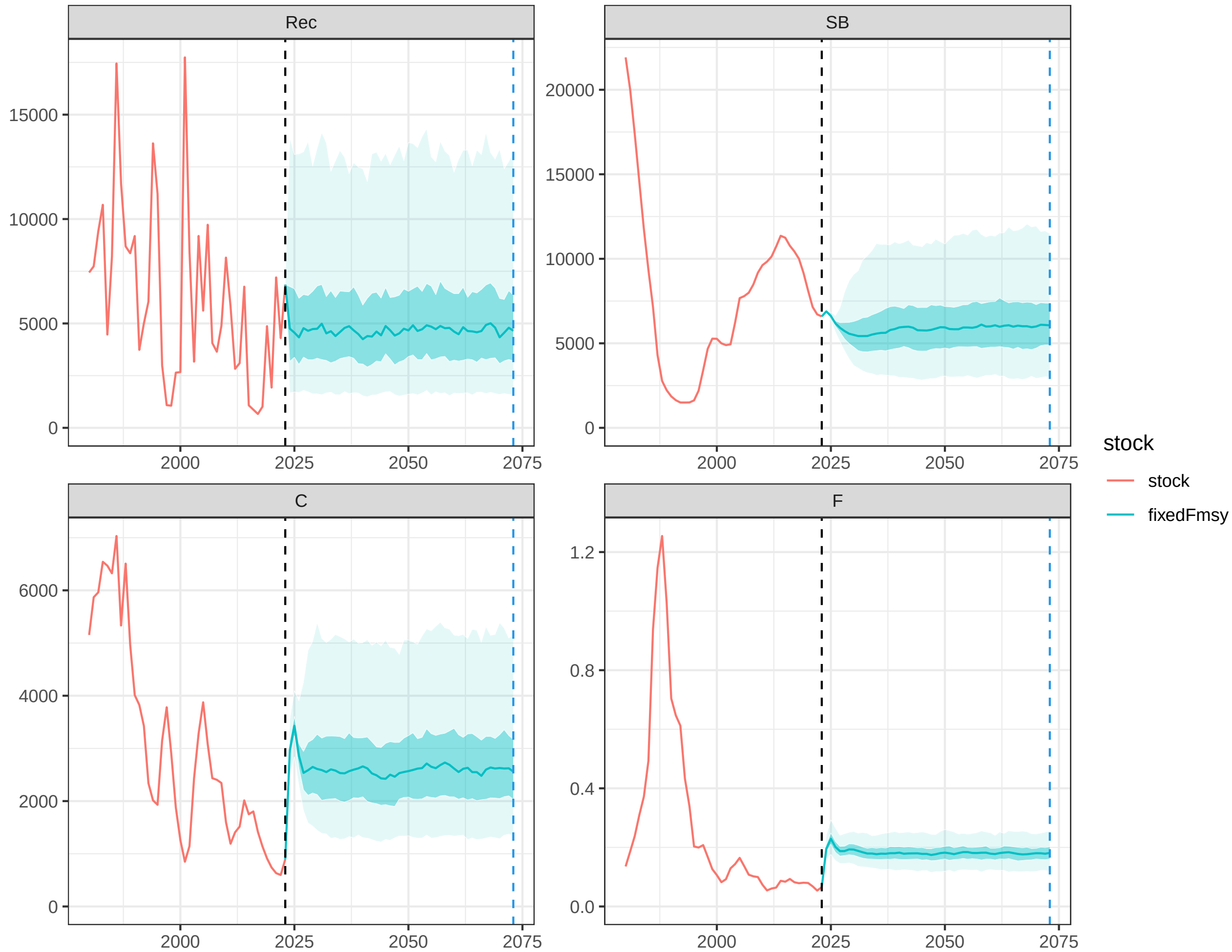

Figure 7: Initial OM and the MSE forecast horizon under a fixed F_{tgt} rule

The `run` can also be appended to the `om` to make a single `FLStockR` object with reference points

```
runR = FLStockR(append(window(om@stock,end=dy),mp.eqsim))
runR@refpts=refpts
```

This allows to quickly evaluate the stock status under a fixed F_{MSY} rule.

It can be seen that despite "perfect" knowledge of the "true" F_{MSY}, and fishing pressure is on average F_{MSY}, the stock fails to attain biomass levels at B_{MSY} with a relative high risk to fall below B_{lim}. This is a well known fact as a result of the lags between data and management and asymetric risks in that exceeding F_{MSY} is more consequential on both SSB and long term yield, then fishing below F_{MSY}, In the case of the latter, more biomass is left in the water, which provides increased future reproduction potential and catch opportunity.

```
thin = seq(1,1000,4)

plotAdvice(iter(runR,thin))+
     geom_vline(xintercept = c(dy),linetype=2,col=1)+
    geom_vline(xintercept = c(fy),linetype=2,col=4)+
  scale_x_continuous(breaks=seq(1970,3000,5))+
  theme(axis.text.x = element_text(size=8, angle=90,vjust=0.5))
```

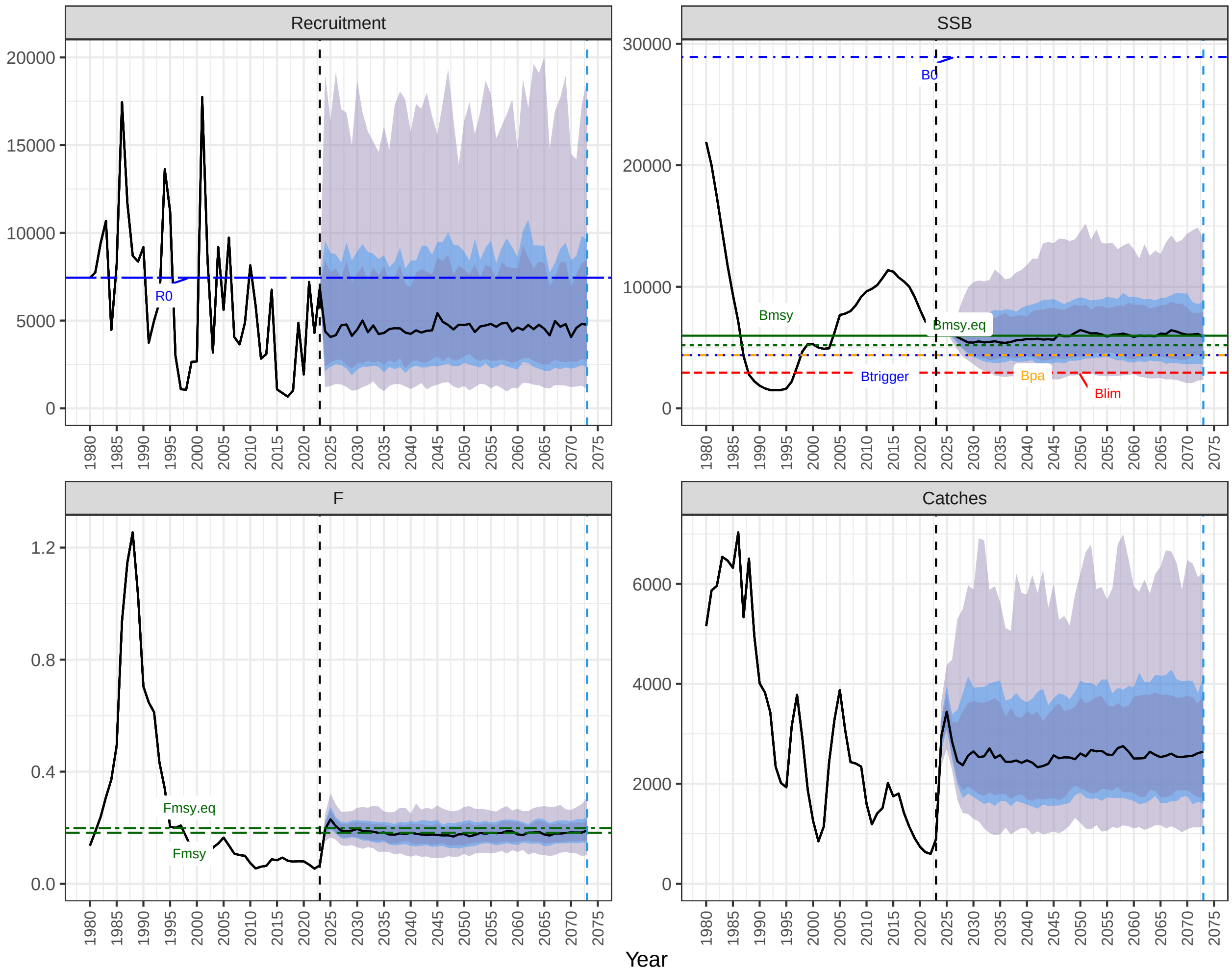

Figure 8: Stock Status under a eqsim based Advice rule

Even relatively simplified short-cut MSE frameworks provide a powerful to explore alternative HCRs to achieve better trade-off between risks and yield.

Here, the convential hockey-stick control rule is explored with different ratios of F_{adv}/F_{tgt} and $B_{trigger}/B_{tgt}$ settings, where the $B_{trigger}$ promts a linear reduction in F_{adv} if SSB falls below it.

```
plotGFCM(fadv = Fmsy.eq,btrigger =Btri.eq,bthr = -1,ftgt = Fmsy,btgt=df$Btgt[2],
         blim=an(refpts["Blim"]),fmin=0,bclose=0,kobe  =FALSE,text=F,rel=F,xmax=2.5)
```

```
Fadv = c(df$Ftgt)
Btgt = c(df$Btgt)
Btri = c(df$Btrigger)
```

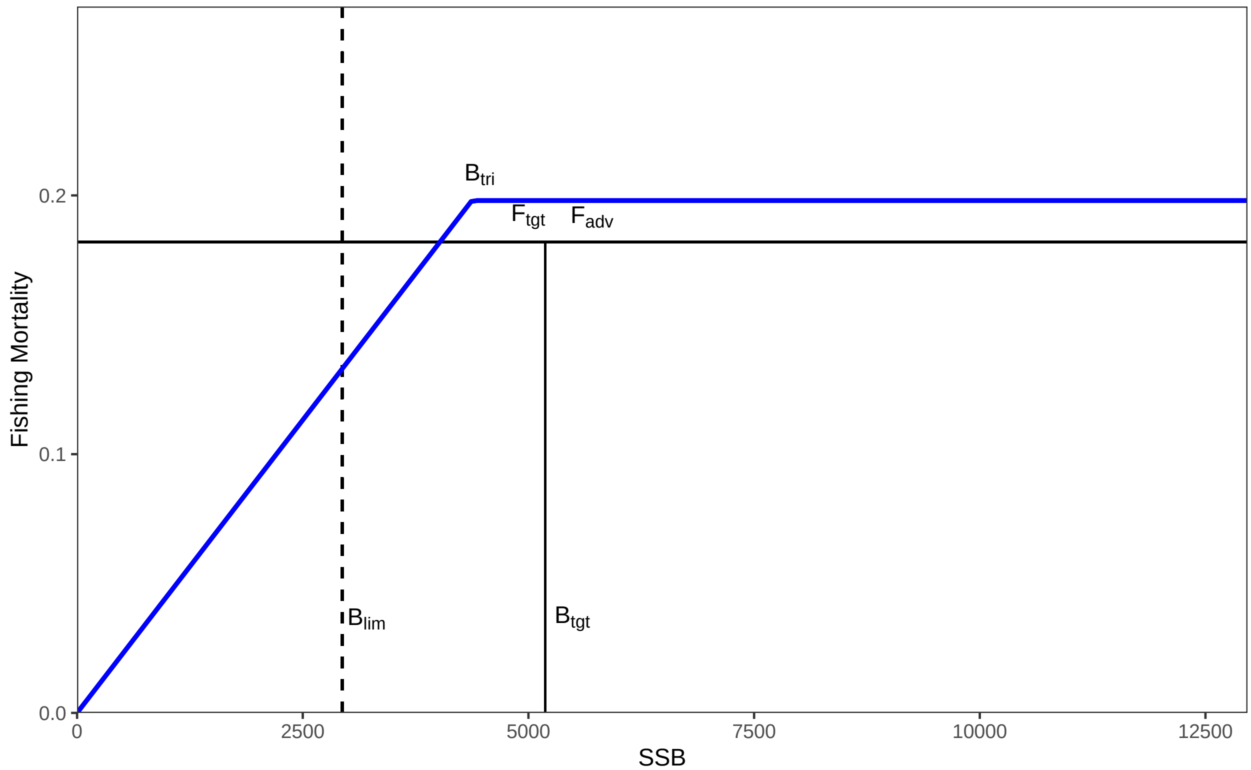

Figure 9: Advice Rule from EQSIM, with *Ftgt* denoting the true *Fmsy* property of the SS3 benchmark model

```
hcrs= Map(function(x,y,z){
p  = plotGFCM(fadv = x,btrigger =y,ftgt = Fmsy,btgt=z,blim=an(refpts["Blim"])
              ,fmin=0,bclose=0,kobe  =FALSE,text=F,rel=F)
p = p
return(p)
},x=Fadv,y=Btri,z=Btgt)
# plot ggplot list
ggarrange(plotlist=hcrs,nrow=5,ncol=4)
```

The same settings can be specified for the new `mps` function in `mse`, which allow to explore variations of the HCR parameters.

The function `combinations` enables to vary more than one parameter at the time.

```
hcrs =combinations(target=Fadv,
                   trigger=0)
hcrs$trigger = Btri # Overwrite
hcrs
   $target
    [1] 0.182 0.198 0.182 0.177 0.173 0.168 0.164 0.159 0.155 0.150 0.146 0.141
   [13] 0.136 0.132 0.127 0.123 0.118 0.114 0.109

   $trigger
    [1]    0 4373 4373 4373 4373 4373 4373 4373 4373 4373 4373 4373 4373 4373 4373
   [16] 4373 4373 4373 4373
```

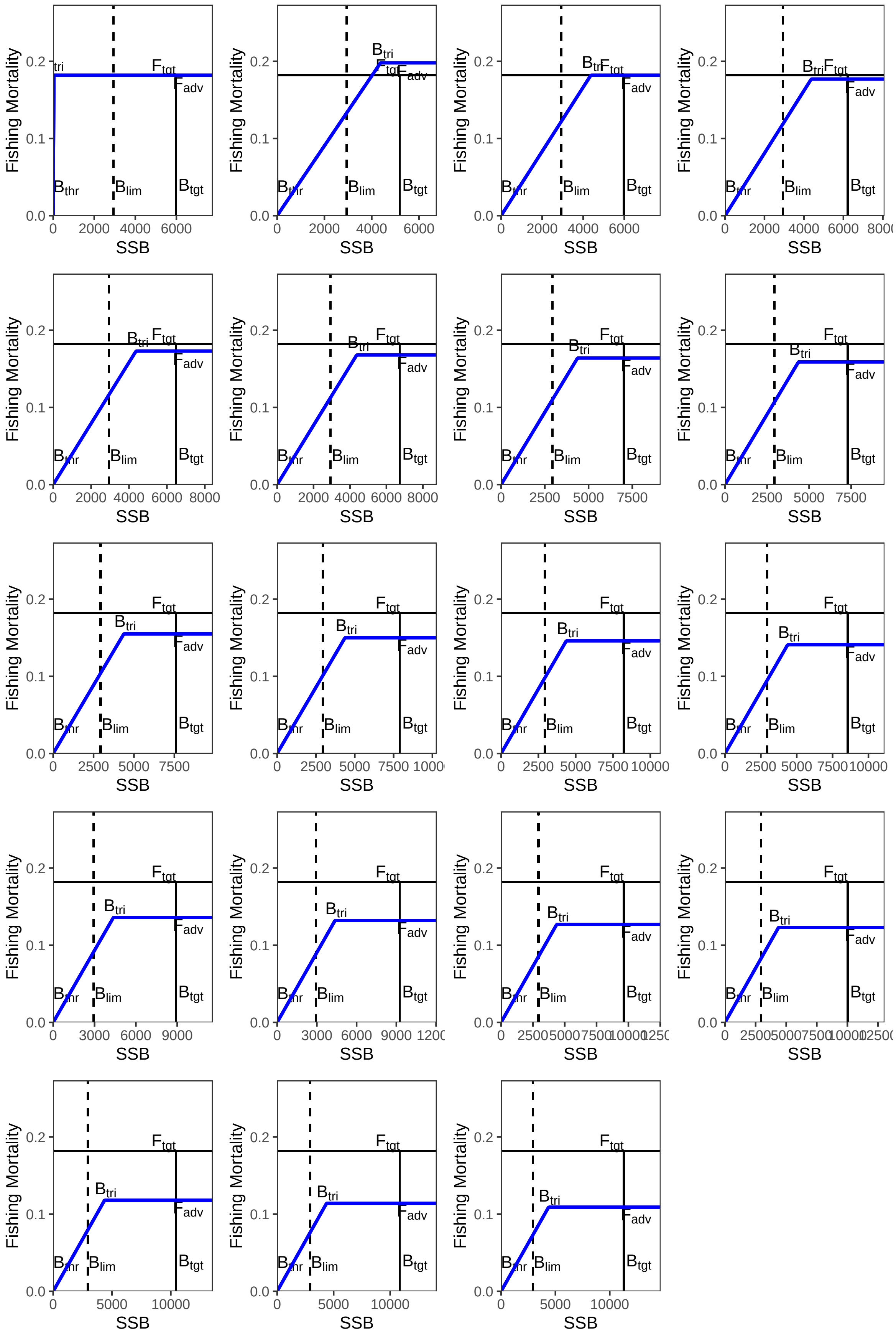

Figure 10: Alternative HCR formulations with different tuning levels relative F_{tgt}, with $Ftgt$ denoting the true $Fmsy$ property of the SS3 benchmark model

```
save(om,oem,arule,hcrs,df,mseargs,file="rdata/inpMP.mon.1000.rdata")
```

These changes in parameters can simply be passed on to the existing `arule` to run the variations with `mps`.

```
runs <- mps(om, oem=oem, ctrl=arule, args=mseargs,
  hcr=hcrs)
```

Note the this was run on a linux cluster in parallel

```
library(doParallel)
library(FLRef)
load("~/mseLite/mse_refpts/rdata/inpMP.mon.1000.rdata",verbose=T)
length(hcrs$target)
# Run in batches

ni = length(hcrs$target)
cl = ni

registerDoParallel(cl)
start = Sys.time()

runs <- foreach(i = seq(ni)) %dopar% {

  # set stock index
  hcr = arule
  hcr$hcr@args$target = hcrs$target[i]
  hcr$hcr@args$trigger = hcrs$trigger[i]

  runi <- mp(om, oem=oem, ctrl=hcr , args=mseargs,verbose=T,parallel = FALSE)

  return(runi)
} # end of loop
end = Sys.time()
time = end-start
time




nfs = length(df$Tune[-1])
tri = c(rep(".Btri.eq",each=nfs))
scenarios = c("Fmsy.OM",paste0(rep(c(df$Tune[-1]),1),tri))

names(runs) = scenarios

stks = FLStocks(lapply(runs,function(x){
  out = x@om@stock
  out = FLStockR(out)
  out@refpts = om@refpts
  out
}))

stkm = FLStocks(lapply(stks,function(x){
  stockMedians(x)
```

```
}))

plotAdvice(stkm)

save(stks ,file=paste0("~/mseLite/mse_refpts/rdata/mps.ftune.mon1000.rdata"))
```

Now combine with the Fixed F_{MSY} run and see if we can do better.

```
nfs = length(df$Tune[-1])
tri = c(rep(".Btri.eq",each=nfs))
scenarios = c("Fmsy.eq",paste0(rep(c(df$Tune[-1]),1),tri))

names(runs) = scenarios
```

Save

An easy way of basic plotting is to extract the `FLStocks` from the list of MSE runs with `lapply`

```
stks = FLStocks(lapply(runs,function(x){
  out = (x@om@stock)
  out = FLStockR(out)
  out@refpts = refpts[c(1:5)]
  out
  }))
```

```
ref = FLStockR(window(stock(om),end=dy))
ref@refpts = refpts
pstks = FLStocks(c(FLStocks(stock=window(ref,end=dy)),
              stks))

plotAdvice(iter(pstks,thin))+facet_wrap(~qname,scales="free")+
  theme_bw()+
  scale_color_manual(values=c("black",ss3col(length(pstks))[-1]))+
  scale_fill_manual(values=c("darkgrey",ss3col(length(pstks))[-1]))+
  geom_vline(xintercept = c(dy),linetype=2,col=1)+
    geom_vline(xintercept = c(fy),linetype=2,col=4)+
  scale_x_continuous(breaks=seq(1900,3000,5))+
  theme(axis.text.x = element_text(size=8, angle=90,vjust=0.5))
```

```
medstks = FLStocks(lapply(stks,function(x){
  stockMedians(x)
}))
medref = stockMedians(ref)
pmstks = FLStocks(c(FLStocks(stock=window(medref,end=dy)),
              medstks))

plotAdvice(pmstks)+facet_wrap(~qname,scales="free")+
  scale_color_manual(values=c("black",ss3col(length(pmstks))[-1]))+
  theme_bw()+xlab("Year")+
  geom_vline(xintercept = c(dy),linetype=2,col=1)+
    geom_vline(xintercept = c(fy),linetype=2,col=4)+
  scale_x_continuous(breaks=seq(1900,3000,5))+
  theme(axis.text.x = element_text(size=8, angle=90,vjust=0.5))
```

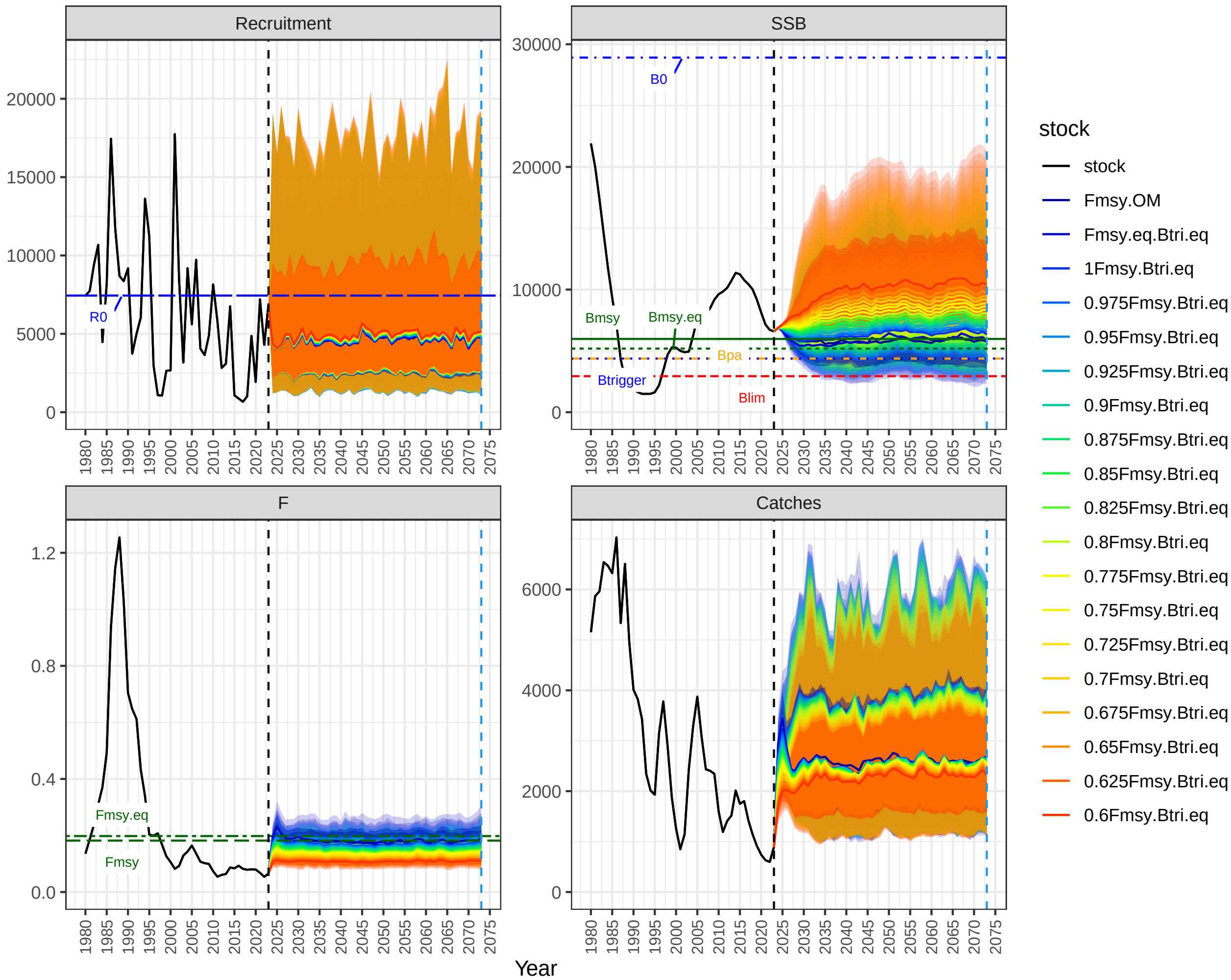

Figure 11: Initial OM and the MSE forecast horizon under a fixed F_{tgt} rule and 6 HCR hockey-stick scenarios

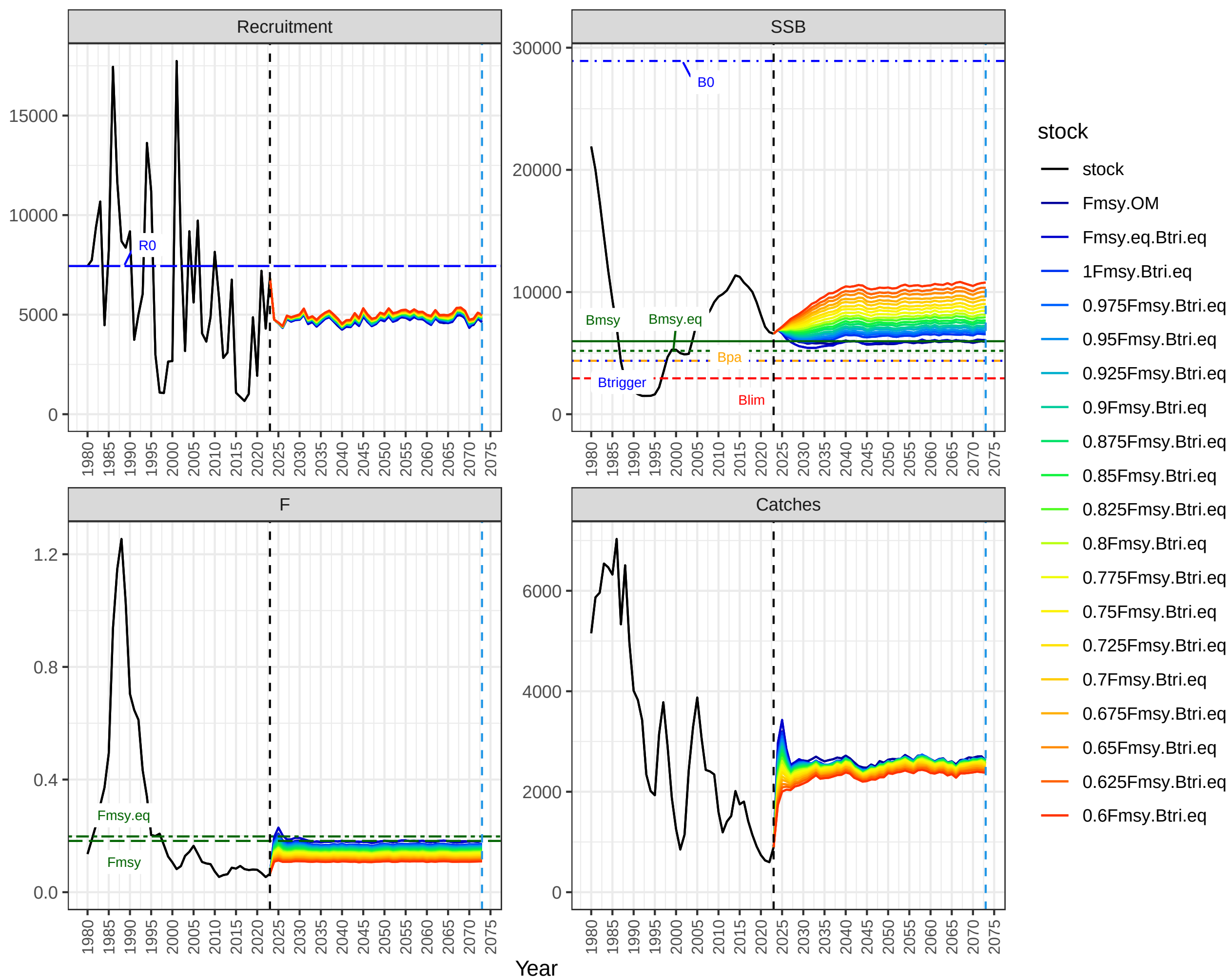

Figure 12: Median MSE forecast horizon under a fixed F_{tgt} rule and 6 HCR scenarios

5 Performance Evaluation with adjustment for sex-tructured stocks

5.1 Define Metrics for performance evaluation of sex-structured

```
metrics <- list(SB = function(x)unitSums(ssb(x)),
                F = function(x)unitMeans(fbar(x)),
                C = function(x)unitSums(catch(x)),
                Rec= function(x)unitSums(rec(x)))
```

Define performance statistics - ordered alphabetically

```

stats <- list(
  a.medianFmsy= list(~yearMedians(F/Fmsy), name="F/Fmsy",
    desc="Median annual F/Fmsy"),
  b.medianBmsy = list(~yearMedians(SB/Bmsy), name="B/Bmsy",
    desc="Median annual B/Bmsy"),
  c.medianCmsy = list(~yearMedians(C/MSY), name="Catch/MSY",
    desc="Median Catch/MSY over years"),
  d.aavC = list(~yearMedians(iav(C)), name="AAV",
    desc="Median annual variation in catches"),
  e.riskBlim = list(~apply(iterMeans((SB/Blim) < 1),1,max),
    name="P3(B<Blim)", desc="Probability that SSB < Blim"),
  f.P80BMSY = list(~apply(iterMeans((SB/(Bmsy * 0.8)) > 1), 1, max),
    name="B>80Bmsy", desc="Probability that SSB > 80% x Bmsy")

)

```

5.2 Long-term last 25 years (75 years)

Replace deterministic MSY with MSE MSY from a run with the true, fixed F_{MSY} in the reference points

```
refpts.perf = refpts
refpts.perf = rbind(refpts.perf,
                    FLPar(MSY= median(unitSums(catch(stks$Fmsy.OM[,ac((fy-24):fy)])))))
```

Compute performance metrics

```
perf <- performance(stks,refpts=refpts.perf,metrics=metrics, statistics=stats, years=list((fy-25):fy))
```

5.3 PLOT performance

```
ncol = length(unique(perf$mp))
pbp = plotBPs(perf,
statistics=c("a.medianFmsy","b.medianBmsy","c.medianCmsy", "d.aavC", "e.riskBlim", "f.P80BMSY"),
size=3, target = c(a.medianFmsy=1,b.medianBmsy=1, c.medianCmsy=1,g.P80BMSY=0.8),
limit= c(e.riskBlim=0.05,c.medianCmsy=0.95),
yminmax = c(0.05, 0.95))+theme_bw()+
facet_wrap(~name,scales = "free_y",ncol=2)+
ggtitle(paste0("Performance: Reference Points"))+
ylab("Performance statistics")+
scale_fill_manual(values=ss3col(ncol))+ # USE FLRef::ss3col
theme(axis.text.x=element_blank())+xlab("Candidates")+ guides(fill= guide_legend(ncol = 1))
pbp
```

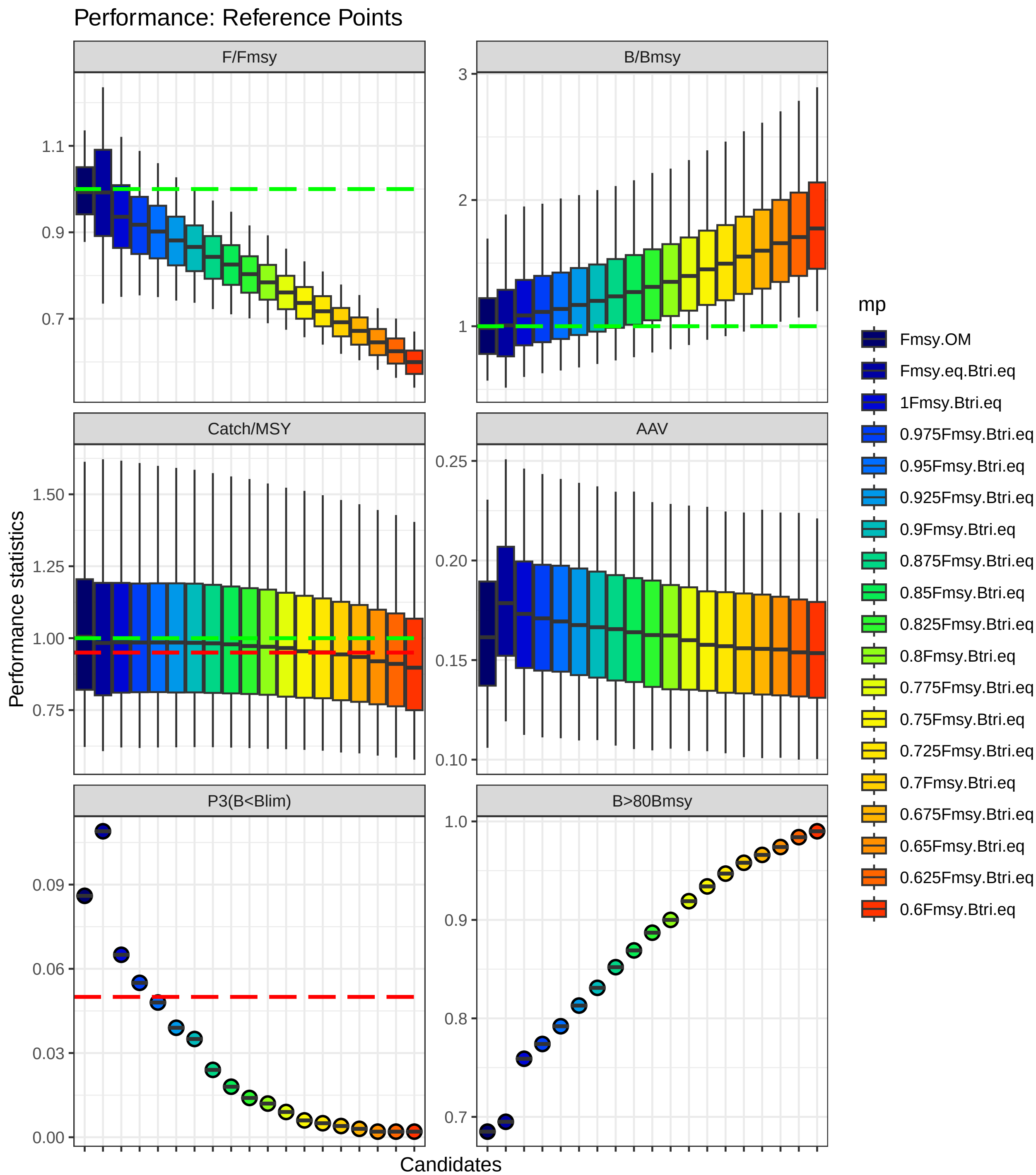

Figure 13: Performance: Long-term performance evaluation

5.3.1 Performance Table

Table 2: Robustness tests of F_{adv} in line with ICES precautionary approach under closed loop simulations with feedback control of assessment advice emulation

AR	Fadv	P3(B<Blim)	Catch/MSY	SD.Catch	AAVC	F/Fmsy	B/Bmsy	P(B>0.8Bmsy)
Fmsy.OM	0.182	0.086	1.041	0.311	0.164	0.999	1.035	0.685
Fmsy.eq.Btri.eq	0.198	0.109	1.024	0.309	0.181	0.992	1.074	0.695
1Fmsy.Btri.eq	0.182	0.065	1.028	0.304	0.175	0.936	1.151	0.759
0.975Fmsy.Btri.eq	0.177	0.055	1.028	0.302	0.173	0.917	1.179	0.774
0.95Fmsy.Btri.eq	0.173	0.048	1.027	0.301	0.172	0.901	1.204	0.792
0.925Fmsy.Btri.eq	0.168	0.039	1.026	0.299	0.170	0.881	1.236	0.813
0.9Fmsy.Btri.eq	0.164	0.035	1.024	0.298	0.169	0.865	1.265	0.831
0.875Fmsy.Btri.eq	0.159	0.024	1.021	0.295	0.167	0.844	1.303	0.852
0.85Fmsy.Btri.eq	0.155	0.018	1.018	0.293	0.166	0.826	1.335	0.869
0.825Fmsy.Btri.eq	0.150	0.014	1.013	0.290	0.164	0.804	1.378	0.887
0.8Fmsy.Btri.eq	0.146	0.012	1.008	0.288	0.163	0.786	1.415	0.900
0.775Fmsy.Btri.eq	0.141	0.009	1.001	0.285	0.162	0.762	1.463	0.919
0.75Fmsy.Btri.eq	0.136	0.006	0.993	0.281	0.161	0.738	1.514	0.934
0.725Fmsy.Btri.eq	0.132	0.005	0.985	0.278	0.160	0.718	1.558	0.947
0.7Fmsy.Btri.eq	0.127	0.004	0.975	0.274	0.159	0.693	1.615	0.958
0.675Fmsy.Btri.eq	0.123	0.003	0.966	0.271	0.159	0.673	1.664	0.966
0.65Fmsy.Btri.eq	0.118	0.002	0.953	0.266	0.158	0.648	1.728	0.974
0.625Fmsy.Btri.eq	0.114	0.002	0.942	0.262	0.157	0.627	1.781	0.984
0.6Fmsy.Btri.eq	0.109	0.002	0.927	0.257	0.156	0.601	1.852	0.990

Determine PA Advice Rule

```
# Choose precautionary threshold P(B<Blim) < 0.05
AR.PA = tab.pf[tab.pf$`P3(B<Blim)`<0.05 ,]
# Maximum catch
AR.PA = AR.PA[AR.PA$`Catch/MSY`==max(AR.PA$`Catch/MSY`,na.rm = T),]

knitr::kable(AR.PA,"pipe",
      align ="lccccc",
      caption="PA Advice Rule based on the staged critia of (1) $F_{adv}<F_{p0.5}$ and (2) maximum catc
```

Table 3: PA Advice Rule based on the staged critia of (1) $F_{adv} < F_{p0.5}$ and (2) maximum catch given (1)

	AR	Fadv	P3(B<Blim)	Catch/MSY	SD.Catch	AAVC	F/Fmsy	B/Bmsy	P(B>0.8Bmsy)
5	0.95Fmsy.Btri.eq	0.173	0.048	1.027	0.301	0.172	0.901	1.204	0.792

```
write.csv(AR.PA,file=file.path("perftabs",paste0(run,".AR.PA.csv")),row.names = F)
```

```
 plotGFCM(fadv =AR.PA$Fadv,btrigger =Btri.eq,bthr = -1,ftgt = Fmsy,btgt=Bmsy,
          blim=Blim,fmin=0,bclose=0,kobe  =FALSE,text=F,rel=F,xmax=2.5)
```

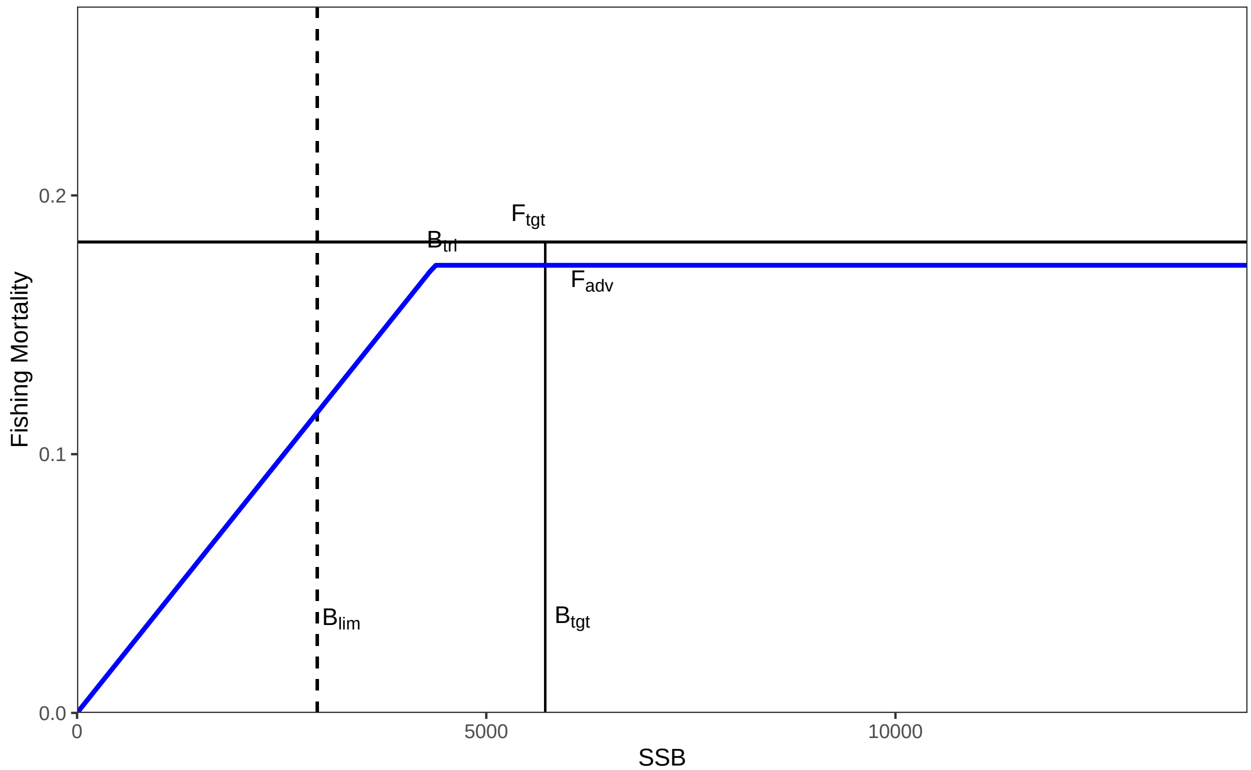

Figure 14: Advice Rule PA, with $Ftgt$ denoting the true $Fmsy$ property of the SS3 benchmark model

5.3.2 MSE kobe plot

```
kbcex =function(){theme(plot.title = element_text(size=10),
                        legend.key.size = unit(0.3, 'cm'), #change legend key size
                        legend.key.height = unit(0.4, 'cm'), #change legend key height
                        legend.key.width = unit(0.4, 'cm'), #change legend key width
                        legend.text = element_text(size=10)) #change legend text font size
}
kobeMPs(perf,y="a.medianFmsy", x="b.medianBmsy", SBlim=NULL, Ftarget = 1)+
  ylab(expression(F/F[MSY]))+xlab(expression(B/B[MSY]))+ylim(0,2.5)+kbcex()+theme()+guides(fill= guide_l
   Scale for y is already present.
   Adding another scale for y, which will replace the existing scale.
```

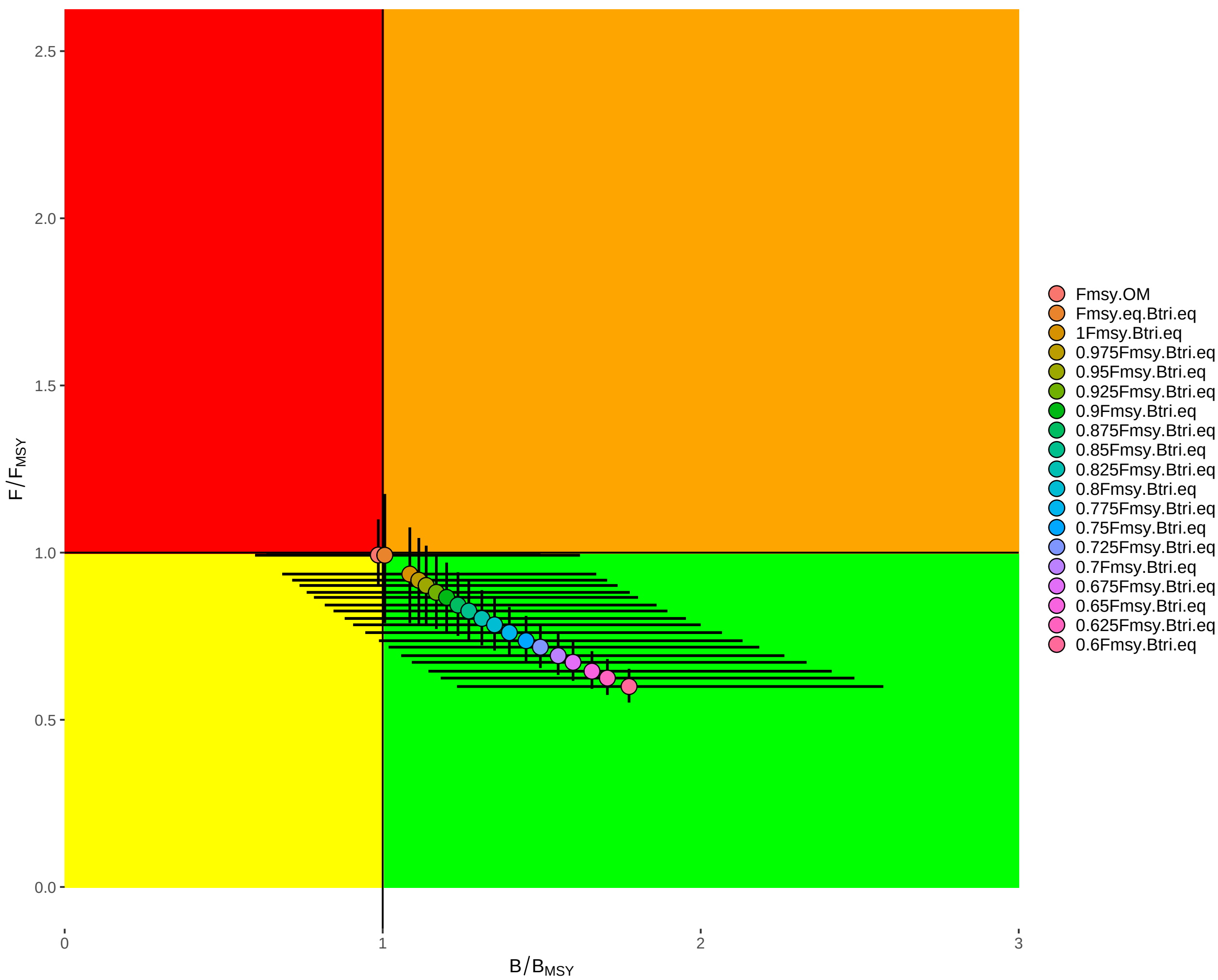

Figure 15: MSE kobe plot Advice rules