A new search approach to improve the accuracy of stock assessment forecasts

GMFMC SSC meeting, Sept 27, 2021

Nathan Vaughan

Vaughan Analytics in support of SEFSC, NOAA, Miami, FL

Stock assessment forecasting: Achieving multiple objectives

1. Achieve target benchmarks for yield or stock status such as MSY or SPR30%

1. Estimate the F that achieves benchmarks at equilibrium (F_{OFL})

1. Project fishery at $F = F_{OFL}$ (or an alternative target) in every year

1. Project with catch fractions between fishing sectors equal to regulated allocation fractions.

1. Project with annual fleet specific effort held constant between fleets within allocated fishing sectors.

Stock assessment forecasting: Achieving multiple objectives

SSB	2471	2652	2813	…	3179
SSB/SSB_{TARGET}	0.78	0.83	0.88	…	1
F_{TOTAL}	0.242	0.242	0.242	…	0.242
$Yield_{TOTAL}$	953	1014	1063	…	1162
$Yield_{S1}$	257	274	287	…	314
$Yield_{S2}$	696	740	776	…	848
F_1	0.177	0.173	0.171	…	0.167
F_2	0.018	0.018	0.017	…	0.017
F_3	0.558	0.548	0.543	…	0.539
F_4	0.010	0.010	0.010	…	0.010
	$Year_1$ 2022	$Year_2$ 2023	$Year_3$ 2024	…	$Year_{100}$ 2119

NOAA FISHERIES

Stock assessment forecasting: Influential assumptions

1. Future recruitment patterns define stock productivity and variability (Recent mean, S/R curve, deviations)

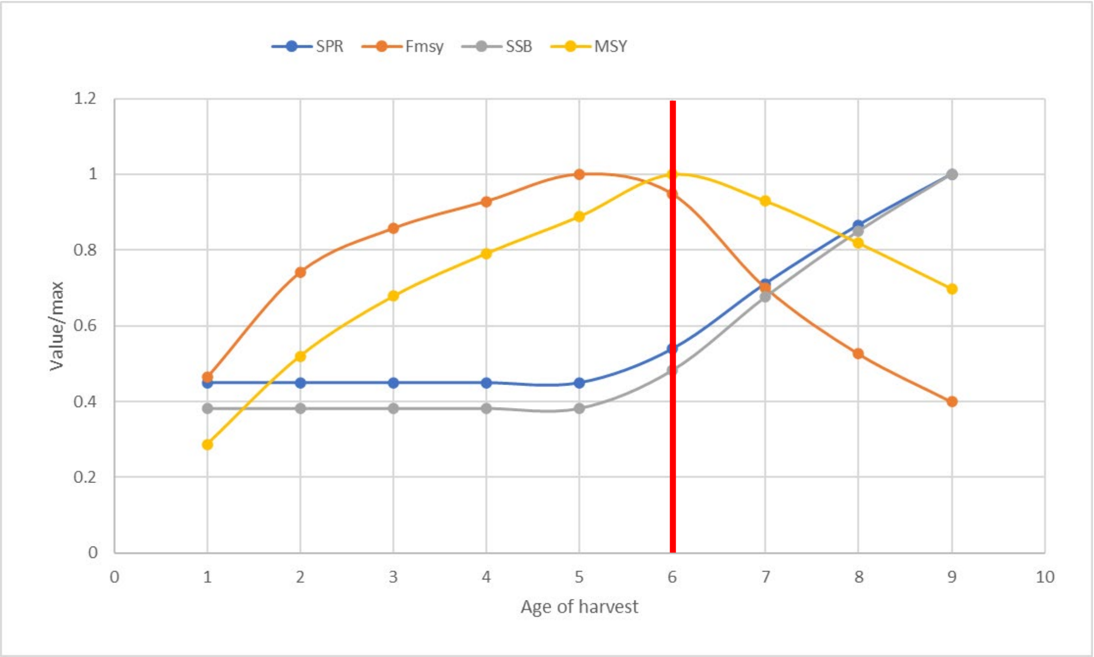
1. Fleet selectivity and retention functions (recent or upcoming size limit regulations)

1. Fishing sector allocations (adjusting catch fraction between fleets adjusts the aggregate fishery selectivity)

1. Benchmark targets SSB/SSB0 or MSY vs SPR or Fmax (Raw SSB or MSY are intuitive but can produce variable F_{OFL} results while SPR and Fmax produce stable F results but could have unintended impacts)

NOAA FISHERIES

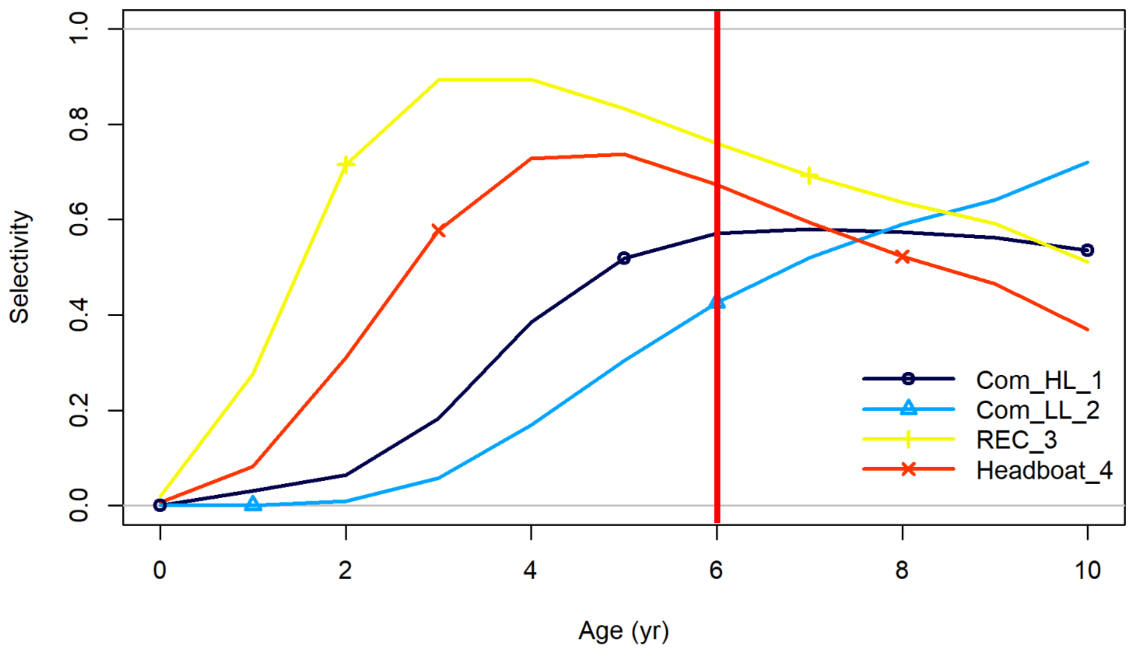
Benchmark dependence on forecast assumptions – Global MSY



- Global MSY search example
- SPR, F_{MSY}, SSB, MSY achieved when only a single age class is harvested
- In practice fisheries represent a weighted average of these results based on fleet selectivity and allocations

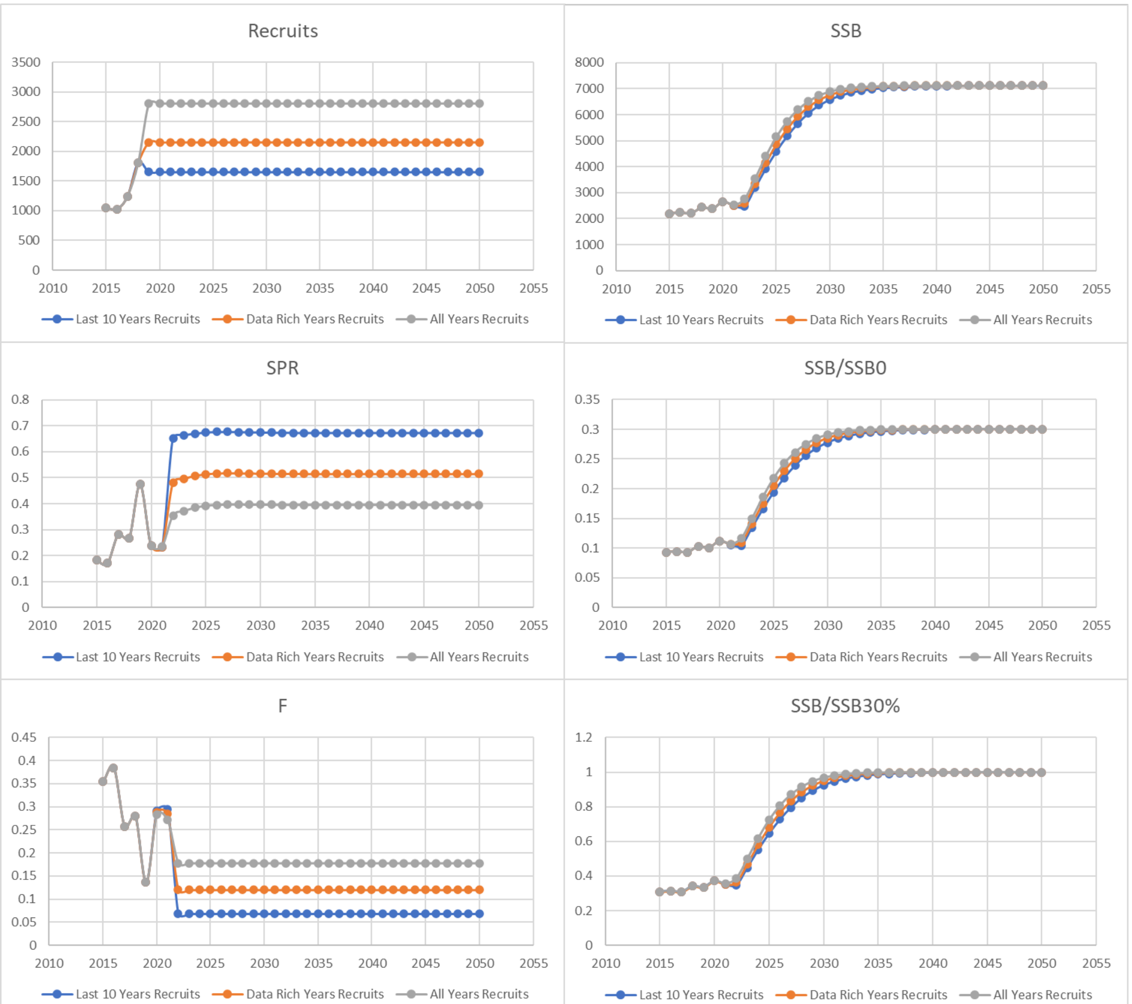
NOAA FISHERIES

Benchmark dependence on forecast assumptions – Fleet Selectivity



- As seen in global MSY calculations, the age/size of capture impacts the sustainable yield of a fishery
- Due to the variability in selectivity between fleets, allocation fractions, sector specific closures, size limits, and discard mortality will often impact benchmark values

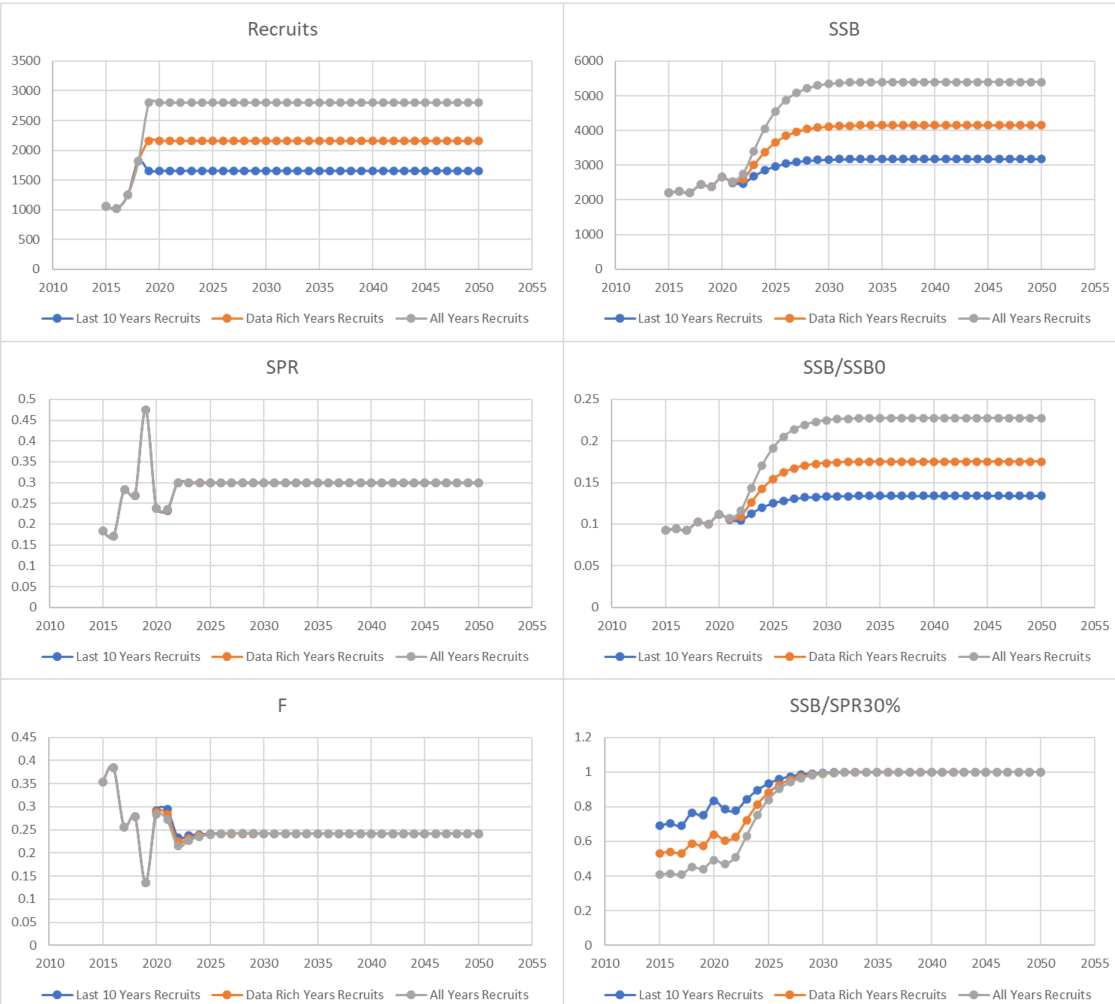
Benchmark dependence on forecast assumptions – Recruitment SSB30%



- Recruitment/productivity assumptions can impact benchmarks (e.g. MSY, SSB_{msy}, F_{msy})
- May impact current overfished and overfishing status determinations
- An SSB% benchmark will achieve variable F_{msy} (overfishing) but stable SSB_{msy} (overfished) determinations

NOAA FISHERIES

Benchmark dependence on forecast assumptions – Recruitment SPR30%



- Recruitment/productivity assumptions can impact benchmarks (e.g. MSY, SSB_{msy}, F_{msy})
- May impact current overfished and overfishing status determinations
- An SPR% benchmark will achieve stable F_{msy} (overfishing) but varying SSB_{msy} (overfished) determinations

NOAA FISHERIES

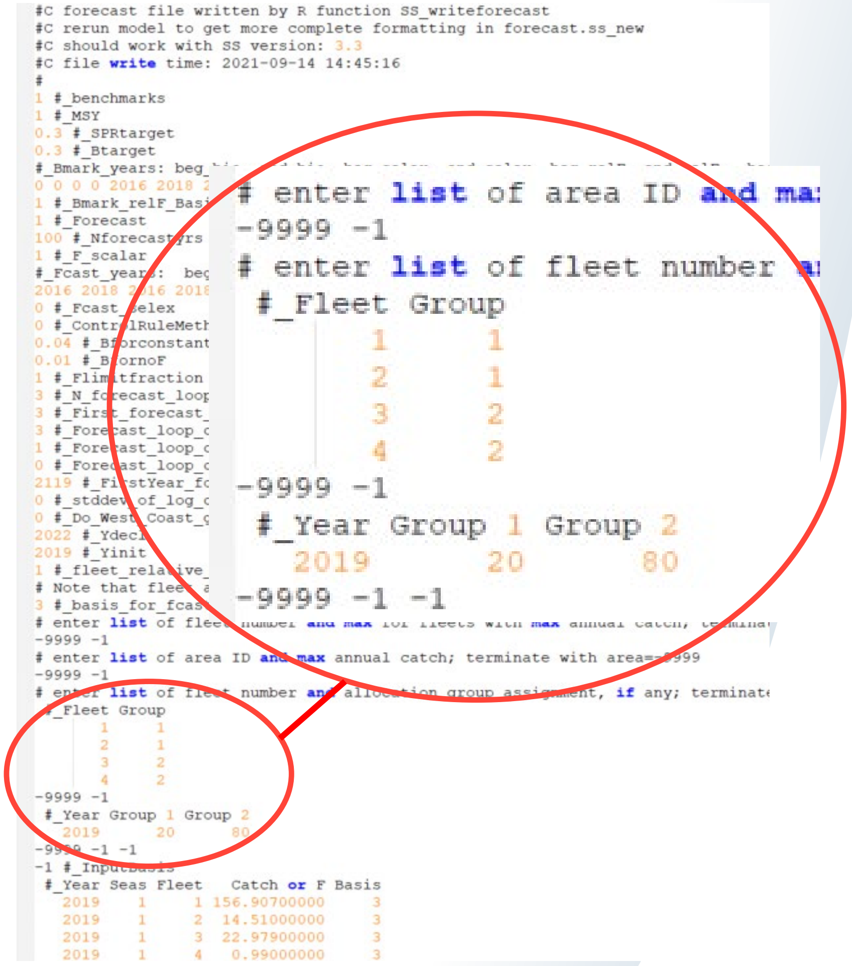
Default Stock Synthesis approach to allocations

- Project with annual fleet specific effort held constant between all managed fleets in the fishery
- Annual total effort scaled to achieve F= F_{ofl} and benchmark targets at equilibrium
- Fleet specific catch is calculated and sector effort is adjusted to achieve target catch allocations
- Single pass adjustment that does not account for the impact of fleet selectivity
- Final equilibrium results may not achieve the target benchmark and/or the target annual F

```
#C forecast file written by R function SS_writeforecast
#C rerun model to get more complete formatting in forecast.ss_new
#C should work with SS version: 3.3
#C file write time: 2021-09-14 14:45:16
#
1 #_benchmarks
1 #_MSY
0.3 #_SPRtarget
0.3 #_Btarget
#_Bmark_years: beg_bio, end_bio, beg_selex, end_selex, beg_relF, end_relF,  beg
0 0 0 0 2016 2018 2009 2018 0 0
1 #_Bmark_relF_Basis
1 #_Forecast
100 #_Nforecastyrs
1 #_F_scalar
#_Fcast_years:  beg_selex, end_selex, beg_relF, end_relF, beg_recruits, end_rec
2016 2018 2016 2018 1984 2018
0 #_Fcast_selex
0 #_ControlRuleMethod
0.04 #_BforconstantF
0.01 #_BfornoF
1 #_Flimitfraction
3 #_N_forecast_loops
3 #_First_forecast_loop_with_stochastic_recruitment
3 #_Forecast_loop_control_3
1 #_Forecast_loop_control_4
0 #_Forecast_loop_control_5
2119 #_FirstYear_for_caps_and_allocations
0 #_stddev_of_log_catch_ratio
0 #_Do_West_Coast_gfish_rebuilder_output
2022 #_Ydecl
2019 #_Yinit
1 #_fleet_relative_F
# Note that fleet allocation is used directly as average F if Do_Forecast=4
3 #_basis_for_fcast_catch_tuning
# enter list of fleet number and max for fleets with max annual catch; terminat
-9999 -1
# enter list of area ID and max annual catch; terminate with area=-9999
-9999 -1
# enter list of fleet number and allocation group assignment, if any; terminate
 #_Fleet Group
      1     1
      2     1
      3     2
      4     2
-9999 -1
 #_Year Group 1 Group 2
   2019      20      80
-9999 -1 -1
-1 #_InputBasis
 #_Year Seas Fleet   Catch or F Basis
   2019    1     1 156.90700000     3
   2019    1     2  14.51000000     3
   2019    1     3  22.97900000     3
   2019    1     4   0.99000000     3
```

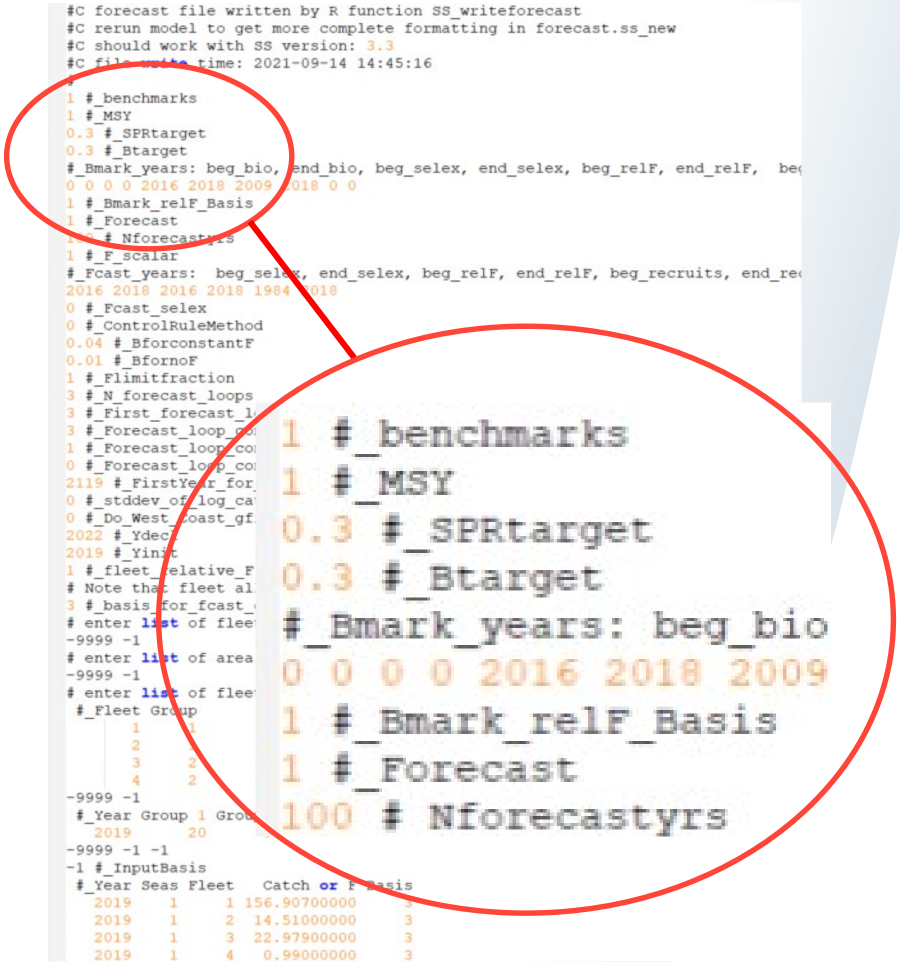
Default Stock Synthesis approach to allocations

- Project with annual fleet specific effort held constant between all managed fleets in the fishery
- Annual total effort scaled to achieve $F = F_{ofl}$ and benchmark targets at equilibrium
- Fleet specific catch is calculated and sector effort is adjusted to achieve target catch allocations
- Single pass adjustment that does not account for the impact of fleet selectivity
- Final equilibrium results may not achieve the target benchmark and/or the target annual F



Previous SEFSC approach to achieve benchmark target

- Iterative search for the target benchmark on top of the base SS allocation adjustment
- Adjusts the target benchmark input to SS until the achieved benchmark is equal to the true target benchmark
- Achieves target benchmarks only for SSB% or SPR% proxies
- MSY or F_{max}(MSY per recruit) benchmarks can not be achieved with this approach
- Does not ensure that annual F targets are achieved



New approach to achieve multiple forecasting targets - inputs

- Uses SS capacity to input fixed Fleet/Year specific catch/F values
- Iteratively adjusts fleet specific annual F for 100 years of projection to achieve all forecasting targets
- Achieves benchmark target, annual F targets, allocation targets, and relative effort targets simultaneously
- Functional for all benchmark targets (SSB%, SPR%, MSY, F_{max})
- Added capacity to automate OFL, ABC, and $F_{rebuild}$ calculations that conform to all targets
- Uses three independent scaling functions to adjust F values

```
#C forecast file written by R f
#C rerun model to get more comp
#C should work with SS version:
#C file write time: 2021-09-14
#
1 #_benchmarks
1 #_MSY
0.3 #_SPRtarget
0.3 #_Btarget
#_Bmark_years: beg_bio, end_bio
0 0 0 0 2016 2018 2009 2018 0 0
1 #_Bmark_relF_Basis
1 #_Forecast
100 #_Nforecastyrs
1 #_F_scalar
#_Fcast_years:  beg_selex, end_
2016 2018 2016 2018 1984 2018
0 #_Fcast_selex
0 #_ControlRuleMethod
0.04 #_BforconstantF
0.01 #_BfornoF
1 #_Flimitfraction
3 #_N_forecast_loops
3 #_First_forecast_loop_with_st
3 #_Forecast_loop_control_3
1 #_Forecast_loop_control_4
0 #_Forecast_loop_control_5
2119 #_FirstYear_for_caps_and_a
0 #_stddev_of_log_catch_ratio
0 #_Do_West_Coast_gfish_rebuild
2022 #_Ydecl
2019 #_Yinit
1 #_fleet_relative_F
# Note that fleet allocation is
3 #_basis_for_fcast_catch_tunin
# enter list of fleet number an
-9999 -1
# enter list of area ID and max
-9999 -1
# enter list of fleet number an
 #_Fleet Group
      1     1
      2     1
      3     2
      4     2
-9999 -1
 #_Year Group 1 Group 2
   2019        20        80
-9999 -1 -1
-1 #_InputBasis
 #_Year Seas Fleet    Catch or F Basis
   2019     1      1 156.90700000      3
   2019     1      2  14.51000000      3
   2019     1      3  22.97900000      3
   2019     1      4   0.99000000      3
```

```
-1 #_InputBasis
#_Year Seas Fleet    Catch or F Basis
 2019     1      1 156.90700000      3
 2019     1      2  14.51000000      3
 2019     1      3  22.97900000      3
 2019     1      4   0.99000000      3
 2020     1      1 184.01000000      3
 2020     1      2  11.89100000      3
 2020     1      3  66.11500000      3
 2020     1      4   1.37700000      3
 2021     1      1 184.01000000      3
 2021     1      2  11.89100000      3
 2021     1      3  66.11500000      3
 2021     1      4   1.37700000      3
 2022     1      1   0.13281858     99
 2022     1      2   0.01355791     99
 2022     1      3   0.61455346     99
 2022     1      4   0.01115655     99
 2023     1      1   0.12842608     99
 2023     1      2   0.01310953     99
 2023     1      3   0.59430693     99
 2023     1      4   0.01078899     99
 2024     1      1   0.12487318     99
 2024     1      2   0.01274685     99
 2024     1      3   0.58115572     99
 2024     1      4   0.01055025     99
 2025     1      1   0.12268673     99
 2025     1      2   0.01252367     99
 2025     1      3   0.57397653     99
 2025     1      4   0.01041992     99
```


NOAA FISHERIES

New approach to achieve multiple forecasting targets – F scaling

1) An equilibrium benchmark scaler which applies a single scalar multiplier to every F

```
#Calculate depletion target adjustment scale depending on the specified target (SPR ra
if(Forecast_target==1){
  search_step<-0.00001
  Target.Depletion <- forecast[["SPRtarget"]]
  Depletion<-SPRfit$SPR

  Achieved.Depletion <- median(Depletion[(length(Depletion)-29):length(Depletion)])
  DepletionScale <- (1-Target.Depletion)/(1-Achieved.Depletion)

  DepletionScale <- (-log(1-((1-exp(-FScale))*DepletionScale))/FScale)

  Depletion_R<-TimeFit3$SpawnBio/Virgin_bio
  Target.Rebuild <- mean(Depletion_R[(length(Depletion_R)-9):length(Depletion_R)])

}else if(Forecast_target==2){
  Depletion <- TimeFit3$SpawnBio/Virgin_bio
```

New approach to achieve multiple forecasting targets – F scaling

2) An annual F scaler that applies a year specific multiplier to all fleets in each year

```
#Fmult2 calculations define the multiplier for adjusting annual F values
#Zero catch years are identified first to prevent divide by zero errors in the scaling and
#to tell the search algorithm that the target has been achieved
zero_catch <- which(SPRfit$F_report[sort(rep(seq_along(SPRfit$F_report),length(seasons)*length(F_cols)))]==0)
if(length(zero_catch)>0){
  if(FScale==0){
    Fmult2[zero_catch] <- 1
  }else{
    Fmult2[zero_catch] <- 2
  }
  Fmult2[-zero_catch] <- FScale/SPRfit$F_report[sort(rep(seq_along(SPRfit$F_report),length(seasons)*length(F_cols)))][-zero_catch]
}else{
  Fmult2 <- FScale/SPRfit$F_report[sort(rep(seq_along(SPRfit$F_report),length(seasons)*length(F_cols)))]
}
#If in a rebuild search phase the rebuild years are now adjusted independently of the later F_OFL years
if(fitting_Rebuild==TRUE){
  Fmult2[rebuild_ref] <- Rebuild.Scale/SPRfit$F_report[sort(rep(seq_along(SPRfit$F_report),length(seasons)*length(F_cols)))][rebuild_ref]
}
```

New approach to achieve multiple forecasting targets – F scaling

3) An annual allocation scaler that applies a year and sector specific multiplier to each fleet within a sector and year

```
#Here the achieved catch fractions by fishing sector and year are calculated and compared relative
#to the target allocations. An adjustment multiplier is then computed to adjust fleet Fs closer to a
#value expected to achieve the target allocations.
if(FScale > 0){
  if(n_groups>0){
    Catch_temp <- TimeFit3[,Catch_cols3]
    Catch_tot <- apply(Catch_temp[,which(groups!=0)],1,sum)
    for(i in 1:n_groups){
      sort.mat <- matrix(NA, nrow = 100*length(seasons)*length(which(groups==i)), ncol = 2)
      sort.mat[,1] <- rep(1:100,length(seasons)*length(which(groups==i)))
      sort.mat[,2] <- rep(apply(Catch_temp[,which(groups==i)],1,sum)/Catch_tot,length(seasons)*length(which(groups==i)))
      sort.mat <- sort.mat[order(sort.mat[,1]),]
      Allocations[Allocations[,4]==i,6] <- sort.mat[,2]
    }
  }
  Fmult3 <- (0.5*(Allocations[,5]/Allocations[,6]-1)+1)
}else{
  Fmult3 <- rep(1,100*length(seasons)*length(F_cols))
}
```

New approach to achieve multiple forecasting targets – F scaling

```
#Adjust any multipliers of fixed catch values to 1 so that the
#search algorithm will consider them to have achieved their target
Fmult1[fixed_ref] <- 1
Fmult2[fixed_ref] <- 1
Fmult3[fixed_ref] <- 1
Comb_Mult <- Fmult1*Fmult2*Fmult3

#Record the previous adjustment values so they can be used to optimize
#step sizes to speed up target convergence
Last_Mult1 <- DepletionScale
Last_Mult2 <- median(Fmult2[-fixed_ref])
Last_Mult2a <- median(Fmult2[rebuild_ref[which(!is.element(rebuild_ref,fixed_ref))]])
Last_Mult2b <- median(Fmult2[-sort(unique(c(fixed_ref,rebuild_ref)))])

#Plot out progess in achieving targets. This is primarily for diagnosis of a
#run that is failing to converge on an answer in a reasonable period of time.
col_options <- c("black","dark red","dark green","dark blue","orange","purple","red","green","blu
point_options <- c(16,15,17,18,8,9,10,11,12,13,0,1,2,3,4,5,6,14,21,22,23,24,25,19,20)
plot(Fmult1,xlab="year/season/fleet",ylab="Depletion Adjustment",col=rep(col_options[seq_along(F_
plot(rep(F_adjust1,100*length(seasons)*length(F_cols)),xlab="year/season/fleet",ylab="Depletion O
plot(Fmult2,xlab="year/season/fleet",ylab="F Adjustment",col=rep(col_options[seq_along(F_cols)],1
plot(rep(F_adjust2,100*length(seasons)*length(F_cols)),xlab="year/season/fleet",ylab="F Optimizat
plot(Fmult3,xlab="year/season/fleet",ylab="Allocation Adjustment",col=rep(col_options[seq_along(F
plot(F_adjust3,xlab="year/season/fleet",ylab="Allocation Optimization Adjustment",col=rep(col_opt

#Check if all targets have been achieved and if so stop fitting
if(max(abs(1-Fmult1))>=Depletion.Threshold | max(abs(1-Fmult2))>=Annual.F.Threshold | max(abs(1-F
if(FScale==0 & loop>2){keepFitting<-FALSE}

#Here we check that no Fs have been reduced to zero that need some catch
#If that has occured repace the zero F with a small starting value 0.05 so that the
#search algorithm can act on it to achieve the true target value.
#This is needed if the ABC loop was used to perform a zero catch run and then
#rebuild run is performed starting from those zero values
zero_Fs <- which(forecast_F[,4]==0)
increase_Fs <- which(Comb_Mult>1)
if(length(zero_Fs)>0 & length(increase_Fs)>0){
  mod_Fs <- zero_Fs[is.element(zero_Fs,increase_Fs)]
  if(length(mod_Fs)>0){
    forecast_F[mod_Fs,4] <- 0.05
  }
}
#Now adjust the previous F values by the estimated multiplier to create a
#new estimate of the target Fs, make sure to overwrite any fixed catches
#with their original values.
forecast_F[,4] <- forecast_F[,4]*Comb_Mult
forecast_F[fixed_ref,4] <- Fixed_catch_target[,4]
forecast[["ForeCatch"]] <- forecast_F
#Write the modified forecast data out to a file and rerun projections
unlink(paste0(getwd(),"/forecast.ss"))
SS_writeforecast(mylist=forecast,overwrite = TRUE)
shell(paste("cd /d ",getwd()," && ss -nohess",sep=""))
#If all values have converged check if this is the OFL, ABC, or Rebuild loop
```

A combined all target multiplier is used to adjust SS input values and the forecast is re-run before again comparing achieved and target outcomes

NOAA FISHERIES

New approach to achieve multiple forecasting targets – OFL/ABC/Rebuild

OFL

- Benchmark scaler is used with the calculation depending on the designated target (MSY, SPR%, SSB%)
- Once completed all results are saved to an OFL folder

ABC

- Currently targets an annual F_{ABC}=X%* F_{OFL}
- Benchmark scaler is set to 1
- Once completed all results are saved to an ABC folder

Rebuild

- Targets F_{OFL} after rebuilding target year a reduced F in earlier years to achieve SSB_{OFL} in the rebuild year
- Benchmark scaler is set to 1
- Once completed all results are saved to a Rebuild folder

New approach to achieve multiple forecasting targets – SSB% and SPR%

- Both the SSB% and SPR% benchmarks targets have a known status target and can therefore use a simple direct search.
- SSB% target is searching to achieve the desired SSB/SSB0 ratio at equilibrium
- SPR% target is adjusted for projected future recruitment by searching to achieve the desired (SSB/Recruits)/(SSB0/R0) ratio at equilibrium
- When projected recruitment is less than R0 SPR% F will be greater than SSB% F. When projected recruitment is greater than R0 SPR% F will be less than SSB% F.

New approach to achieve multiple forecasting targets – MSY and Fmax

```r
}else if(Forecast_target==2){
  Depletion <- TimeFit3$SpawnBio/Virgin_bio
  Achieved.Depletion <- median(Depletion[(length(Depletion)-29):length(Depletion)])
  if(First_run == TRUE){
    Target.Depletion <- Achieved.Depletion
    First_run <- FALSE
  }
  Target.Rebuild <- Target.Depletion

  if(max(abs(1-Fmult3))>Allocation.Threshold |
     max(abs(1-Fmult2))>Annual.F.Threshold |
     max(abs(1-Fmult1))>Depletion.Threshold){
    loop<-loop-1
    subloop<-subloop+1
    if(F_max==TRUE){
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])/
                        sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),"Recruit_0"])
    }else{
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])
    }
    MSY.Fit[1,] <- c(Achieved.Catch,FScale,Achieved.Depletion,Target.Depletion)
  }else{
    subloop<-0
    if(F_max==TRUE){
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])/
                        sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),"Recruit_0"])
    }else{
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])
    }
    MSY.Fit <- rbind(MSY.Fit[1,],MSY.Fit)
    MSY.Fit[1,] <- c(Achieved.Catch,FScale,Achieved.Depletion,Target.Depletion)
    if(loop>1){
      if(Achieved.Catch<Last_Achieved_Catch){
        search_step <- -0.5*search_step
      }

      Target.Depletion <- Target.Depletion+search_step

      min_diff <- which(abs(MSY.Fit[,4]-Target.Depletion)<0.001)
      if(length(min_diff)>0){
        Old.Catch <- MSY.Fit[min_diff[1],1]
        if(Old.Catch<Achieved.Catch){
          search_step <- -0.5*search_step
        }
        Target.Depletion <- Target.Depletion+search_step
        Achieved.Catch <- Old.Catch
      }
    }else{
      steps <- seq(0.1,0.9,0.1)
      New.Target.Depletion <- steps[which(abs(steps-Target.Depletion)==min(abs(steps-Target.Depletion)))[1]]
      if(New.Target.Depletion<Target.Depletion){
        search_step <- -1*search_step
      }
      Target.Depletion <- New.Target.Depletion
    }
    Last_Achieved_Catch <- Achieved.Catch
  }
  DepletionScale <- (1-Target.Depletion)/(1-Achieved.Depletion)
  if(F_max==TRUE){
```

- For MSY (raw retained catch) and Fmax (retained catch per recruit) search the value of the benchmark target is unknown
- A two stage search is needed to only include results with correct allocations in the search for MSY
- Stage one = SSB% target search
- Stage two = search SSB% values to find SSB% that achieves MSY or Fmax

New approach to achieve multiple forecasting targets – MSY and Fmax

```
}else if(Forecast_target==2){
  Depletion <- TimeFit3$SpawnBio/Virgin_bio
  Achieved.Depletion <- median(Depletion[(length(Depletion)-29):length(Depletion)])
  if(First_run == TRUE){
    Target.Depletion <- Achieved.Depletion
    First_run <- FALSE
  }
  Target.Rebuild <- Target.Depletion

  if(max(abs(1-Fmult3))>Allocation.Threshold |
     max(abs(1-Fmult2))>Annual.F.Threshold |
     max(abs(1-Fmult1))>Depletion.Threshold){
    loop<-loop-1
    subloop<-subloop+1
    if(F_max==TRUE){
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])/
                        sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),"Recruit_0"])
    }else{
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])
    }
    MSY.Fit[1,] <- c(Achieved.Catch,FScale,Achieved.Depletion,Target.Depletion)
  }else{
    subloop<-0
    if(F_max==TRUE){
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])/
                        sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),"Recruit_0"])
    }else{
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])
    }
    MSY.Fit <- rbind(MSY.Fit[1,],MSY.Fit)
    MSY.Fit[1,] <- c(Achieved.Catch,FScale,Achieved.Depletion,Target.Depletion)
    if(loop>1){
      if(Achieved.Catch<Last_Achieved_Catch){
        search_step <- -0.5*search_step
      }

      Target.Depletion <- Target.Depletion+search_step

      min_diff <- which(abs(MSY.Fit[,4]-Target.Depletion)<0.001)
      if(length(min_diff)>0){
        Old.Catch <- MSY.Fit[min_diff[1],1]
        if(Old.Catch<Achieved.Catch){
          search_step <- -0.5*search_step
        }
        Target.Depletion <- Target.Depletion+search_step
        Achieved.Catch <- Old.Catch
      }
    }else{
      steps <- seq(0.1,0.9,0.1)
      New.Target.Depletion <- steps[which(abs(steps-Target.Depletion)==min(abs(steps-Target.Depletion)))[1]]
      if(New.Target.Depletion<Target.Depletion){
        search_step <- -1*search_step
      }
      Target.Depletion <- New.Target.Depletion
    }
    Last_Achieved_Catch <- Achieved.Catch
  }
  DepletionScale <- (1-Target.Depletion)/(1-Achieved.Depletion)
  if(F_max==TRUE){
```

- For MSY (raw retained catch) and Fmax (retained catch per recruit) search the value of the benchmark target is unknown
- A two stage search is needed to only include results with correct allocations in the search for MSY
- Stage one = SSB% target search
- Stage two = search SSB% values to find SSB% that achieves MSY or Fmax

NOAA FISHERIES

New approach to achieve multiple forecasting targets – MSY and Fmax

```
}else if(Forecast_target==2){
  Depletion <- TimeFit3$SpawnBio/Virgin_bio
  Achieved.Depletion <- median(Depletion[(length(Depletion)-29):length(Depletion)])
  if(First_run == TRUE){
    Target.Depletion <- Achieved.Depletion
    First_run <- FALSE
  }
  Target.Rebuild <- Target.Depletion

  if(max(abs(1-Fmult3))>Allocation.Threshold |
     max(abs(1-Fmult2))>Annual.F.Threshold |
     max(abs(1-Fmult1))>Depletion.Threshold){
    loop<-loop-1
    subloop<-subloop+1
    if(F_max==TRUE){
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])/
                        sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),"Recruit_0"])
    }else{
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])
    }
    MSY.Fit[1,] <- c(Achieved.Catch,FScale,Achieved.Depletion,Target.Depletion)
  }else{
    subloop<-0
    if(F_max==TRUE){
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])/
                        sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),"Recruit_0"])
    }else{
      Achieved.Catch <- sum(TimeFit3[(length(TimeFit3[,1])-9):length(TimeFit3[,1]),Catch_cols3])
    }
    MSY.Fit <- rbind(MSY.Fit[1,],MSY.Fit)
    MSY.Fit[1,] <- c(Achieved.Catch,FScale,Achieved.Depletion,Target.Depletion)
    if(loop>1){
      if(Achieved.Catch<Last_Achieved_Catch){
        search_step <- -0.5*search_step
      }

      Target.Depletion <- Target.Depletion+search_step

      min_diff <- which(abs(MSY.Fit[,4]-Target.Depletion)<0.001)
      if(length(min_diff)>0){
        Old.Catch <- MSY.Fit[min_diff[1],1]
        if(Old.Catch<Achieved.Catch){
          search_step <- -0.5*search_step
        }
        Target.Depletion <- Target.Depletion+search_step
        Achieved.Catch <- Old.Catch
      }
    }else{
      steps <- seq(0.1,0.9,0.1)
      New.Target.Depletion <- steps[which(abs(steps-Target.Depletion)==min(abs(steps-Target.Depletion)))[1]]
      if(New.Target.Depletion<Target.Depletion){
        search_step <- -1*search_step
      }
      Target.Depletion <- New.Target.Depletion
    }
    Last_Achieved_Catch <- Achieved.Catch
  }
  DepletionScale <- (1-Target.Depletion)/(1-Achieved.Depletion)
  if(F_max==TRUE){
```

- For MSY (raw retained catch) and Fmax (retained catch per recruit) search the value of the benchmark target is unknown
- A two stage search is needed to only include results with correct allocations in the search for MSY
- Stage one = SSB% target search
- Stage two = search SSB% values to find SSB% that achieves MSY or Fmax

NOAA FISHERIES

Validating forecast results

- All projection approaches utilize the underlying SS population model mechanics for projections
- Results for every approach are validated by comparing reported forecast outcomes to those targeted by the stock assessment analyst
- Each approach has required an increase in the computational complexity of the projection search algorithms and software
- These changes represent an ongoing effort to continuously improve the realism of forecast projections

Plug for upcoming RESTORE planning project

- A collaborative effort between Vaughan Analytics (Nathan Vaughan, SEFSC (John Walter, Kate Siegfried, Skyler Sagarese), GMFMC (Ryan Rindone), SERO (Nick Farmer)
- Phase 1: Identify the key desirable features that could be improved for a future next generation forecasting platform
- Phase 2: Collaborate with the SSC and Council to determine best approaches for incorporating the resulting advice into management
- Phase 3: Work with range of potential users to identify optimal interface complexity/capacity tradeoffs for future software development

NOAA FISHERIES

Questions

- Forecasting function github repository: https://github.com/SEFSC/SFD-AllocationForecasting
- Email: nathan.vaughan@noaa.gov

NOAA FISHERIES