Management strategy evaluations for mean length-based management procedures using DLMtool

Quang C. Huynh

SEDAR46-RW-02

22 February 2016

Please cite this document as:

Huynh, Q.C. 2016. Management strategy evaluations for mean length-based management procedures using DLMtool. SEDAR46-RW-02. SEDAR, North Charleston, SC. 19 pp.

Management strategy evaluations for
mean length-based management procedures using DLMtool

February 22, 2016

Quang C. Huynh
Virginia Institute of Marine Science, P.O. Box 1346, Gloucester Point, VA 23062

1. Management strategy evaluations using DLMtool

Management strategy evaluations (MSEs) were undertaken for each of the 6 Caribbean stocks using DLMtool version 2.1.2. Three management procedures (MPs) were tested in the MSE, where the general equation for the overfishing limit (OFL) is:

$$OFL = F_{MSY} \times Abun = F_{MSY} \times \frac{C_{recent}}{F_{recent}} \quad (1)$$

where abundance $Abun$ is estimated from the ratio of recent catch and F_{recent} ,the latter derived from the application of the mean length mortality estimator of Gedamke and Hoenig (2006). The three MPs used three different F_{MSY} proxies: $F_{0.1}$ from a yield-per-recruit analysis (YPR_ML); and $F_{SPR=30\%}$ (SPR30_ML) and $F_{SPR=40\%}$ (SPR40_ML) using spawning potential ratios (SPR).

Populations were simulated with 75 historical years and 40 projection years, with an assessment applied every third year. A total of 500 simulations were performed for each stock with 250 stochastic replicates. In the base set of MSEs, the simulations were performed assuming 15% variability in life history parameters; asymptotic selectivity for Puerto Rico yellowtail snapper, Puerto Rico hogfish, and St. Croix stoplight parrotfish but highly dome-shaped selectivity for St. Thomas queen triggerfish, St. Thomas spiny lobster, and St. Croix spiny lobster (base fleet); and precise, unbiased observations (Base). Two alternative MSEs were also considered: (1) imprecise, biased observation dynamics in all 6 stocks (Alternative Observation), and (2) a moderate dome-shaped selectivity for St. Thomas queen triggerfish, St. Thomas spiny lobster, and St. Croix spiny lobster in the fleet dynamics (Alternative Fleet). Two changes in mortality were assumed in the application of the mean length estimator in the MSEs.

Four management metrics specified by the SEDAR 46 DW/AW Panel were used to examine the performance of the mean length-based MPs in the MSEs for each of the 6 stocks: the probability of not overfishing (PNOF), the probability of biomass above half B_{MSY} (B50), the probability of achieving long term yield (at least 50% of F_{MSY} yield over the latest ten projection years; LTY), and the probability of annual variability in yield to remain within 15% (AAVY). MPs were deemed suitable for calculating OFLs if PNOF > 50%, B50 > 50%, and AAVY > 50% in the base MSEs. These metrics were also compared relative to the FMSYref method which assumes perfect knowledge of F_{MSY}. The R code for the MPs and management criteria are provided in Appendix 1 and 2, respectively.

1.1. Results

All MSE simulations converged at the 1% threshold using the CheckConverg() function in DLMtool. Within each stock, the YPR_ML using $F_{0.1}$ as the F_{MSY} proxy generally performed the best, followed by SPR40_ML and SPR30_ML (Tables 1-2). From the base MSEs, the YPR_ML method met the management criteria for four of the six stocks: Puerto Rico yellowtail snapper, Puerto Rico hogfish, St. Thomas queen triggerfish, and St. Croix stoplight parrotfish. The SPR-based MPs did not perform as well using a spawning potential ratio of 30% (SPR30_ML) and 40% (SPR40_ML) and did not meet the management criteria for any of the 6 stocks considered. That SPR40 performed better than SPR30 indicates that the SPR thresholds considered may be too low and a higher SPR would be needed to meet the management criteria.

The mean length MPs in the Alternative Observation MSEs performed better as the values of the management metrics were higher compared to the base run. YPR_ML still met the management criteria for the same 4 stocks as in the base MSEs while SPR40_ML met the management criteria for Puerto Rico yellowtail snapper, Puerto Rico hogfish, and St. Thomas queen triggerfish.

Finally, the values of the performance metrics did not vary by assuming a moderate dome-shaped selectivity (Alternative Fleet) for St. Thomas queen triggerfish and St. Croix spiny lobster. The MSE for St. Thomas spiny lobster had computational issues and did not run.

2. Application of DLMtool to data

In SEDAR46-RW-01, distributions of OFLs stocks were obtained from DLMtool using real data using the YPR_ML management procedure for all 6 stocks; and SPR40_ML for St. Thomas spiny lobster and SPR30_ML for the other five stocks. Quantiles of the OFLs obtained from the MPs that met the management criteria in Section 1 are provided in Table 3.

Sensitivity analysis of the OFLs to input values in the MPs which met management criteria are shown in Figures 1-4. For all MPs for all 4 stocks considered, the most sensitive parameters were *K* and *Linf* from the von Bertalanffy growth equation, with higher OFLs prescribed for lower values. The next set of sensitive parameters were the catch and natural mortality rate, with higher OFLs prescribed for higher values. The quantiles for the OFL did not appear to vary with alternate values of von Bertalanffy parameter t_0 and length-weight parameters *a* and *b*.

3. Conclusions

- From the MSEs, the MP using $F_{0.1}$ and the mean length estimator met the management criteria and may be considered for developing OFLs for Puerto Rico yellowtail snapper, Puerto Rico hogfish, St. Thomas queen triggerfish, and St. Croix stoplight parrotfish.
- The MSEs for St. Thomas queen triggerfish and St. Croix spiny lobster suggest that performance of the mean length MPs were not significantly altered by the severity of the assumed dome-shaped selectivity pattern.
- MPs based on spawning potential ratio proxies may meet management criteria if a higher threshold for spawning potential ratio is used.

- The improved performance of the MPs relative to management criteria in the Alternative Observation MSEs (with biased and less precise observations of the data) was unexpected and requires further investigation.

Table 1. Performance metrics of the management procedures from the MSEs of the 6 Caribbean stocks: the probability of not overfishing (PNOF), the probability of biomass above half B_{MSY} (B50), the probability of achieving long term yield (at least 50% of F_{MSY} yield over the latest ten projection years; LTY), and the probability of annual variability in yield to remain within 15% (AAVY). Base stock and fleet dynamics were considered with an unbiased and biased observation dynamics.

	Base (Unbiased Observation)				Alternative (Biased Observation)			
MP	PNOF	B50	LTY	AAVY	PNOF	B50	LTY	AAVY
Yellowtail snapper								
FMSYref	91	98	100	100	90	98	100	100
YPR_ML*	54	73	69	95	71	81	46	94
SPR30_ML	15	41	60	96	40	55	52	94
SPR40_ML	40	62	68	94	59	71	47	96
Hogfish								
FMSYref	96	96	100	100	96	97	100	100
YPR_ML*	70	84	77	78	79	87	43	85
SPR30_ML	24	48	55	82	44	59	48	87
SPR40_ML	49	70	73	77	62	74	46	86
Queen triggerfish								
FMSYref	93	97	97	100	94	98	98	100
YPR_ML*	68	83	70	92	75	87	41	94
SPR30_ML	23	50	69	96	38	59	59	97
SPR40_ML	46	68	76	93	57	74	49	96
Spiny lobster STT								
FMSYref	73	94	92	100	70	93	90	100
YPR_ML	25	56	40	99	52	72	33	97
SPR30_ML	2	26	16	91	12	36	21	93
SPR40_ML	7	36	25	93	26	48	26	95
Spiny lobster STX								
FMSYref	72	91	85	100	72	90	87	100
YPR_ML	35	63	46	99	55	74	34	99
SPR30_ML	3	35	26	96	15	44	30	96
SPR40_ML	9	43	32	97	29	55	33	97
Stoplight parrotfish								
FMSYref	86	97	99	100	86	96	100	100
YPR_ML*	52	72	75	96	67	81	52	98
SPR30_ML	14	40	81	98	26	49	73	98
SPR40_ML	31	56	82	97	47	66	64	98

* Indicates the MPs which met management criteria (PNOF > 50%, B50 > 50%, and AAVY > 50%) from the MSEs using the base stock dynamics.

Table 2. Performance metrics of the management procedures from the MSEs of the 6 Caribbean stocks: the probability of not overfishing (PNOF), the probability of biomass above half B_{MSY} (B50), the probability of achieving long term yield (at least 50% of F_{MSY} yield over the latest ten projection years; LTY), and the probability of annual variability in yield to remain within 15% (AAVY). Alternate fleet dynamics were considered with a moderate dome-shaped selectivity for St. Thomas queen triggerfish and St. Croix spiny lobster. The MSE for St. Thomas spiny lobster did not run.

	Alternative Fleet (Moderate Dome)			
MP	PNOF	B50	LTY	AAVY
Queen triggerfish				
FMSYref	94	98	100	100
YPR_ML*	66	72	75	93
SPR30_ML	22	47	77	97
SPR40_ML	44	67	82	92
Spiny lobster STX				
FMSYref	70	91	100	100
YPR_ML	30	60	59	97
SPR30_ML	2	36	40	95
SPR40_ML	9	44	46	96

* Indicates the MPs which met management criteria (PNOF > 50%, B50 > 50%, and AAVY > 50%) from the base MSEs.

Table 3. Summary of the distribution of OFLs for the 6 stocks from the MPs which met management criteria for the respective stock.

	Quantile (x 1000 pounds)				
MP	Min	25%	Median	75%	Max
Yellowtail snapper					
YPR_ML	31.9	109.1	166.0	241.2	734.1
Hogfish					
YPR_ML	4.2	26.1	40.2	72.8	890.7
Queen triggerfish					
YPR_ML	4.7	13.0	18.9	30.2	189.3
Stoplight parrotfish					
YPR_ML	1.0	3.8	5.6	8.2	27.7

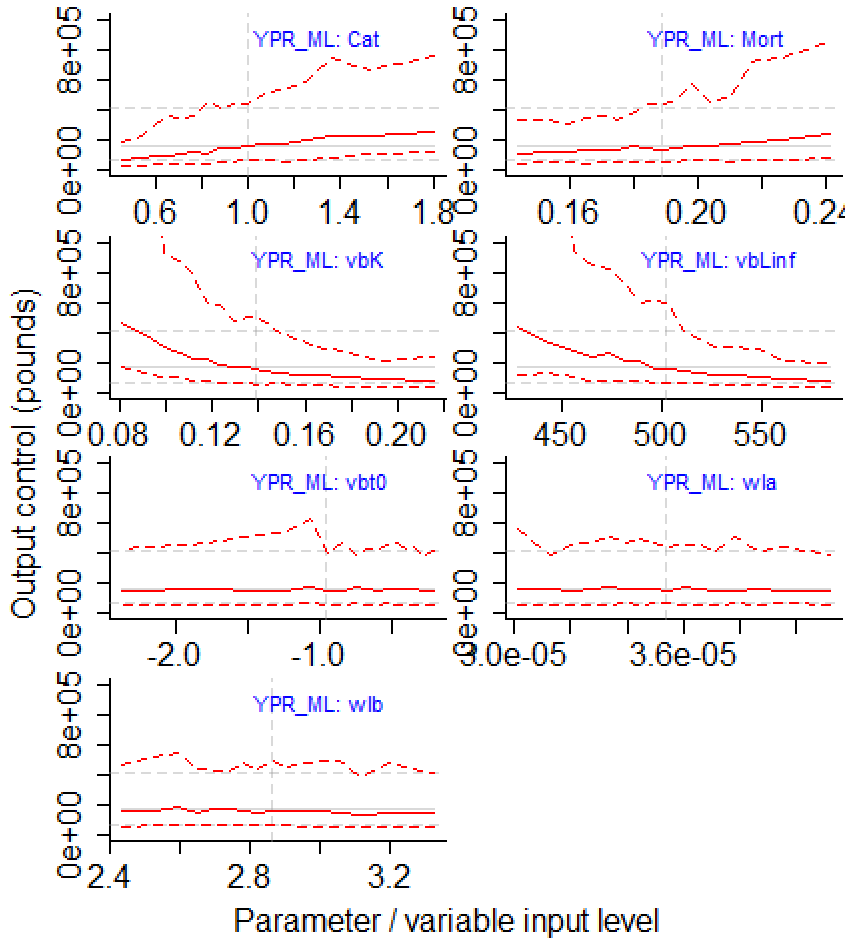

Figure 1. Sensitivity of the OFL for Puerto Rico yellowtail snapper using YPR_ML.

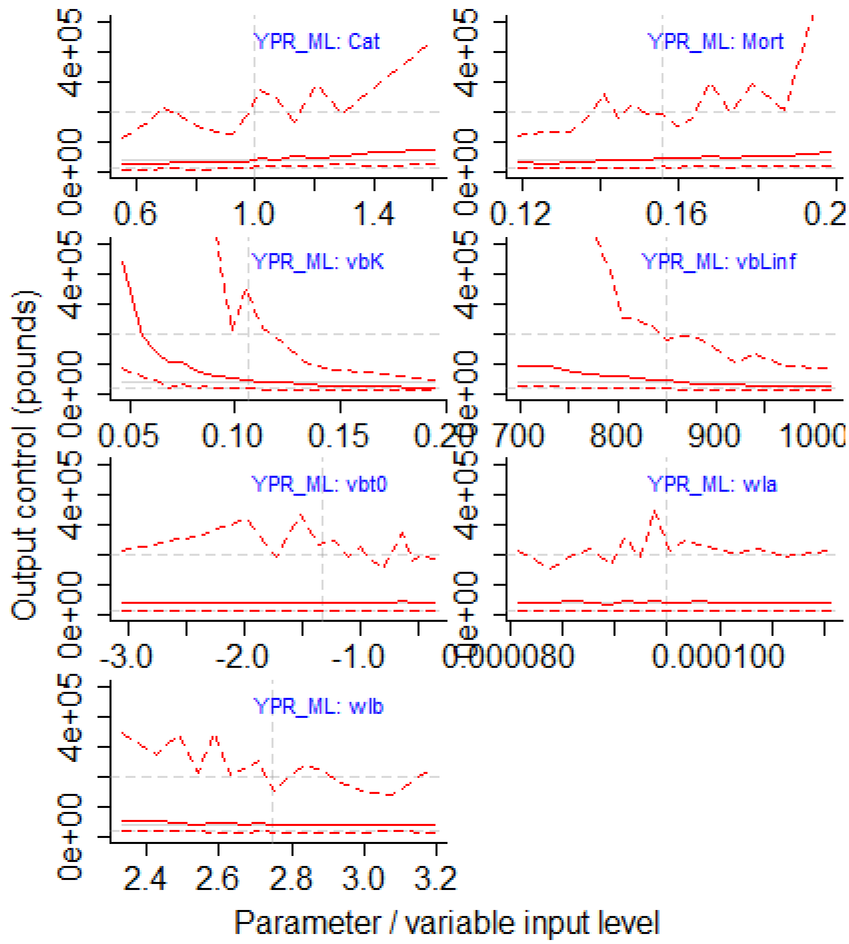

Figure 2. Sensitivity of the OFL for Puerto Rico hogfish using YPR_ML.

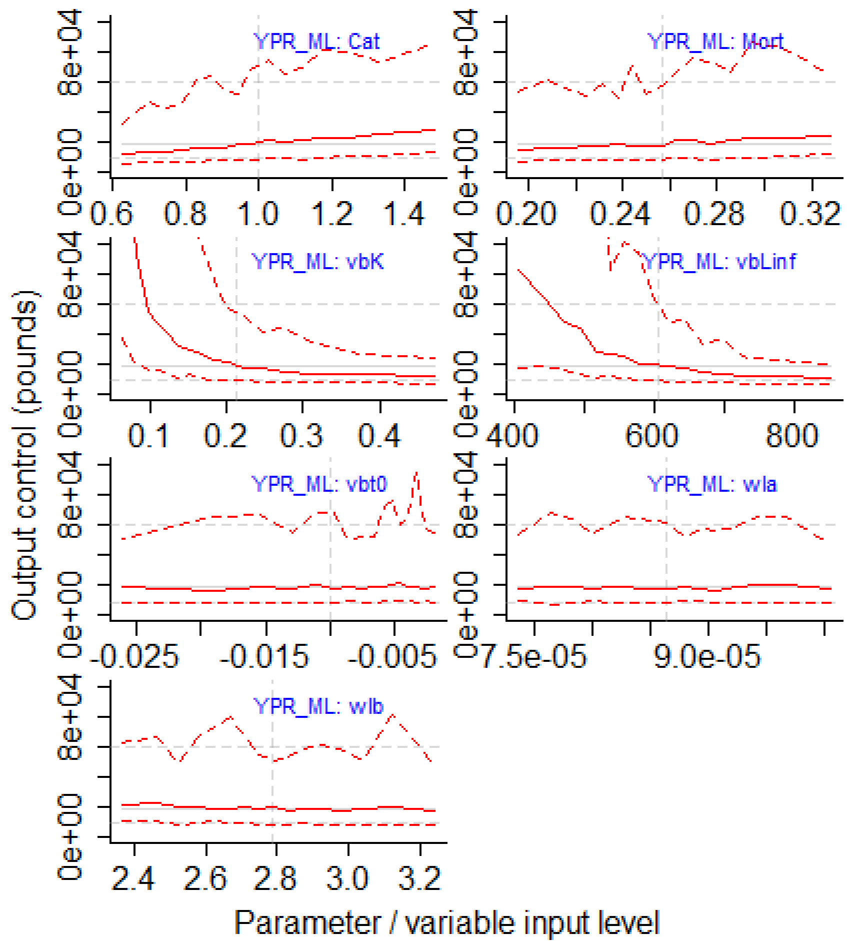

Figure 3. Sensitivity of the OFL for STT queen triggerfish using YPR_ML.

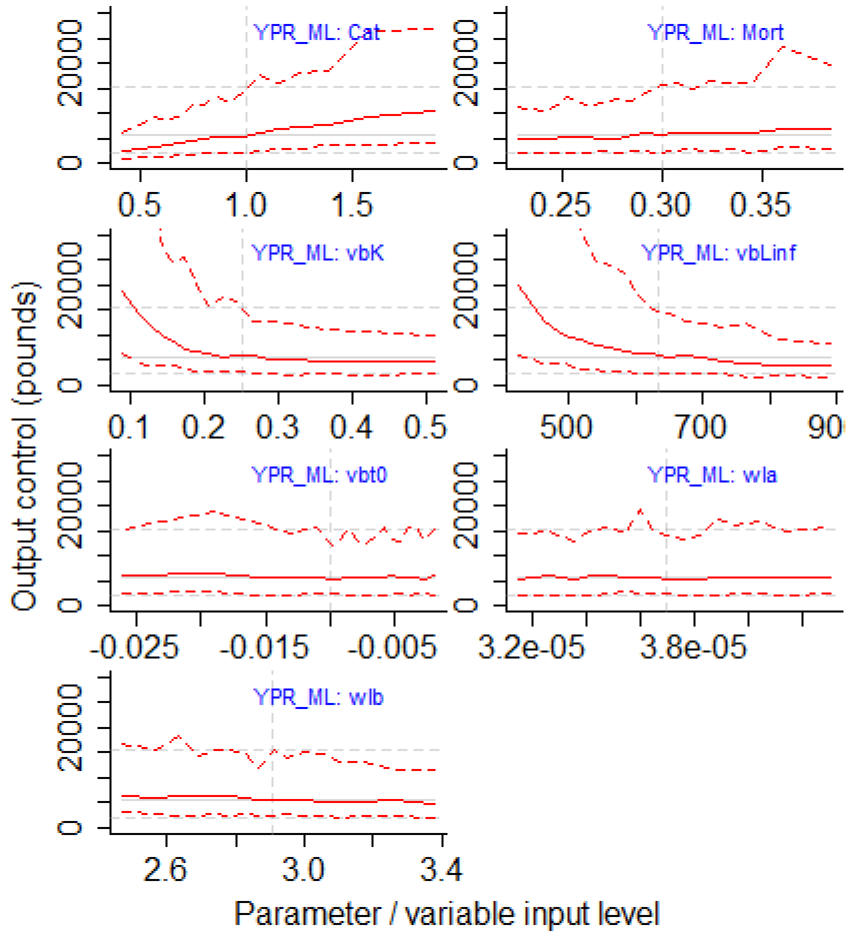

Figure 5. Sensitivity of the OFL for STX stoplight parrotfish using YPR_ML.

Appendix 1. R code for mean length MPs

```
YPR_ML<-function(x,DLM_data,reps=100){
  dependencies="DLM_data@Mort, DLM_data@CV_Mort, DLM_data@vbK,
DLM_data@CV_vbK, DLM_data@vbLinf, DLM_data@CV_vbLinf, DLM_data@vbt0,
DLM_data@CV_vbt0, DLM_data@MaxAge, DLM_data@wla, DLM_data@wlb,
DLM_data@CAL, DLM_data@Cat, DLM_data@CV_Cat"
  Mdb<-trlnorm(reps,DLM_data@Mort[x],DLM_data@CV_Mort[x])
  Linfc<-trlnorm(reps,DLM_data@vbLinf[x],DLM_data@CV_vbLinf[x])
  Kc<-trlnorm(reps,DLM_data@vbK[x],DLM_data@CV_vbK[x])
  if(DLM_data@vbt0[x]<0) t0c<--trlnorm(reps,-
DLM_data@vbt0[x],DLM_data@CV_vbt0[x])
  if(DLM_data@vbt0[x]>0) {
    t0c<-trlnorm(reps,DLM_data@vbt0[x],DLM_data@CV_vbt0[x])
    t0c[t0c>1]<-.95
  }
  Lc<-DLM_data@Lc[x]
  a<-DLM_data@wla[x]
  b<-DLM_data@wlb[x]
  MuC<-DLM_data@Cat[x,length(DLM_data@Cat[x,])]
  Cc<-trlnorm(reps,MuC,DLM_data@CV_Cat[x])
  Z<-MLne(x,DLM_data,Linfc=Linfc,Kc=Kc,Lc=Lc,ML_reps=reps,MLtype="F")
  FM<-Z-Mdb
  Ac<-Cc/(1-exp(-FM))
  FMSY<-YPRopt(Linfc,Kc,t0c,Mdb,a,b,Lc,DLM_data@MaxAge,reps)
  TAC<-Ac*FMSY
  TACfilter(TAC)
}
class(YPR_ML)<-"DLM_output"

SPR30_ML<-function(x,DLM_data,reps=100){
  dependencies="DLM_data@Mort, DLM_data@CV_Mort, DLM_data@vbK,
DLM_data@CV_vbK, DLM_data@vbLinf, DLM_data@CV_vbLinf, DLM_data@vbt0,
DLM_data@CV_vbt0, DLM_data@MaxAge, DLM_data@wla, DLM_data@wlb,
DLM_data@CAL, DLM_data@Cat, DLM_data@CV_Cat, DLM_data@L50,
DLM_data@CV_L50"
  Mdb<-trlnorm(reps,DLM_data@Mort[x],DLM_data@CV_Mort[x])
  Linfc<-trlnorm(reps,DLM_data@vbLinf[x],DLM_data@CV_vbLinf[x])
  Kc<-trlnorm(reps,DLM_data@vbK[x],DLM_data@CV_vbK[x])
  if(DLM_data@vbt0[x]<0) t0c<--trlnorm(reps,-
DLM_data@vbt0[x],DLM_data@CV_vbt0[x])
  if(DLM_data@vbt0[x]>0) {
    t0c<-trlnorm(reps,DLM_data@vbt0[x],DLM_data@CV_vbt0[x])
    t0c[t0c>1]<-.95
  }
  Lc<-DLM_data@Lc[x]
  a<-DLM_data@wla[x]
  b<-DLM_data@wlb[x]
  MuC<-DLM_data@Cat[x,length(DLM_data@Cat[x,])]
  Cc<-trlnorm(reps,MuC,DLM_data@CV_Cat[x])
  L50c<-trlnorm(reps,DLM_data@L50[x],DLM_data@CV_L50[x])
  Z<-MLne(x,DLM_data,Linfc=Linfc,Kc=Kc,Lc=Lc,ML_reps=reps,MLtype="F")
```

```
  FM<-Z-Mdb
  Ac<-Cc/(1-exp(-FM))
  FMSY<-SPR30opt(Linfc,Kc,t0c,Mdb,a,b,Lc,DLM_data@MaxAge,L50c,reps)
  TAC<-Ac*FMSY
  TACfilter(TAC)
}
class(SPR30_ML)<-"DLM_output"

SPR40_ML<-function(x,DLM_data,reps=100){
  dependencies="DLM_data@Mort, DLM_data@CV_Mort, DLM_data@vbK,
DLM_data@CV_vbK, DLM_data@vbLinf, DLM_data@CV_vbLinf, DLM_data@vbt0,
DLM_data@CV_vbt0, DLM_data@MaxAge, DLM_data@wla, DLM_data@wlb,
DLM_data@CAL, DLM_data@Cat, DLM_data@CV_Cat, DLM_data@L50,
DLM_data@CV_L50"
  Mdb<-trlnorm(reps,DLM_data@Mort[x],DLM_data@CV_Mort[x])
  Linfc<-trlnorm(reps,DLM_data@vbLinf[x],DLM_data@CV_vbLinf[x])
  Kc<-trlnorm(reps,DLM_data@vbK[x],DLM_data@CV_vbK[x])
  if(DLM_data@vbt0[x]<0) t0c<--trlnorm(reps,-
DLM_data@vbt0[x],DLM_data@CV_vbt0[x])
  if(DLM_data@vbt0[x]>0) {
    t0c<-trlnorm(reps,DLM_data@vbt0[x],DLM_data@CV_vbt0[x])
    t0c[t0c>1]<-.95
  }
  Lc<-DLM_data@Lc[x]
  a<-DLM_data@wla[x]
  b<-DLM_data@wlb[x]
  MuC<-DLM_data@Cat[x,length(DLM_data@Cat[x,])]
  Cc<-trlnorm(reps,MuC,DLM_data@CV_Cat[x])
  L50c<-trlnorm(reps,DLM_data@L50[x],DLM_data@CV_L50[x])
  Z<-MLne(x,DLM_data,Linfc=Linfc,Kc=Kc,Lc=Lc,ML_reps=reps,MLtype="F")
  FM<-Z-Mdb
  Ac<-Cc/(1-exp(-FM))
  FMSY<-SPR40opt(Linfc,Kc,t0c,Mdb,a,b,Lc,DLM_data@MaxAge,L50c,reps)
  TAC<-Ac*FMSY
  TACfilter(TAC)
}
class(SPR40_ML)<-"DLM_output"

YPRopt<-function(Linfc,Kc,t0c,Mdb,a,b,Lc,maxage,reps=100) {

  nf<-200
  frates<-seq(0,3,length.out=nf)
  rat<-Lc/Linfc
  rat[rat>0.8]<-0.8     # need to robustify this for occasionally very
high samples of LFS
  tc=log(1-rat)/-Kc+t0c
  tc=round(tc,0)
  tc[tc<1]<-1
  tc[tc>maxage]<-maxage

  vul<-array(0,dim=c(reps,maxage))
```

```
  lx<-array(NA,dim=c(reps,maxage))

  ypr<-array(NA,dim=c(reps,nf))

  #average weight at age - follow von Bertalanffy growth
  age<-array(rep(1:maxage,each=reps),dim=c(reps,maxage))
  la<-Linfc*(1-exp(-Kc*((age-t0c))))
  wa<-a*la^b

  #vulnerability schedule - assumes knife-edge vulnerability, where
all individuals age tc to maxage are fully vulnerbale
  #all individulas less than age tc are not vulnerable
  for(i in 1:reps) if(tc[i]>0)vul[i,tc[i]:maxage]<-1

  lx[,1]<-1
  for(k in 1:nf){
    for(i in 2:maxage){
      lx[,i]=lx[,i-1]*exp(-(Mdb+vul[,i-1]*frates[k]))
    }
    phi_vb=apply(lx*wa*vul,1,sum)

    ypr[,k]=(1-exp(-frates[k]))*phi_vb
  }

  # More code that derived F0.1 in 'per recruit analysis.R' (Meaghan
Bryan)
  slope.origin=(ypr[,2]-ypr[,1])/(frates[2]-frates[1])
  slope.10=round(0.1*slope.origin,2)

  slope=array(NA,dim=dim(ypr))#vector(length=length(ypr))
  slope[,1]=slope.origin
  for(i in 3:ncol(ypr))
  {
    slope[,i-1]=round((ypr[,i]-ypr[,i-1])/(frates[i]-frates[i-1]),2)
  }
  dif=abs(slope-slope.10)
  dif[is.na(dif)]<-10e10

  frates[apply(dif,1,which.min)]#frates[which.min(dif)]

}

SPR30opt<-function(Linfc,Kc,t0c,Mdb,a,b,Lc,maxage,L50,reps=100) {

  nf<-200
  frates<-seq(0,3,length.out=nf)
  rat<-Lc/Linfc
  rat[rat>0.8]<-0.8     # need to robustify this for occasionally very
high samples of LFS
  tc=log(1-rat)/-Kc+t0c
  tc=round(tc,0)
  tc[tc<1]<-1
```

```
  tc[tc>maxage]<-maxage

  tmat<-iVB(t0c,Kc,Linfc,L50)
  tmat[is.nan(tmat)]<-maxage
  tmat<-round(tmat,0)
  tmat[tmat<1]<-1
  tmat[tmat>maxage]<-maxage

  vul<-array(0,dim=c(reps,maxage))
  mat<-array(0,dim=c(reps,maxage))
  lx<-array(NA,dim=c(reps,maxage))

  sbpr<-array(NA,dim=c(reps,nf))

  #average weight at age - follow von Bertalanffy growth
  age<-array(rep(1:maxage,each=reps),dim=c(reps,maxage))
  la<-Linfc*(1-exp(-Kc*((age-t0c))))
  wa<-a*la^b

  #vulnerability schedule - assumes knife-edge vulnerability, where
all individuals age tc to maxage are fully vulnerbale
  #all individulas less than age tc are not vulnerable
  for(i in 1:reps){
    if(tc[i]>0)vul[i,tc[i]:maxage]<-1
    if(tmat[i]>0)mat[i,tmat[i]:maxage]<-1
  }

  lx[,1]<-1
  for(k in 1:nf){
    for(i in 2:maxage){
      lx[,i]=lx[,i-1]*exp(-(Mdb+vul[,i-1]*frates[k]))
    }

    sbpr[,k]=apply(lx*wa*mat,1,sum)
  }

  sbpr.ratio<-sbpr/sbpr[,1]
  sbpr.dif<-abs(sbpr.ratio-0.3)

  sbpr.dif[is.na(sbpr.dif)]<-10e10

  frates[apply(sbpr.dif,1,which.min)]

}

SPR40opt<-function(Linfc,Kc,t0c,Mdb,a,b,Lc,maxage,L50,reps=100) {

  nf<-200
  frates<-seq(0,3,length.out=nf)
  rat<-Lc/Linfc
  rat[rat>0.8]<-0.8     # need to robustify this for occasionally very
high samples of LFS
```

```
  tc=log(1-rat)/-Kc+t0c
  tc=round(tc,0)
  tc[tc<1]<-1
  tc[tc>maxage]<-maxage

  tmat<-iVB(t0c,Kc,Linfc,L50)
  tmat[is.nan(tmat)]<-maxage
  tmat<-round(tmat,0)
  tmat[tmat<1]<-1
  tmat[tmat>maxage]<-maxage

  vul<-array(0,dim=c(reps,maxage))
  mat<-array(0,dim=c(reps,maxage))
  lx<-array(NA,dim=c(reps,maxage))

  sbpr<-array(NA,dim=c(reps,nf))

  #average weight at age - follow von Bertalanffy growth
  age<-array(rep(1:maxage,each=reps),dim=c(reps,maxage))
  la<-Linfc*(1-exp(-Kc*((age-t0c))))
  wa<-a*la^b

  #vulnerability schedule - assumes knife-edge vulnerability, where
all individuals age tc to maxage are fully vulnerbale
  #all individulas less than age tc are not vulnerable
  for(i in 1:reps){
    if(tc[i]>0)vul[i,tc[i]:maxage]<-1
    if(tmat[i]>0)mat[i,tmat[i]:maxage]<-1
  }

  lx[,1]<-1
  for(k in 1:nf){
    for(i in 2:maxage){
      lx[,i]=lx[,i-1]*exp(-(Mdb+vul[,i-1]*frates[k]))
    }

    sbpr[,k]=apply(lx*wa*mat,1,sum)
  }

  sbpr.ratio<-sbpr/sbpr[,1]
  sbpr.dif<-abs(sbpr.ratio-0.4)

  sbpr.dif[is.na(sbpr.dif)]<-10e10

  frates[apply(sbpr.dif,1,which.min)]

}

MLne<-function(x,DLM_data,Linfc,Kc,Lc,ML_reps=100,MLtype="F"){
  year<-1:dim(DLM_data@CAL)[2]
  nlbin<-ncol(DLM_data@CAL[x,,])
  nlyr<-nrow(DLM_data@CAL[x,,])
```

```
  mlbin<-(DLM_data@CAL_bins[1:nlbin]+DLM_data@CAL_bins[2:(nlbin+1)])/2
  nbreaks<-2
  Z<-matrix(NA,nrow=ML_reps,ncol=nbreaks+1)
  for(i in 1:ML_reps){
    mlen<-rep(NA,length(year))
    for(y in 1:length(year)) {
      lsamp<-
sample(mlbin,ceiling(sum(DLM_data@CAL[x,y,])/2),replace=T,prob=DLM_dat
a@CAL[x,y,])
      mlen[y]<-mean(lsamp[lsamp>=Lc])
    }
    ss<-ceiling(apply(DLM_data@CAL[x,,],1,sum)/2)
    getZ<-
try(bhnoneq(year=year,mlen=mlen,ss=ss,K=Kc[i],Linf=Linfc[i],Lc=Lc,nbre
aks=nbreaks,

styrs=ceiling(length(year)*((1:nbreaks)/(nbreaks+1))),stZ=rep(0.5,nbre
aks+1)))
    if(class(getZ)=="try-error")Z[i,]<-rep(NA,nbreaks+1)
    else Z[i,] <- getZ
  }
  if(MLtype=="F")return(Z[,ncol(Z)])
  if(MLtype=="dep")return(Z[,c(1,ncol(Z))])
}

bhnoneq<-function(year,mlen,ss,K,Linf,Lc,nbreaks,styrs,stZ) {
  mlen[mlen<=0|is.na(mlen)]<--99
  ss[ss<=0|is.na(ss)]<-0
  stpar<-c(stZ,styrs)
  results <-
optim(stpar,bh_LL,method="BFGS",year=year,Lbar=mlen,ss=ss,
                   nbreaks=nbreaks,K=K,Linf=Linf,Lc=Lc,hessian=F)
  return(results$par[1:(nbreaks+1)])
}

#include <Rcpp.h>
using namespace Rcpp;
// [[Rcpp::export]]
double bh_LL(NumericVector stpar, NumericVector year, NumericVector
Lbar, NumericVector ss,
  double Linf, double K, double Lc, int nbreaks) {

  int i;
  int j;
  int m;

  int count = year.size()-1;
  int nbr = nbreaks-1;

  NumericVector Z(nbreaks+1);
  NumericVector ggyr(nbreaks);
  NumericMatrix dy(nbreaks,count+1);
```

```
  NumericMatrix a(nbreaks+1,count+1);
  NumericMatrix s(nbreaks+1,count+1);
  NumericMatrix r(nbreaks+1,count+1);
  NumericVector denom(count+1);
  NumericVector numsum(count+1);
  NumericVector num(count+1);
  NumericVector Lpred(count+1);

  double sum_square_Lpred = 0.;
  double nyear = 0.;
  double sigma;
  double nLL;

  for(i=0;i<=nbr;++i) {
    Z[i] = stpar[i];
    ggyr[i] = stpar[nbr+2+i];
  }
  Z[nbr+1] = stpar[nbr+1];

  for(i=0;i<=nbr;++i) {
    for(m=1;m<=count+1;++m) {
      if(ggyr[i]>=m) dy(i,m-1) = 0.;
      else dy(i,m-1) = m-ggyr[i];
    }
  }
  if(nbreaks>1) {
    for(i=0;i<nbr;++i){
      for(m=0;m<=count;++m) dy(i,m) -= dy(i+1,m);
    }
  }

  for(m=0;m<=count;++m) {
    denom[m] = 0.;
    numsum[m] = 0.;

    for(i=0;i<=nbr+1;i++) {
      a(i,m) = 1.;
      r(i,m) = 1.;

      if(i<nbr+1) s(i,m) = 1. - exp(-(Z[nbr+1-i]+K) * dy(nbr-i,m));
      if(i==nbr+1) s(i,m) = 1.;

      if(i>0) {
        for(j=0;j<=i-1;++j) {
          a(i,m) *= exp(-Z[nbr+1-j] * dy(nbr-j,m));
          r(i,m) *= exp(-(Z[nbr+1-j] + K) * dy(nbr-j,m));
        }
      }

      if(i<=nbr) denom[m] += a(i,m)*(1. - exp(-Z[nbr+1-i] * dy(nbr-
i,m)))/Z[nbr+1-i];
```

```
      if(i==nbr+1) denom[m] += a(i,m)/Z[nbr+1-i];

      numsum(m) += r(i,m) * s(i,m) / (Z[nbr+1-i] + K);
    }
  }
  num = Linf * (denom - (1. - Lc/Linf) * numsum);
  Lpred = num/denom;
  for(m=0;m<=count;++m) {
    if(ss[m]>0) {
      sum_square_Lpred += ss[m] * pow(Lbar[m]-Lpred[m],2.);
      nyear += 1.;
    }
  }
  sigma = sqrt(sum_square_Lpred/nyear);
  nLL = -nyear * log(sigma) - 0.5 * sum_square_Lpred/(sigma*sigma);
  nLL *= -1;
  return nLL;
}
```

Appendix 2. R code for management criteria

```
.AAVY <- function(MSE) {
  Cat <- MSE@C
  np <- dim(Cat)[3]
  meanCat <- apply(Cat,1:2,mean)

  accum <- array(0,dim=dim(meanCat))
  for(i in 1:(np-1)) accum <- accum + abs(Cat[,,i+1] - Cat[,,i])
  result <- (np+1)*accum/(np*meanCat)

  apply(result, 2, function(x)
sum(x<15,na.rm=T))/apply(result,2,function(x) sum(!is.na(x)))

}

.LTY <- function(MSE) {
  Cat <- MSE@C
  ind <- max(1,dim(Cat)[3]-9):dim(Cat)[3]
  Cat <- Cat[,,ind]

  RefY <- 0.5 * MSE@OM$RefY

  x <- Cat>RefY
  y <- apply(x, 1:2, function(x) any(x==TRUE))
  apply(y, 2, sum, na.rm=T)/apply(y, 2, function(x) sum(!is.na(x)))
}

S46_metrics <- function(MSE) {
  x <- summary(MSE)
  PNOF <- 100 - x$POF
  PB50 <- 100 - x$P50
  LTY <- .LTY(MSE)*100
  AAVY <- .AAVY(MSE)*100

  data.frame(x$MP, PNOF = PNOF, PB50 = PB50, LTY = LTY, AAVY = AAVY)
}
```