DLMtool: Data-Limited Methods Toolkit (v1.34)

Tom Carruthers[*]

September 2014

Contents

[*]t.carruthers@fisheries.ubc.ca

1 Introduction

As many as 90 per cent of the world's fish populations have insufficient data to conduct a conventional stock assessment (Costello et al. 2012). Although a wide range of data-limited approaches have been described in the primary and gray literature these are not readily available, easily tested or compared. Critically, the path forward is not clear: what methods should we be using for a given stock/fishery/data quality? What is the value of collecting additional data? What is an appropriate stop-gap method?

DLMtool is a collaboration between the University of British Columbia and the Natural Resources Defense Council aimed at addressing these issues by offering a powerful, transparent approach to selecting and applying various data-limited stock assessment methods. The DLMtool takes a similar approach to Carruthers et al. (2014) and uses Management Stratey Evaluation (MSE) and parallel computing to make powerful diagnostics accessible. A streamlined command structure and operating model builder allow for rapid simulation testing and graphing of results.

DLMtool is specifically designed to be extensible in order to encourage the development and testing of new approaches for informing the management of data-limited fish stocks. The package is designed such that the very same methods that are tested by MSE can be applied to provide real management recommendations from real data. Easy incorporation of real data is central advantage of the software and a set of related functions automatically detect what method can be applied given the available data and what additional data are required to get other methods working.

2 A note on version 1.34

The package is still in a beta-testing phase but is now available on CRAN-R. If you find a bug or a problem please send a report to t.carruthers@fisheries.ubc.ca so that I can fix it!

Fundamentally the package is stochastic so if you run into problems with the code, please report it and in the mean time simply try running it again: it might just have been a rare combination of sampled parameters that caused the problem.

I highly recommend that you use parallel-processing. If however you abort a parallel process (e.g. runMSE()) half-way through you are in the lap of the gods! It will often be necessary to quit R and restart the code.

– New to 1.34 –

(1) Temporal autocorrelation in recruitment deviations. This is a simple AR1 type process controlled by a fraction A which controls the weighed mean according to previous recruitment deviation and new recruitment deviation (sigma) in which dev(1)=sigma(1), dev(t)=A*dev(t-1)+(1-A)*sigma(t)

(2) Ricker stock recruitment relationship. Similarly to the Beverton Holt model this is parameterized according to steepness. This is now a slot in every stock object Stock@SRrel. SRrel=1 is BH stock-recruitment. SRrel=2 is Ricker stock-recruitment.

(3) SPMSY is no longer bugged. When it can't find a pair of r and K pairs that allow current depletion (inferred from catch trajectory) it widens the search to a larger range. Type in SPMSY at the R prompt to see what this new version looks like.

(4) DynF, Gcontrol, Rcontrol and Rcontrol2 management proceedures now infer the gradient in Surplus production with biomass according to smoothed trends in catch and biomass. These are 1-degree loess smoothers applied to log(catch) and log(biomass). These approaches now seem to work a bit better, particularly DynF. Again type these in at the R prompt to see whats new.

(5) One of the requirements of submission to CRAN-R is that parallel processing be limited to a maximum of two processors. Build time must also be relatively quick. These two restrictions make sense but they limit the full range of funcitonality that can be demonstrated in these vignettes. If you have a quad-core computer you should specify cpus=8 when calling sfInit()

(6) Deterministic methods. If the user specifies reps=1, any DLM quota method will produce a single quota recommendation that is calculated from the mean values of all the inputs. In some cases this provides very similar outcomes to sampling stochastic quotas and choosing some intermediate percentile (although rarely exactly the same). The central advantage is that the closed loop calculations are many fewer and it takes much less time to get comparable results.

3 Prerequisites

At the start of every session there are a few things to do: load the DLMtool library, make data available and set up cluster computing.

3.1 Loading the library

```
library(DLMtool)

## Loading required package:  snowfall
## Loading required package:  snow
## Loading required package:  boot
## Loading required package:  MASS
```

3.2 Unpacking data and objects

```
for(i in 1:length(DLMdat))assign(DLMdat[[i]]@Name,DLMdat[[i]])
```

3.3 Initiating the cluster

Set up the cluster (note that most computers make use of hyperthreading technology so a quad-core PC has 8 threads, this is set to 2 here to meet CRAN-R package submission requirements).

```
sfInit(parallel=T,cpus=2)

## R Version:  R Under development (unstable) (2014-09-10 r66562)

## snowfall 1.84-6 initialized (using snow 0.3-13):  parallel execution on 2 CPUs.
```

Export all the data and functions to the cluster

```
sfExportAll()
```

Set a random seed in order make results reproducible

```
set.seed(1)
```

4 Quick start

Here is a quick demonstration of core DLMtool functionality.

4.1 Define an operating model

The operating model is the 'simulated reality': a series of known simulations against which to test various data-limited methods. Operating models can either be specified in detail according to each variable in turn (e.g. sample natural mortality rate between 0.2 and 0.3, trajectory in fishing effort of between 0.5 and 1 per cent per annum) or alternatively the user can rapidly construct an operating model based on a set of default Stock, Fleet and Observation models. In this case we take the latter approach and pick the Rockfish Stock type, a Generic fleet type and an observation model that generates data that can be both imprecise and biased.

```
OM<-new('OM',
        Rockfish,
        Generic_fleet,
        Imprecise_Biased)
```

The operating model class 'OM' has many different slots which control the ranges of population and fleet parameters that may be sampled in addition to parameters that control the quality of the data simulated. You can list these using slotNames() or can look up the help file entry:

```
slotNames(OM)

##  [1] "Name"          "nyears"        "maxage"        "R0"
##  [5] "M"             "Msd"           "Mgrad"         "h"
##  [9] "SRrel"         "Linf"          "K"             "t0"
## [13] "Ksd"           "Kgrad"         "Linfsd"        "Linfgrad"
## [17] "recgrad"       "a"             "b"             "ageM"
## [21] "ageMsd"        "D"             "Size_area_1"   "Frac_area_1"
## [25] "Prob_staying"  "Source"        "beta"          "Spat_targ"
## [29] "AFS"           "age05"         "Vmaxage"       "Fsd"
## [33] "Fgrad"         "AC"            "Cobs"          "Cbiascv"
## [37] "CAAobs"        "CALobs"        "Iobs"          "Perr"
## [41] "Mcv"           "Kcv"           "t0cv"          "Linfcv"
## [45] "LFCcv"         "LFScv"         "B0cv"          "FMSYcv"
## [49] "FMSY_Mcv"      "BMSY_B0cv"     "ageMcv"        "rcv"
## [53] "Fgaincv"       "A50cv"         "Dbiascv"       "Dcv"
## [57] "Btbiascv"      "Btcv"          "Fcurbiascv"    "Fcurcv"
## [61] "hcv"           "Icv"           "maxagecv"      "Reccv"
## [65] "Irefcv"        "Crefcv"        "Brefcv"

class?OM
```

4.2 Define a subset of data-limited methods

There are three different types of assessment method / management procedure currently included in DLMtool: DLM quota (output controls), DLM size (size/age controls) and DLM space (spatial controls). In this example we use a generic class finder 'avail' to list all available methods of class 'DLM quota' and select some for simulation testing.

```
avail('DLM quota')

##  [1] "BK"           "BK_CC"        "BK_ML"        "DBSRA"        "DBSRA4010"
##  [6] "DBSRA_40"     "DBSRA_ML"     "DCAC"         "DCAC4010"     "DCAC_40"
## [11] "DCAC_ML"      "DD"           "DD4010"       "DepF"         "DynF"
## [16] "FMSYref"      "FMSYref50"    "FMSYref75"    "Fdem"         "Fdem_CC"
## [21] "Fdem_ML"      "Fratio"       "Fratio4010" "Fratio_CC"    "Fratio_ML"
## [26] "GB_CC"        "GB_slope"     "GB_target"    "Gcontrol"     "MMHCR"
## [31] "Rcontrol"     "Rcontrol2"    "SBT1"         "SBT2"         "SPMSY"
## [36] "SPSRA"        "SPSRA_ML"     "YPR"          "YPR_CC"       "YPR_ML"

methods<-c("Fratio",
           "DCAC",
           "Fdem",
           "DD")
```

To find out more about these methods you can use the built-in R help functions. E.g:

```
?Fratio
?DBSRA
```

Or simply view the code. E.g:

```
Fratio

## function (x, DLM, reps = 100)
## {
##     depends = "DLM@Abun,DLM@CV_Abun,DLM@FMSY_M,DLM@CV_FMSY_M,DLM@Mort,DLM@CV_Mort"
##     Ac <- trlnorm(reps, DLM@Abun[x], DLM@CV_Abun[x])
##     OFLfilter(Ac * trlnorm(reps, DLM@Mort[x], DLM@CV_Mort[x]) *
##         trlnorm(reps, DLM@FMSY_M[x], DLM@CV_FMSY_M[x]))
## }
## <environment: namespace:DLMtool>
## attr(,"class")
## [1] "DLM quota"
```

4.3 Run an MSE and plot results

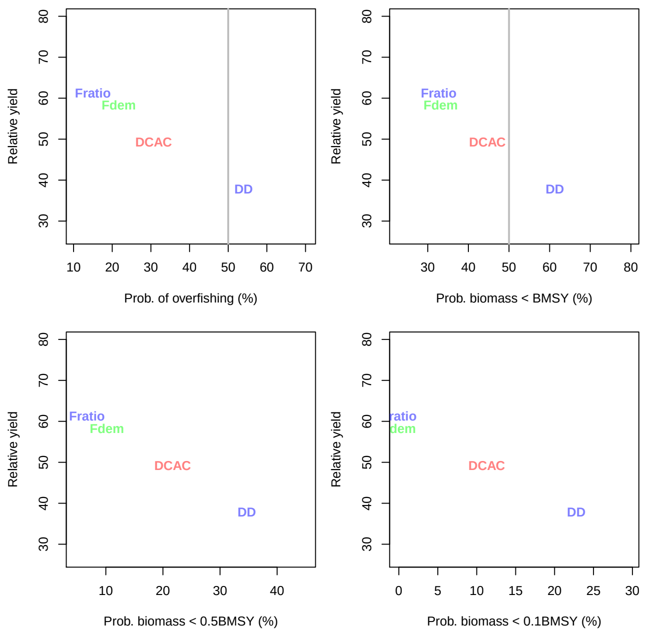
The methods can now be tested using the operating model. NOTE that this is just a demonstration, in a real MSE you should use many more simulations (¿200), reps (samples per method ¿100) and perhaps a more frequent assessment interval (e.g. 2 or 3 years). Note that when reps is set to 1, all stochastic methods use the mean value of an input and do not sample from the distribution according to the specified CV (the methods become deterministic and no longer produce a distribution of the quota recommendation).

```
RockMSE<-runMSE(OM,methods,nsim=20,reps=1,proyears=30,interval=5)
```

```
## [1] "Loading operating model"
## [1] "Optimizing for user-specified movement"
## [1] "Optimizing for user-specified depletion"
## [1] "Calculating historical stock and fishing dynamics"
## [1] "Calculating MSY reference points"
## [1] "Calculating reference yield - best fixed F strategy"
## [1] "Determining available methods"
## [1] "1/4 Running MSE for Fratio"
## ..............................
## [1] "2/4 Running MSE for DCAC"
## ..............................
## [1] "3/4 Running MSE for Fdem"
## ..............................
## [1] "4/4 Running MSE for DD"
## ..............................
```

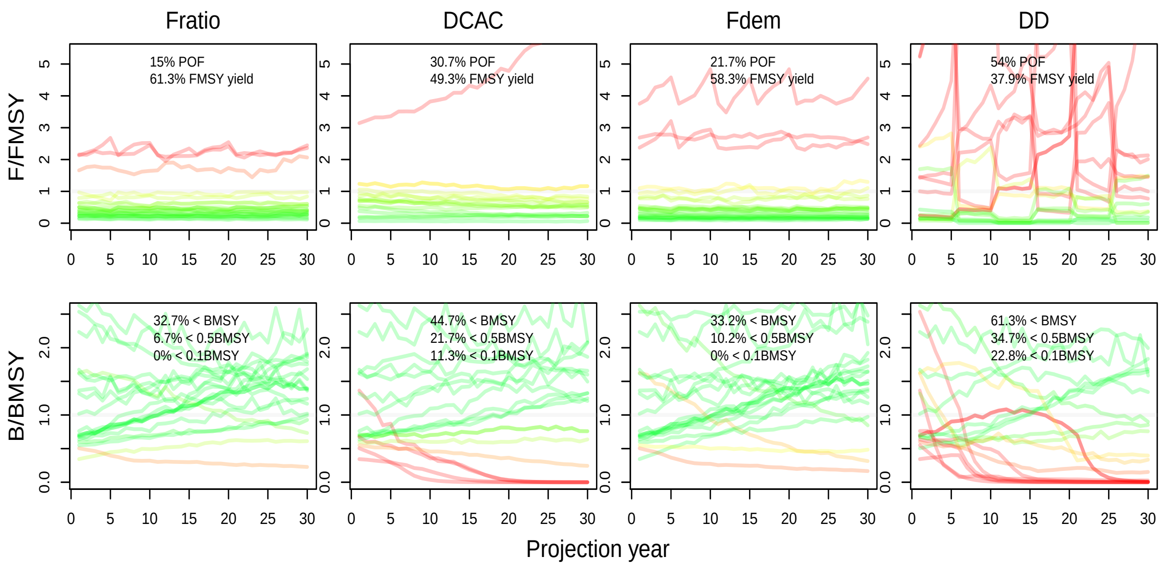
A trade-off function provides a visualization of the expected (mean) performance of the methods in terms of stock status, overfishing and yield.

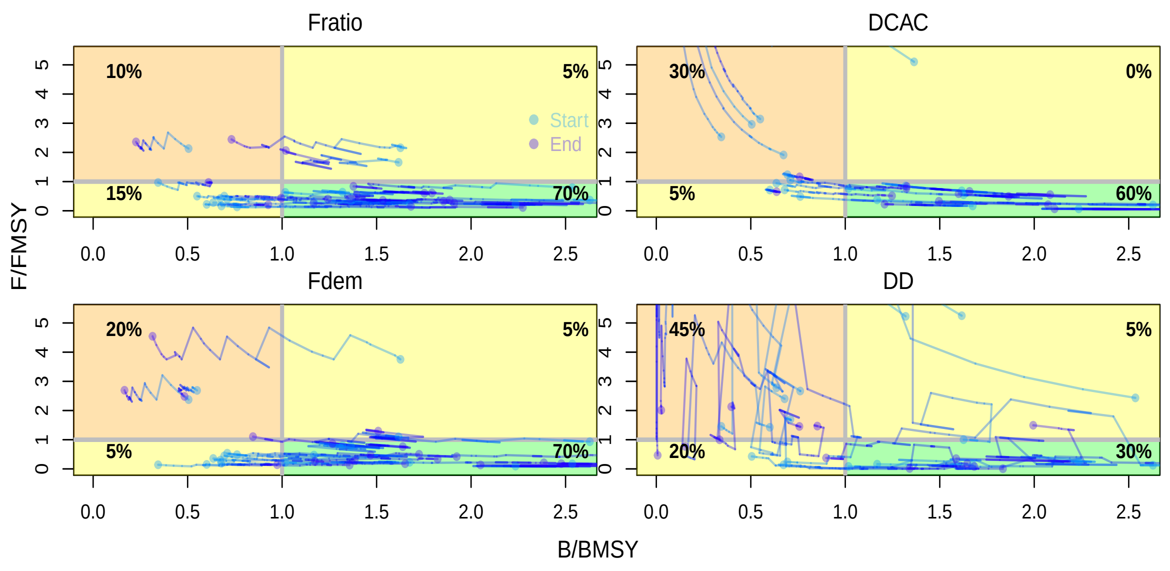
```
Tplot(RockMSE)
```

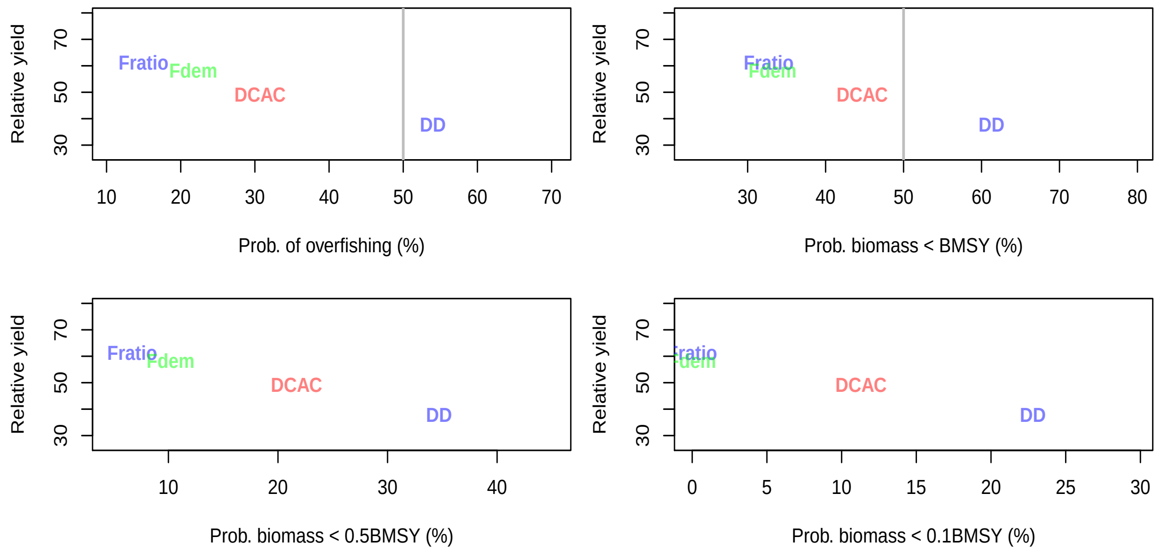


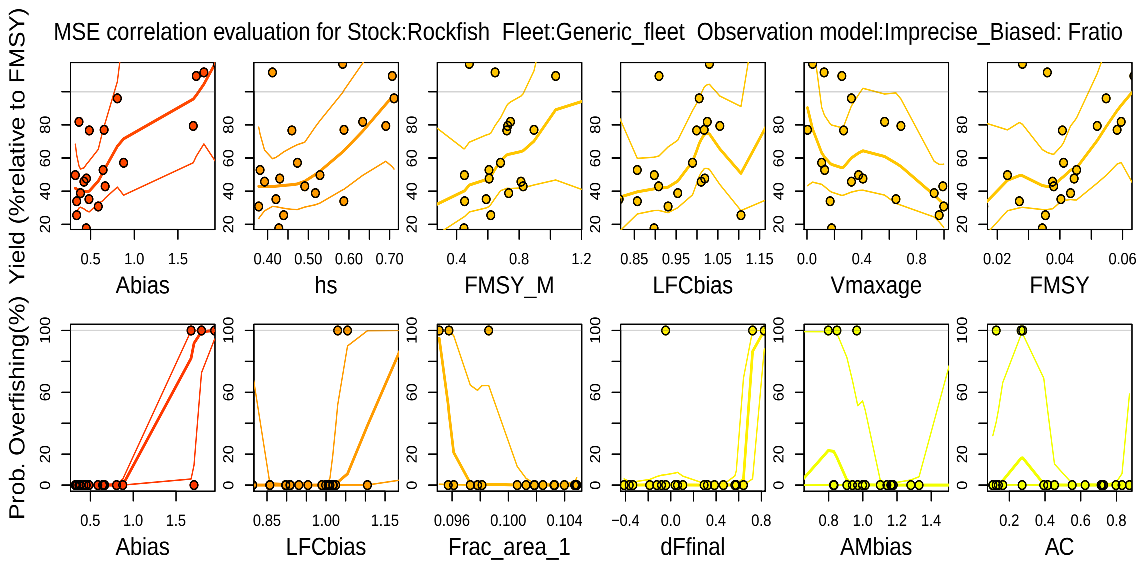
Much more detailed plots are also available that include stock and overfishing trajectories, kobe plots and sensitivity analyses:

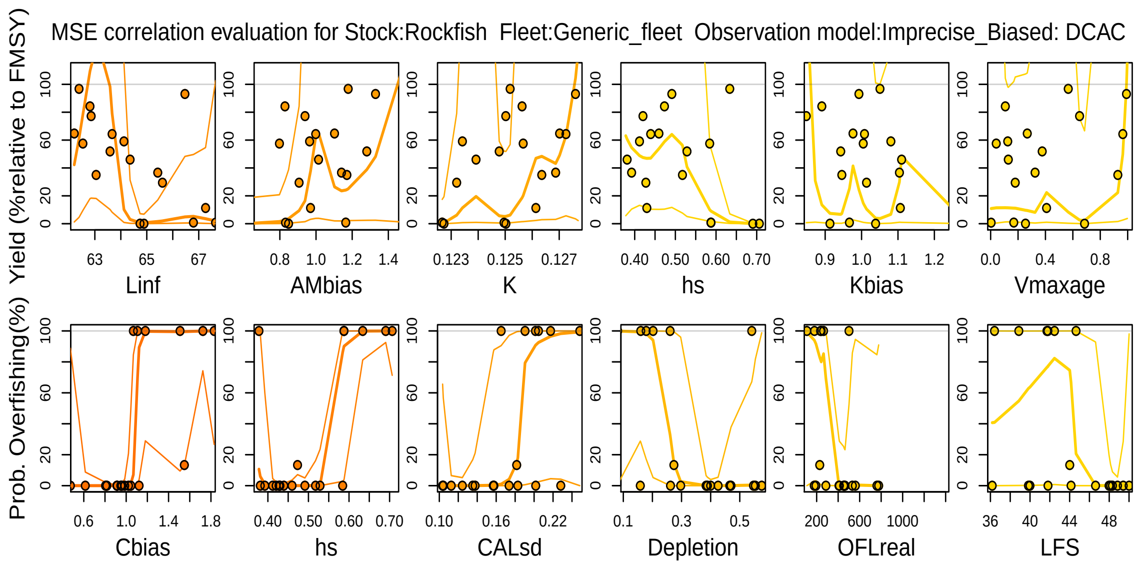
```
plot(RockMSE)
```

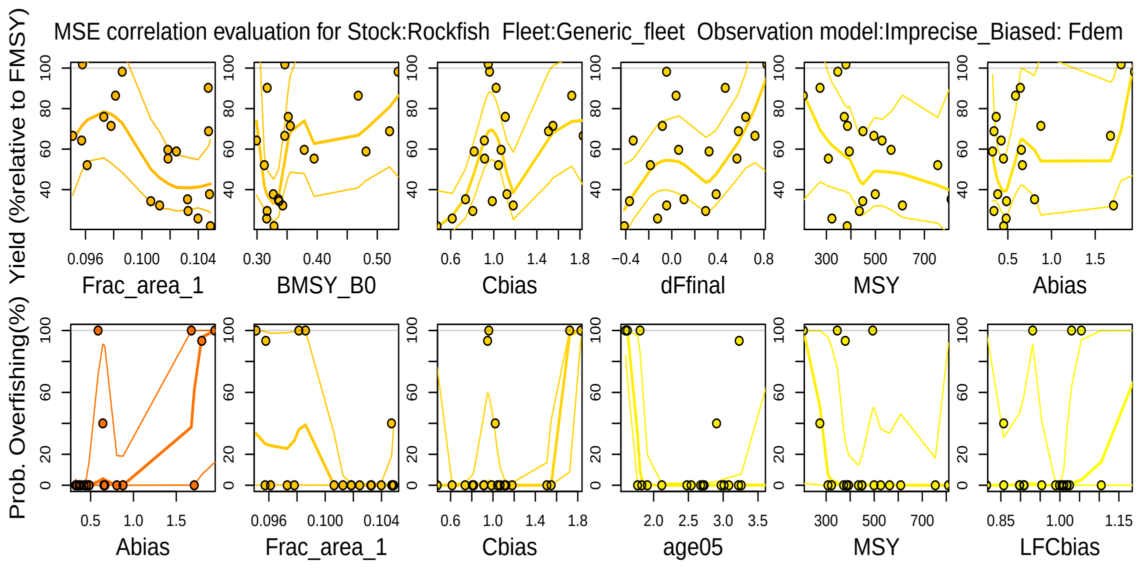


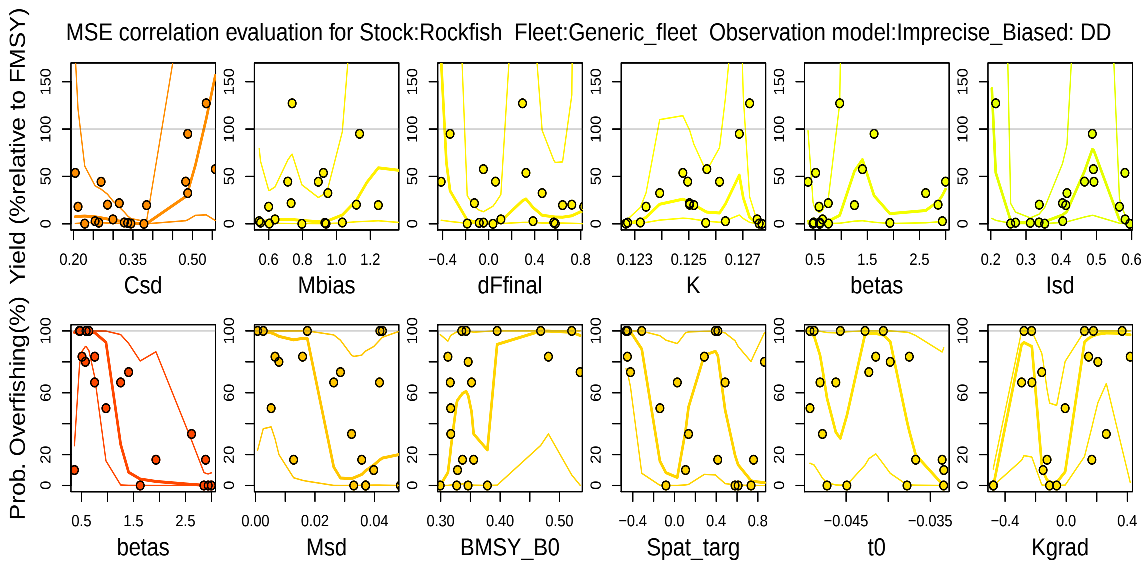

Fratio
DCAC
Fdem
DD
F/FMSY
B/BMSY
Start
End
10%
5%
15%
70%
30%
0%
5%
60%
20%
5%
5%
70%
45%
5%
20%
30%


Relative yield
Prob. of overfishing (%)
Prob. biomass < BMSY (%)
Prob. biomass < 0.5BMSY (%)
Prob. biomass < 0.1BMSY (%)
Fratio
Fdem
DCAC
DD


MSE correlation evaluation for Stock:Rockfish Fleet:Generic_fleet Observation model:Imprecise_Biased: Fratio
Yield (%relative to FMSY)
Prob. Overfishing(%)
Abias
hs
FMSY_M
LFCbias
Vmaxage
FMSY
Abias
LFCbias
Frac_area_1
dFfinal
AMbias
AC


MSE correlation evaluation for Stock:Rockfish Fleet:Generic_fleet Observation model:Imprecise_Biased: DCAC
Yield (%relative to FMSY)
Prob. Overfishing(%)
Linf
AMbias
K
hs
Kbias
Vmaxage
Cbias
hs
CALsd
Depletion
OFLreal
LFS





The final plot you see here illustrates how different simulated parameters (x axis) affect yield (top row) and the probability of overfishing (bottom row). The plots are organised in terms of most important (left, high correlation) to least important (right, lower correlation).

This particular plot is for DD, a delay-difference model. It is clear from the lower left hand plot that probability of overfishing is most strongly determined by the beta parameter that controls hyperstability in the relative abundance index. We can see that below a value of 1 (hyperstable, eg. relative abundance index declines more slowly than 'true' simulated biomass) the propensity for overfishing is greatly increased. This is a predictable result but these plots demonstrate that this is a dominant effect relative to other aspects of the simulation.

4.4 Applying methods to real data

A number of real DLM data-objects were loaded into the workspace when we ran SendAll() at the start. In this section we examine a real data object and apply data-limited methods to it. Just like the operating model we can find all the objects of real data class 'DLM', we can list the slots of a DLM data object and also look up this class in the help file:

```
avail('DLM')

##  [1] "Atlantic_mackerel"  "Canary_Rockfish"    "China_rockfish"
##  [4] "Cobia"              "Example_datafile"   "Gulf_blue_tilefish"
##  [7] "Red_snapper"        "Simulation_1"       "Simulation_2"
## [10] "ourReefFish"

slotNames(Canary_Rockfish)

##  [1] "Name"       "Year"       "Cat"        "Ind"        "Rec"
##  [6] "t"          "AvC"        "Dt"         "Mort"       "FMSY_M"
## [11] "BMSY_B0"    "Cref"       "Bref"       "Iref"       "AM"
## [16] "LFC"        "LFS"        "CAA"        "Dep"        "Abun"
## [21] "vbK"        "vbLinf"     "vbt0"       "wla"        "wlb"
## [26] "steep"      "CV_Cat"     "CV_Dt"      "CV_AvC"     "CV_Ind"
## [31] "CV_Mort"    "CV_FMSY_M"  "CV_BMSY_B0" "CV_Cref"    "CV_Bref"
## [36] "CV_Iref"    "CV_Rec"     "CV_Dep"     "CV_Abun"    "CV_vbK"
## [41] "CV_vbLinf"  "CV_vbt0"    "CV_AM"      "CV_LFC"     "CV_LFS"
## [46] "CV_wla"     "CV_wlb"     "CV_steep"   "sigmaL"     "MaxAge"
## [51] "Units"      "Ref"        "Ref_type"   "Log"        "params"
## [56] "PosMeths"   "Meths"      "OM"         "Obs"        "quota"
## [61] "quotabias"  "Sense"      "CAL_bins"   "CAL"

class?DLM
```

DLMtool includes functions to interrogate a real data object to see what methods can be applied, those that cannot and also what data are needed to get those methods working:

```
Can(Canary_Rockfish)

##  [1] "BK"         "DBSRA"      "DBSRA4010"  "DBSRA_40"   "DCAC"
##  [6] "DCAC4010"   "DCAC_40"    "DD"         "DD4010"     "DepF"
## [11] "DynF"       "Fdem"       "Fratio"     "Fratio4010" "GB_slope"
## [16] "Gcontrol"   "MMHCR"      "Rcontrol"   "Rcontrol2"  "SBT1"
## [21] "SPMSY"      "SPSRA"      "YPR"        "matsizlim"  "area1MPA"

Cant(Canary_Rockfish)

##       [,1]         [,2]
##  [1,] "BK_CC"      "Insufficient data"
##  [2,] "BK_ML"      "Insufficient data"
##  [3,] "DBSRA_ML"   "Insufficient data"
##  [4,] "DCAC_ML"    "Insufficient data"
##  [5,] "FMSYref"    "Insufficient data"
##  [6,] "FMSYref50"  "Insufficient data"
##  [7,] "FMSYref75"  "Insufficient data"
##  [8,] "Fdem_CC"    "Insufficient data"
##  [9,] "Fdem_ML"    "Insufficient data"
## [10,] "Fratio_CC"  "Insufficient data"
## [11,] "Fratio_ML"  "Insufficient data"
## [12,] "GB_CC"      "Produced all NA scores"
## [13,] "GB_target"  "Produced all NA scores"
## [14,] "SBT2"       "Produced all NA scores"
## [15,] "SPSRA_ML"  "Insufficient data"
## [16,] "YPR_CC"    "Insufficient data"
## [17,] "YPR_ML"    "Insufficient data"
```

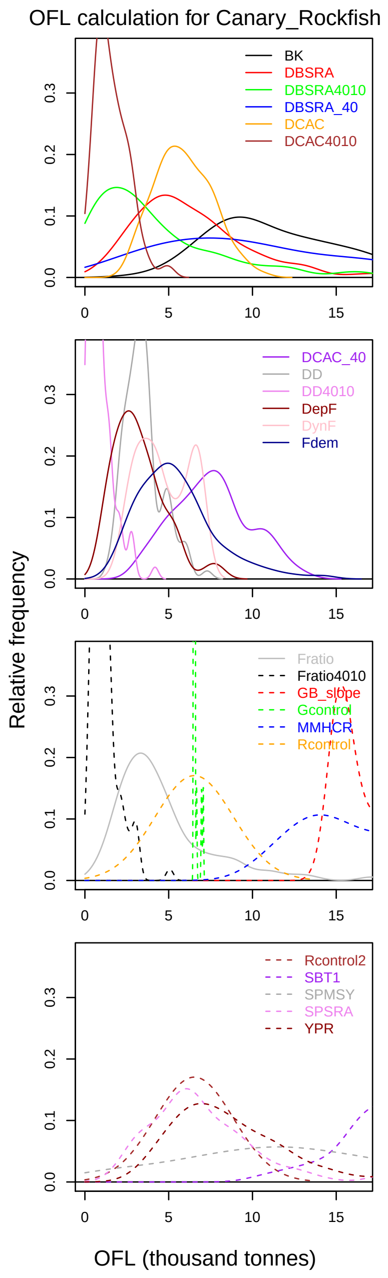
```
Needed(Canary_Rockfish)
```

```
##  [1] "BK_CC: CAA"             "BK_ML: CAL"
##  [3] "DBSRA_ML: CAL"          "DCAC_ML: CAL"
##  [5] "FMSYref: OM"            "FMSYref50: OM"
##  [7] "FMSYref75: OM"          "Fdem_CC: CAA"
##  [9] "Fdem_ML: CAL"           "Fratio_CC: CAA"
## [11] "Fratio_ML: CAL"         "GB_CC: Cref"
## [13] "GB_target: Cref, Iref" "SBT2: Rec, Cref"
## [15] "SPSRA_ML: CAL"          "YPR_CC: CAA"
## [17] "YPR_ML: CAL"
```

The function getQuota() automatically detects which methods can be applied and calculates an OFL distribution for each method which can then be plotted:

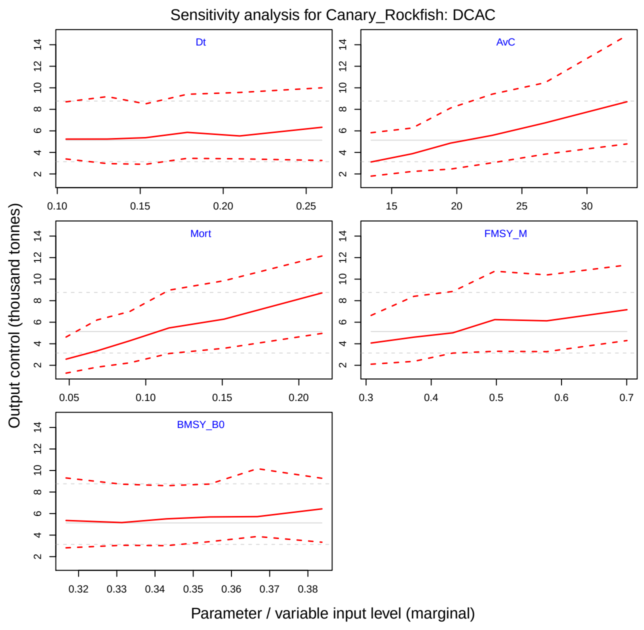
```
RockReal<-getQuota(Canary_Rockfish)
```

```
plot(RockReal)
```



4.5 Conduct a sensitivity analysis

```
RockReal<-Sense(RockReal,"DCAC")
```



The sensitivity plot reveals which inputs to a method most strongly affect the quota recommendation. The idea is to focus discussion on those inputs and their credibility.

5 From MSE to management recommendations

In this section we take a more thorough systematic approach to MSE and data implementation. This is an example of how DLMtool may be used to select methods and then apply them to real data. This is intended to be a straw-man demonstration and in no way is a recommendation about appropriate management objectives!

In this example our real-life stock is a moderately long-lived reef-fish of moderately high recruitment compensation that has been subject to fairly consistent fishing pressure over recent years. We suspect that fishing activities do not effectively operate on older age classes since the fish exhibit ontogenetic offshore movements where there is less fishing. In general the stock is thought to be a relatively low stock levels going by catch rate observations but frankly, we don't have a precise handle on stock depletion. Since fishing activities have changed spatial distribution and the stock is targetted there is the potential for hyperstability in our observations of catch rates over time.

This section assumes that you have completed the four prerequisites already: library(DLMtool), SendAll(), sfInit(parallel=T,cpus=8) and sfExportAll().

5.1 Building an appropriate operating model

We start by specifying an operating model. Looking at the prebuilt stock objects we decide that the 'Snapper' stock object is the closest fit.

```
avail('Stock')

##  [1] "Albacore"     "Blue_shark"   "Bluefin_tuna" "Butterfish"
##  [5] "Herring"      "Mackerel"     "Porgy"        "Rockfish"
##  [9] "Snapper"      "Sole"         "Toothfish"

ourstock<-Snapper
```

We make some modifications to better suite our particular case study such as stock depletion between 5 and 30 per cent of unfished levels and a candidate MPA (between 5 - 15 percent of unfished biomass) with retention (probability of staying in the MPA) of 80 - 99 percent. Remember to get help on the OM objects and their slots type class?OM at the command line.

```
ourstock@D<-c(0.05,0.3)
ourstock@Frac_area_1<-c(0.05,0.15)
ourstock@Prob_staying<-c(0.4,0.99)
```

We now choose a fleet type for our operating model and choose to modify a generic fleet of flat recent effort, adding dome-shaped vulnerability as a possibility for older age classes and some spatial targetting:

```
ourfleet<-Generic_FlatE
ourfleet@Vmaxage<-c(0.5,1)
ourfleet@Spat_targ<-c(1,1.5)
```

Finally, Using our fleet and stock objects we construct an operating model object assuming that the data we have are likely to be imprecise and potentially biased. Type avail('Observation') at the command line to see the various pre-defined observation model objects.

```
ourOM<-new('OM',ourstock,ourfleet,Imprecise_Biased)
```

5.2 MSE evaluation of methods

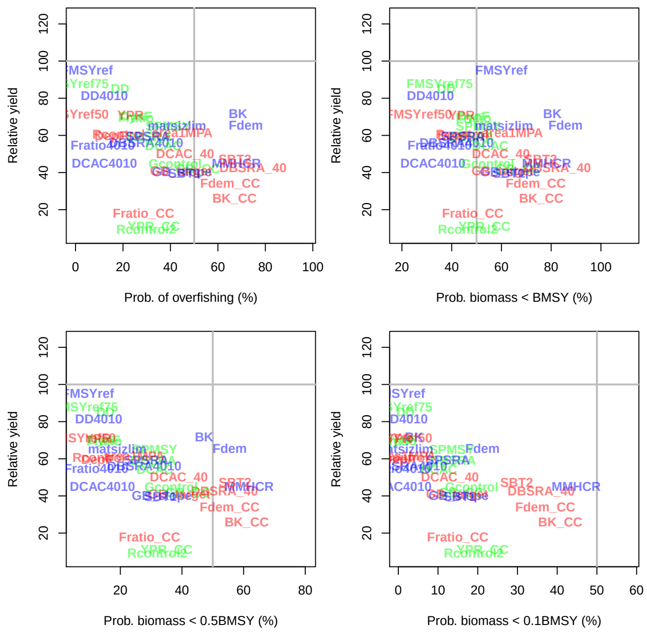
Now that we have our operating model we run a trial MSE. In this case we use a very small number of simulations (20, which is very low to meet CRAN-R package building requirments) but change the length of the projection and the length of the interval between updates to reflect our stock and management system.

Since we do not specify a vector of particular methods, the MSE will run for all possible methods. Note that this could take a few minutes depending on how monstrous you computer is. Note that in a real setting it might be advisable to increase the number of simulations to at least 192 and, if stochastic methods are to be used, increase the samples per method (reps) to at least 100 for this first stage to obtain stable aggregate results.

```
ourMSE<-runMSE(ourOM,proyears=20,interval=5,nsim=20,reps=1)

## [1] "Loading operating model"
## [1] "Optimizing for user-specified movement"
## [1] "Optimizing for user-specified depletion"
## [1] "Calculating historical stock and fishing dynamics"
## [1] "Calculating MSY reference points"
## [1] "Calculating reference yield - best fixed F strategy"
## [1] "Determining available methods"
## [1] "1/35 Running MSE for BK"
## ....................
## [1] "2/35 Running MSE for BK_CC"
## ....................
## [1] "3/35 Running MSE for DBSRA"
## ....................
## [1] "4/35 Running MSE for DBSRA4010"
## ....................
## [1] "5/35 Running MSE for DBSRA_40"
## ....................
## [1] "6/35 Running MSE for DCAC"
## ....................
## [1] "7/35 Running MSE for DCAC4010"
## ....................
## [1] "8/35 Running MSE for DCAC_40"
## ....................
## [1] "9/35 Running MSE for DD"
## ....................
## [1] "10/35 Running MSE for DD4010"
## ....................
## [1] "11/35 Running MSE for DepF"
## ....................
## [1] "12/35 Running MSE for DynF"
## ....................
## [1] "13/35 Running MSE for FMSYref"
## ....................
## [1] "14/35 Running MSE for FMSYref50"
## ....................
## [1] "15/35 Running MSE for FMSYref75"
## ....................
## [1] "16/35 Running MSE for Fdem"
## ....................
## [1] "17/35 Running MSE for Fdem_CC"
## ....................
## [1] "18/35 Running MSE for Fratio"
## ....................
## [1] "19/35 Running MSE for Fratio4010"
## ....................
## [1] "20/35 Running MSE for Fratio_CC"
## ....................
## [1] "21/35 Running MSE for GB_CC"
## ....................
## [1] "22/35 Running MSE for GB_slope"
## ....................
## [1] "23/35 Running MSE for GB_target"
## ....................
## [1] "24/35 Running MSE for Gcontrol"
## ....................
## [1] "25/35 Running MSE for MMHCR"
## ....................
## [1] "26/35 Running MSE for Rcontrol"
## ....................
## [1] "27/35 Running MSE for Rcontrol2"
## ....................
## [1] "28/35 Running MSE for SBT1"
## ....................
## [1] "29/35 Running MSE for SBT2"
## ....................
## [1] "30/35 Running MSE for SPMSY"
## ....................
## [1] "31/35 Running MSE for SPSRA"
## ....................
## [1] "32/35 Running MSE for YPR"
## ....................
## [1] "33/35 Running MSE for YPR_CC"
## ....................
## [1] "34/35 Running MSE for matsizlim"
## ....................
## [1] "35/35 Running MSE for area1MPA"
## ....................
```

A summary trade-off plot reveals a wide range of performance:

```
Tplot(ourMSE)
```



In this example process, we decide that we would like to select a targetted subset of these methods that have greater than 20 percent of long-term yield (given ideal fixed fishing mortality rate), less than a 50 percent rate of overfishing and less than a 10 percent chance of dropping below a low stock level, in this case 10 percent of BMSY. To do this we calculate the summary table and subset it:

```
Results<-summary(ourMSE)
Results
```

```
##         Method Yield         POF         P10         P50         P100
## 1           BK 71.83 25.60 68.50 41.68  4.00 12.83 47.25 41.21 80.50 34.41
## 2        BK_CC 25.80 45.25 67.00 45.58 40.75 35.33 61.00 36.58 76.00 32.51
## 3        DBSRA 59.27 44.40 31.00 46.13 13.75 29.19 30.75 41.49 48.00 37.71
## 4    DBSRA4010 56.29 38.43 29.75 43.63  3.00 12.29 27.75 38.27 42.00 40.86
## 5     DBSRA_40 42.32 42.80 74.75 39.42 36.00 36.94 53.75 43.43 82.50 27.36
## 6         DCAC 54.46 33.17 37.00 41.47 10.25 21.67 31.25 40.19 55.50 38.79
## 7     DCAC4010 45.11 27.12 12.25 31.01  0.25  1.12 14.25 23.80 32.50 31.81
## 8      DCAC_40 49.90 35.36 46.25 46.96 13.00 24.89 39.00 43.40 59.75 40.73
## 9           DD 85.28 84.42 18.75 22.70  1.75  6.74 15.25 24.36 37.75 33.19
## 10      DD4010 81.39 84.92 12.25 19.90  1.50  5.64 13.00 18.38 31.50 25.91
## 11        DepF 59.10 31.71 14.75 30.15  0.25  1.12 12.50 22.45 40.75 37.21
## 12        DynF 69.58 28.24 25.50 40.29  0.25  1.12 14.75 23.81 48.75 40.23
## 13      FMSYref 95.33  8.57  4.75  8.66  0.25  1.12  9.50 12.24 60.00 25.44
## 14    FMSYref50 71.45  4.78  0.00  0.00  0.25  1.12  7.75 10.06 26.75 20.02
## 15    FMSYref75 87.95  6.68  0.00  0.00  0.25  1.12  8.50 11.01 35.25 28.31
## 16       Fdem 65.60 33.44 71.75 42.90 21.25 34.10 55.50 43.07 85.75 28.02
## 17    Fdem_CC 33.85 49.72 65.25 41.72 37.00 36.18 55.50 39.43 73.75 35.83
## 18     Fratio 69.58 28.24 25.50 40.29  0.25  1.12 14.75 23.81 48.75 40.23
## 19 Fratio4010 54.92 35.58 11.50 29.02  0.25  1.12 12.25 22.80 35.25 34.01
## 20  Fratio_CC 17.70 40.33 28.75 40.13 15.00 26.80 29.50 32.32 48.50 33.41
## 21      GB_CC 41.74 39.16 51.50 45.22 15.75 24.24 42.00 39.82 66.75 38.87
## 22   GB_slope 39.83 43.22 44.75 43.12 15.25 24.57 33.50 37.03 63.50 34.03
## 23  GB_target 40.50 34.09 44.50 44.42 14.75 24.03 39.25 39.04 60.50 37.41
## 24   Gcontrol 44.92 41.04 42.00 43.78 18.50 28.93 36.50 40.79 54.50 34.52
## 25      MMHCR 45.06 90.13 67.75 33.97 44.75 34.01 61.75 37.39 78.25 30.62
## 26   Rcontrol 60.75 40.44 18.25 26.57  1.00  3.48 13.00 14.27 44.25 34.27
## 27  Rcontrol2  9.38  6.97 29.75 44.11 19.00 31.06 32.00 41.31 46.50 38.22
## 28       SBT1 39.75 45.92 46.00 43.85 15.50 25.64 33.00 37.36 63.00 33.69
## 29       SBT2 47.38 41.05 67.25 42.00 29.75 33.93 57.25 45.26 75.75 39.01
## 30      SPMSY 65.05 45.30 39.00 40.93 13.50 24.61 31.25 40.32 50.75 39.25
## 31      SPSRA 59.63 29.50 30.50 46.73 12.50 31.06 28.50 39.71 44.50 39.27
## 32        YPR 70.97 26.12 23.25 39.01  0.25  1.12 12.75 20.36 44.00 36.15
## 33     YPR_CC 10.78 16.19 33.00 43.48 21.25 32.11 35.00 36.42 53.25 32.94
## 34   matsizlim 65.44 27.22 42.75 40.34  1.50  5.64 19.00 21.74 61.00 39.19
## 35   area1MPA 61.45 27.64 45.00 43.16  1.75  6.74 24.50 30.60 64.50 39.10
```

```
Targetted<-subset(Results, Results$Yield>20 & Results$POF<50 & Results$P10<10)
Targetted
```

```
##         Method Yield         POF         P10          P50         P100
## 4   DBSRA4010 56.29 38.43 29.75 43.63 3.00 12.29 27.75 38.27 42.00 40.86
## 7    DCAC4010 45.11 27.12 12.25 31.01 0.25  1.12 14.25 23.80 32.50 31.81
## 9          DD 85.28 84.42 18.75 22.70 1.75  6.74 15.25 24.36 37.75 33.19
## 10     DD4010 81.39 84.92 12.25 19.90 1.50  5.64 13.00 18.38 31.50 25.91
## 11       DepF 59.10 31.71 14.75 30.15 0.25  1.12 12.50 22.45 40.75 37.21
## 12       DynF 69.58 28.24 25.50 40.29 0.25  1.12 14.75 23.81 48.75 40.23
## 13    FMSYref 95.33  8.57  4.75  8.66 0.25  1.12  9.50 12.24 60.00 25.44
## 14  FMSYref50 71.45  4.78  0.00  0.00 0.25  1.12  7.75 10.06 26.75 20.02
## 15  FMSYref75 87.95  6.68  0.00  0.00 0.25  1.12  8.50 11.01 35.25 28.31
## 18     Fratio 69.58 28.24 25.50 40.29 0.25  1.12 14.75 23.81 48.75 40.23
## 19 Fratio4010 54.92 35.58 11.50 29.02 0.25  1.12 12.25 22.80 35.25 34.01
## 26   Rcontrol 60.75 40.44 18.25 26.57 1.00  3.48 13.00 14.27 44.25 34.27
## 32        YPR 70.97 26.12 23.25 39.01 0.25  1.12 12.75 20.36 44.00 36.15
## 34   matsizlim 65.44 27.22 42.75 40.34 1.50  5.64 19.00 21.74 61.00 39.19
## 35   area1MPA 61.45 27.64 45.00 43.16 1.75  6.74 24.50 30.60 64.50 39.10
```

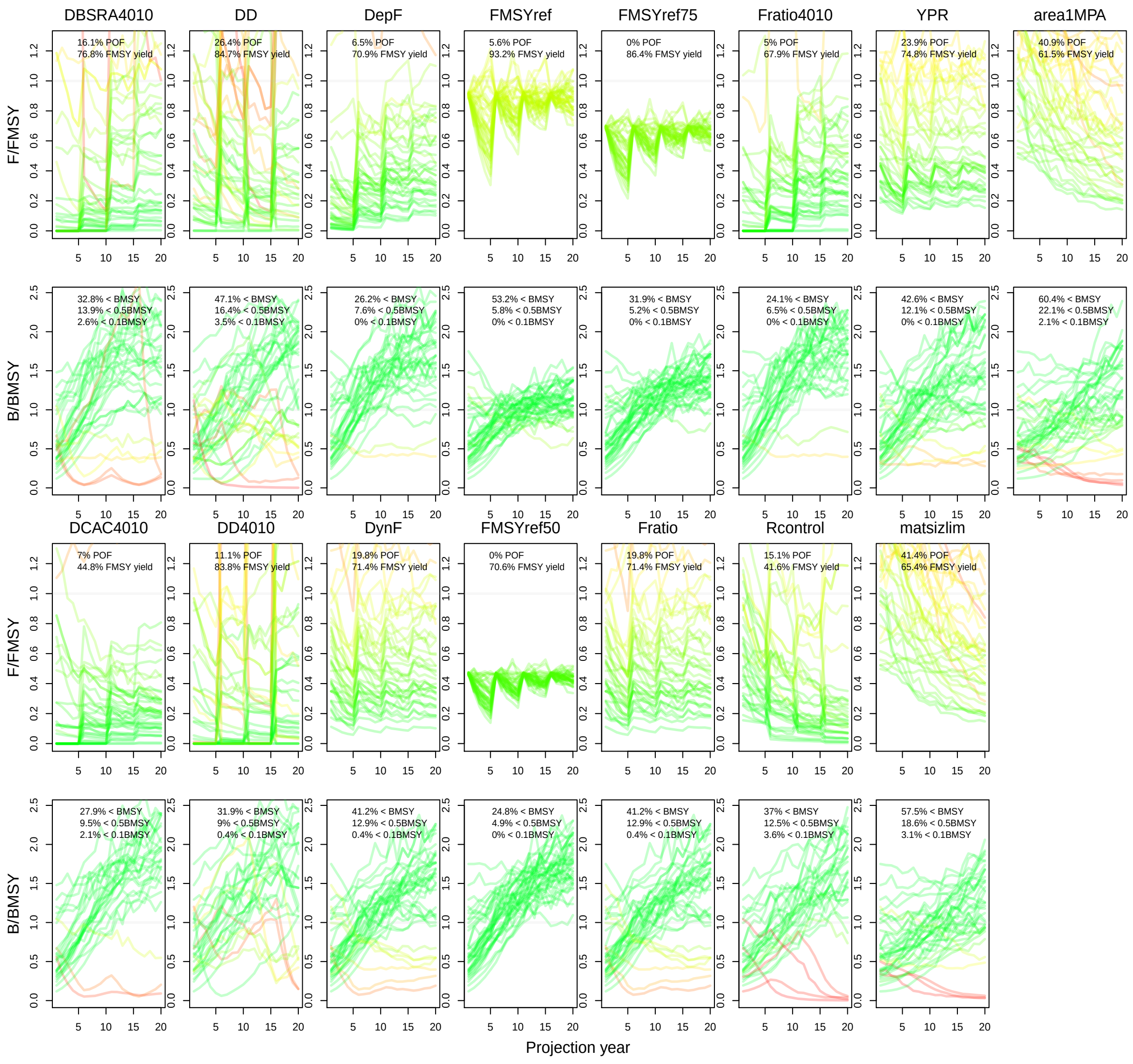
Our new subsetted methods can be used to run a more focused MSE that includes a greater number of simulations for a detailed assessment of performance. Again note that in a real setting it would be advisable to increase the number of simulations further to at least 400.You might also want to increase the number of stochastic samples per method (reps) to a higher number.

```
ourMSE2<-runMSE(ourOM,Targetted$Method,proyears=20,interval=5,nsim=40,reps=1)
```

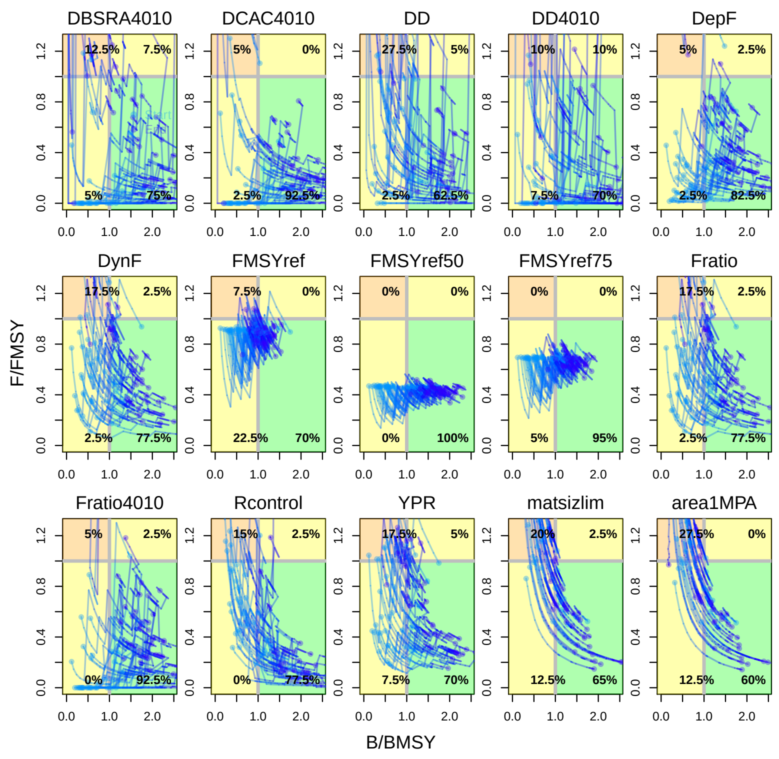
```
## [1] "Loading operating model"
## [1] "Optimizing for user-specified movement"
## [1] "Optimizing for user-specified depletion"
## [1] "Calculating historical stock and fishing dynamics"
## [1] "Calculating MSY reference points"
## [1] "Calculating reference yield - best fixed F strategy"
## [1] "Determining available methods"
## [1] "1/15 Running MSE for DBSRA4010"
## ....................
## [1] "2/15 Running MSE for DCAC4010"
## ....................
## [1] "3/15 Running MSE for DD"
## ....................
## [1] "4/15 Running MSE for DD4010"
## ....................
## [1] "5/15 Running MSE for DepF"
## ....................
## [1] "6/15 Running MSE for DynF"
## ....................
## [1] "7/15 Running MSE for FMSYref"
## ....................
## [1] "8/15 Running MSE for FMSYref50"
## ....................
## [1] "9/15 Running MSE for FMSYref75"
## ....................
## [1] "10/15 Running MSE for Fratio"
## ....................
## [1] "11/15 Running MSE for Fratio4010"
## ....................
## [1] "12/15 Running MSE for Rcontrol"
## ....................
## [1] "13/15 Running MSE for YPR"
## ....................
## [1] "14/15 Running MSE for matsizlim"
## ....................
## [1] "15/15 Running MSE for area1MPA"
## ....................
```

Several detailed plots can provide greater information about exactly how each method performed over the projected time period including Projection and Kobe plots:

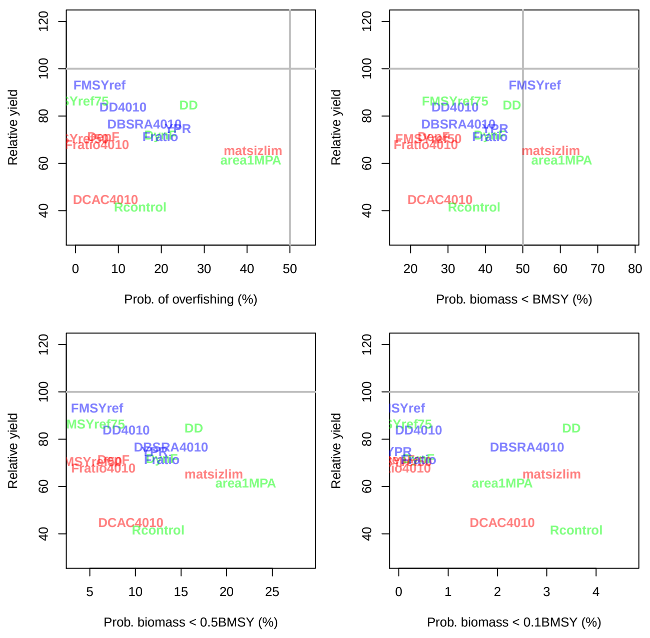
```
Pplot(ourMSE2)
```



```
Kplot(ourMSE2)
```



```
Tplot(ourMSE2)
```

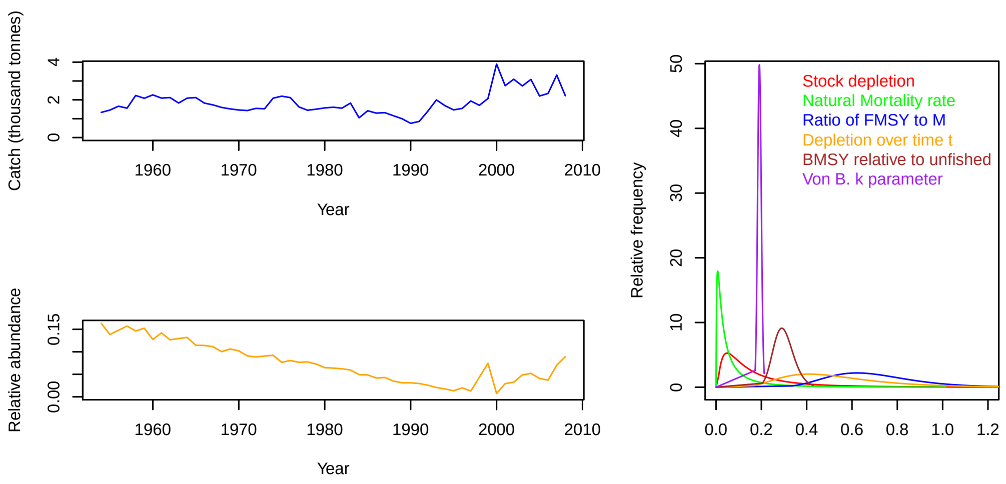


These plots indicate that methods based on a stable F strategy (DynF and Fratio) and the Yield per recruit (YPR) analysis are at the upper limit of the trade-off space (ie all other methods provide worse performance in one or more dimensions). In this case the three methods offer a contrasting trade-off between probability of overfishing and long-term yield. It remains to be seen whether any of these approaches can be applied to the real data for our reef fish...

5.3 Applying methods to our real data

A real DLM data object 'ourReefFish', was loaded into the current R session when we ran the SendAll() function. We can summarise some of the data in this DLM data object using the generic function summary():

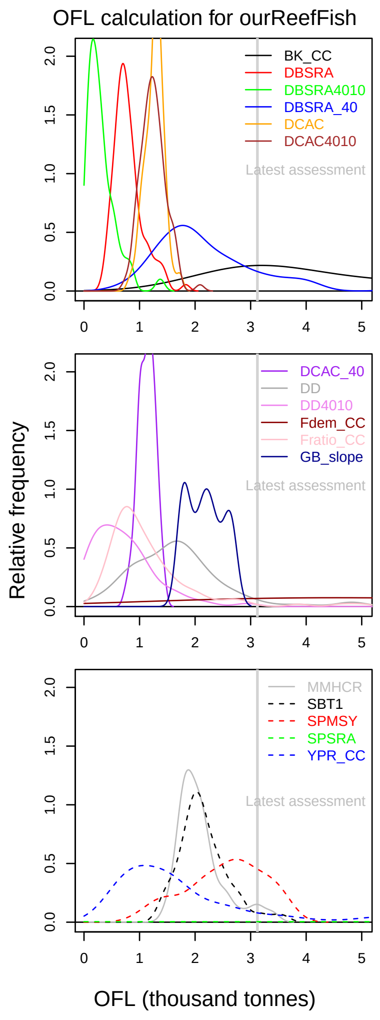
```
summary(ourReefFish)
```



The Can() function reveals that a range of methods are available but not the three best performing methods identified by the MSE (DynF, Fratio, YPR). Nonetheless we can calculate and plot the OFLs for the available methods:

```
ourReefFish<-getQuota(ourReefFish)

## [1] "Method MMHCR produced greater than 50% NA values"
```

```
plot(ourReefFish)
```

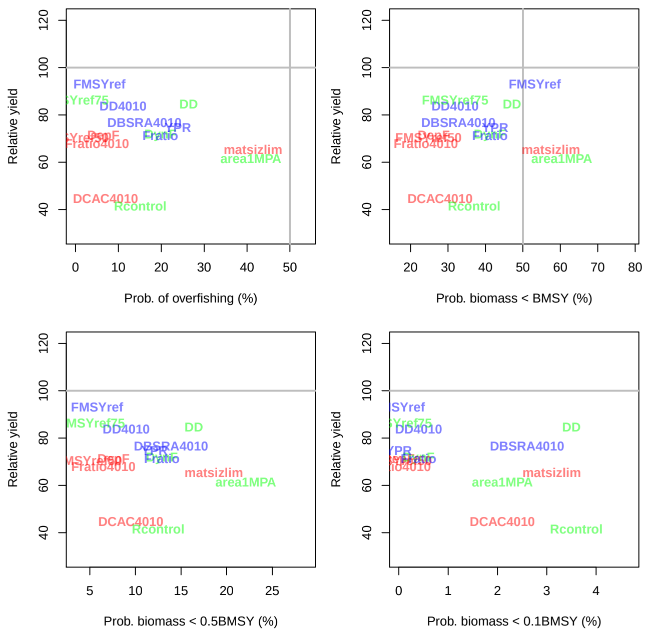


The Needed() function shows that there is only one remaining data requirement to make each of these methods work, a current estimate of abundance (Abun, a slot in the DLM data object. see ?DLM). While this may seem like rather a demanding data requirement it is worth remembering that DynF, Fratio and YPR outperformed the remaining methods on average despite fairly bad current abundance information that could have observation error with a CV of up to 100 per cent and a bias sampled from a distribution with CV of 75 percent:

```
ourOM@Btcv

## [1] 0.5 1.0

ourOM@Btbiascv

## [1] 0.75
```

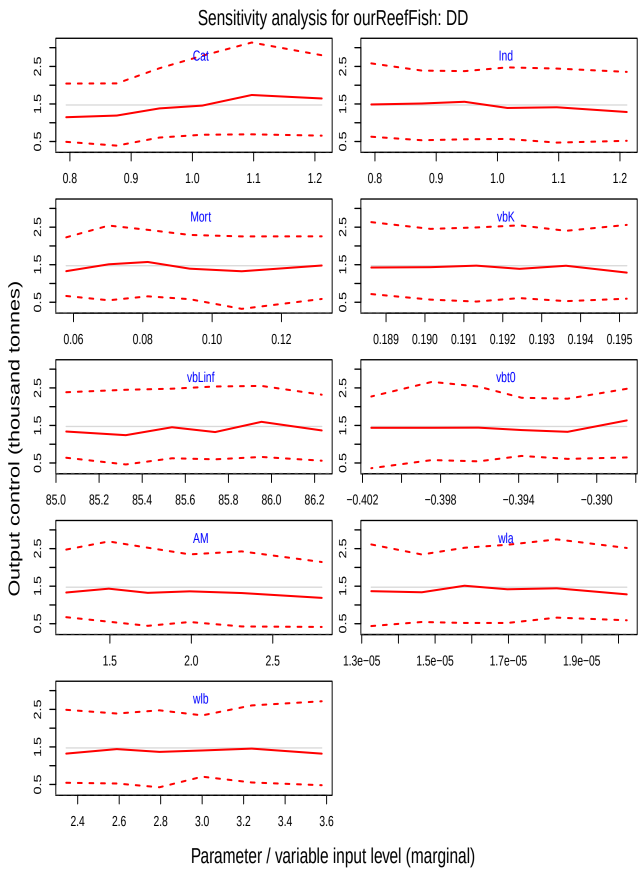
In other words our MSE generated observations of current biomass that could easily be double or half the true level across the whole time series and were very noisy. The question now is whether gathering such data would be worthwhile (e.g. a systematic fishery independent survey). To refresh our memory we can re-plot he tradeoffs of the targetted MSE:

```
Tplot(ourMSE2)
```



If we focus on output controls there are a cluster of methods that offer comparable performance that are available for our real data such as the delay-difference stock assessment (DD). We can use sensitivity testing to better understand how fragile quota recommendations are to changes in our data inputs:

```
ourReefFish<-Sense(ourReefFish,'DD')
```



In this case it is a toss-up between them. Both show primary sensitivity to bias in reported catches (fairly obviously) and little sensitivity to the remaining inputs. In this case it might be necessary to try an alternative run of the operating model given other credible parameters to see if we can distinguish between these methods.

5.4 What have we learned?

In this simple walkthrough we have established what methods work best for our stock, fishery and observation type. It was possible to establish the frailties of these methods by examining what simulated parameters drive yield and probability of overfishing (using the plot() function). Our application to real data produced actual OFL recommendations for methods that were available. At least three methods could not be applied that the MSE indicated could provide benefits in terms of both yield and limiting overfishing. We know what data are necessary to make these work but have yet to decide whether collecting these data is worthwhile. Above all, the approach is transparent and reproducible.

Depending on how utility is characterised, it may be possible to establish the cost-efficacy of future data-collection based on the long-term yield differential of the methods that are available and those that need additional data.

6 Designing new methods

DLMtool was designed to be extensible in order to promote the development of new methods. In this section we design a series of new methods that include spatial controls and input controls in the form of age-restrictions. The central requirement of any method is that it can be applied to a DLM data object using the function sapply (sfSapply() in parallel processing). DLM data object have a single position x for each data entry, e.g. one value for natural mortality rate (DLM@M[x]), a single vector of historical catches (DLM@Cat[x,]) etc. In the MSE analysis this is extended to nsim positions. It follows that any method arranged to function sapply(x,Method,DLM) will work. For example we can get 5 stochastic samples of the OFL for the demographic FMSY method paired to catch-curve analysis FdemCC applied to a real data-limited data object for red snapper using:

```
sapply(1,Fdem_CC,Red_snapper,reps=5)
```

```
##        [,1]
## [1,] 16.305
## [2,] 28.186
## [3,]  2.598
## [4,]  3.474
## [5,] 13.177
```

The MSE just populates a DLM data object with many simulations and uses sapply() (or sfSapply() in cluster computing mode) to calculate an management recommendation for each simulation. By making methods compatible with this standard the very same equations are used in both the MSE and the real management advice.

The following new methods illustrate this.

6.1 Average historical catch method

The average historical catch has been suggested as a starting point for setting OFLs in the most data-limited situations (following Restrepo et al. 1998). Here we design such an approach:

```
AvC<-function(x,DLM,reps)rlnorm(reps,log(mean(DLM@Cat[x,],na.rm=T)),0.1)
```

Note that all methods have to be stochastic in this framework which is why we sample from a log-normal distribution with a CV of roughly 10 per cent.

Before the method can be 'seen' by the rest of the DLM package we have to do three more things. The method must be assigned a class based on what outputs it provides. Since this is an output control (quota) based method we assign it class 'DLM quota'. The method must also be assigned to the DLMtool namespace and -if we are using parallel computing- exported to the cluster:

```
class(AvC)<-"DLM quota"
environment(AvC) <- asNamespace('DLMtool')
sfExport("AvC")
```

6.2 Third-highest catch

In some data-limited settings third highest historical catch has been suggested as a possible catch-limit. Here we use a similar approach to the average catch method above (AvC) and take draws from a log-normal distribution with CV of 10 per cent:

```
THC<-function(x,DLM,reps){
  rlnorm(reps,log(DLM@Cat[x,order(DLM@Cat[x,],decreasing=T)[3]]),0.1)
}
class(THC)<-"DLM quota"
environment(THC) <- asNamespace('DLMtool')
sfExport("THC")
```

6.3 Fishing starting at age 5

To simulate input controls that aim to alter the age-vulnerability to fishing it is possible to design a method of class 'DLM size'. These simply describe a vector of fractional vulnerability (0-1) across ages. In this example we freeze fishing on age-classes 1-4 and maintain fishing for ages 5+:

```
agelim5<-function(x,DLM)c(rep(0,4),rep(1,DLM@MaxAge-4))
class(agelim5)<-"DLM size"
environment(agelim5) <- asNamespace('DLMtool')
sfExport("agelim5")
```

Note that these approaches still require an 'x' argument even if they don't make use of it (ie they are the same regardless of the data or simulated data).

6.4 Reducing fishing rate in area 1 by 50 per cent

Spatial controls operate similarly to the age/size based controls: a vector of length 2 (the spatial simulator is a 2-box model) that indicates the fraction of current spatial catches. This is an early version of spatial control in the DLMtool so it only deals with reductions in catch and does not simulate reallocation of fishing effort. In this example we reduce catches in area 1 by 50 percent and assign the method class 'DLM space'.

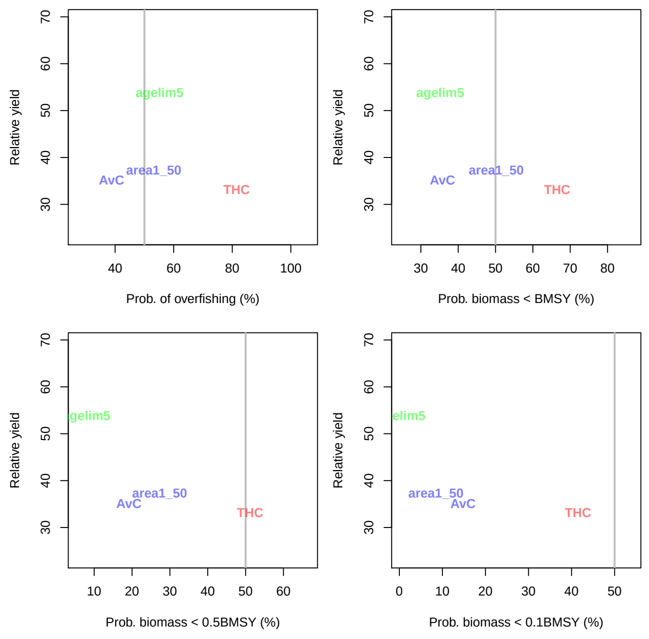
```
area1_50<-function(x,DLM)c(0.5,1)
class(area1_50)<-"DLM space"
environment(area1_50) <- asNamespace('DLMtool')
sfExport("area1_50")
```

6.5 Applying the new methods

Our methods are now compatible with all of the DLMtool functionality. Lets run a quick MSE and see how they fare:

```
new_methods<-c("AvC","THC","agelim5","area1_50")
OM<-new('OM',Porgy, Generic_IncE, Imprecise_Unbiased)
PorgMSE<-runMSE(OM,new_methods,maxF=1,nsim=20,reps=1,proyears=20,interval=5)

## [1] "Loading operating model"
## [1] "Optimizing for user-specified movement"
## [1] "Optimizing for user-specified depletion"
## [1] "Calculating historical stock and fishing dynamics"
## [1] "Calculating MSY reference points"
## [1] "Calculating reference yield - best fixed F strategy"
## [1] "Determining available methods"
## [1] "1/4 Running MSE for AvC"
## ....................
## [1] "2/4 Running MSE for THC"
## ....................
## [1] "3/4 Running MSE for agelim5"
## ....................
## [1] "4/4 Running MSE for area1_50"
## ....................
```

```
Tplot(PorgMSE)
```

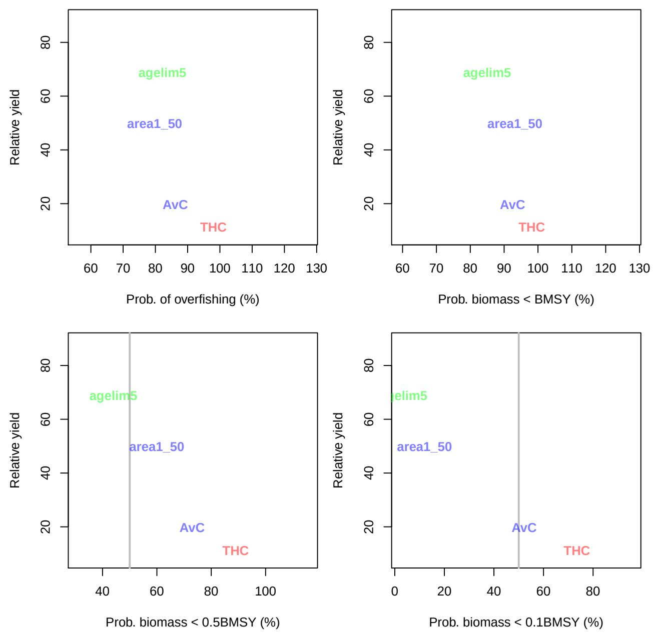


What if starting depletion were different, e.g. likely to be under BMSY?

```
OM@D

## [1] 0.05 0.60
```

```
OM@D<-c(0.05,0.3)
PorgMSE2<-runMSE(OM,new_methods,maxF=1,nsim=20,reps=1,proyears=20,interval=5)

## [1] "Loading operating model"
## [1] "Optimizing for user-specified movement"
## [1] "Optimizing for user-specified depletion"
## [1] "Calculating historical stock and fishing dynamics"
## [1] "Calculating MSY reference points"
## [1] "Calculating reference yield - best fixed F strategy"
## [1] "Determining available methods"
## [1] "1/4 Running MSE for AvC"
## ....................
## [1] "2/4 Running MSE for THC"
## ....................
## [1] "3/4 Running MSE for agelim5"
## ....................
## [1] "4/4 Running MSE for area1_50"
## ....................
```

```
Tplot(PorgMSE2)
```



Putting aside the likelihood of implementing a perfect knife-edge vulnerability to fishing at age 5, it appears that we have a clear winner in agelim5 even under different starting depletion levels. Third highest catch on the other hand appears risky to say the least. You could try some other starting depletion levels to see under what circumstances the trade-off space changes dramatically.

7 Managing real data

DLMtool has a series of functions to make importing data and applying data-limited methods relatively straightforward. There are two approaches: (1) fill out a .csv data file in excel or a text editor and automatically create a DLM data object (class DLM) or (2) create a blank DLM data object in R and populate it in R.

7.1 Importing data

Probably the easiest way to get your data into the DLMtool is to populate a .csv datafile. These files have a line for each slot of the DLMdata object e.g:

```
slotNames('DLM')

##  [1] "Name"       "Year"       "Cat"        "Ind"        "Rec"
##  [6] "t"          "AvC"        "Dt"         "Mort"       "FMSY_M"
## [11] "BMSY_B0"    "Cref"       "Bref"       "Iref"       "AM"
## [16] "LFC"        "LFS"        "CAA"        "Dep"        "Abun"
## [21] "vbK"        "vbLinf"     "vbt0"       "wla"        "wlb"
## [26] "steep"      "CV_Cat"     "CV_Dt"      "CV_AvC"     "CV_Ind"
## [31] "CV_Mort"    "CV_FMSY_M"  "CV_BMSY_B0" "CV_Cref"    "CV_Bref"
## [36] "CV_Iref"    "CV_Rec"     "CV_Dep"     "CV_Abun"    "CV_vbK"
## [41] "CV_vbLinf"  "CV_vbt0"    "CV_AM"      "CV_LFC"     "CV_LFS"
## [46] "CV_wla"     "CV_wlb"     "CV_steep"   "sigmaL"     "MaxAge"
## [51] "Units"      "Ref"        "Ref_type"   "Log"        "params"
## [56] "PosMeths"   "Meths"      "OM"         "Obs"        "quota"
## [61] "quotabias"  "Sense"      "CAL_bins"   "CAL"
```

You do not have to enter data for every line of the data file, if data are not available simply put an 'NA' next to any given field.

A number of example .csv files can be found in the directory where the DLMtool package was installed:

```
DLMDataDir()

## [1] "C:/Users/Tom/AppData/Local/Temp/Rtmp6v9C7G/Rinst12d42e8371d7/DLMtool/"
```

To get data from a .csv file you need only specify its location e.g new('DLM',"I:/Mackerel.csv"):

7.2 Populating a DLM data object in R

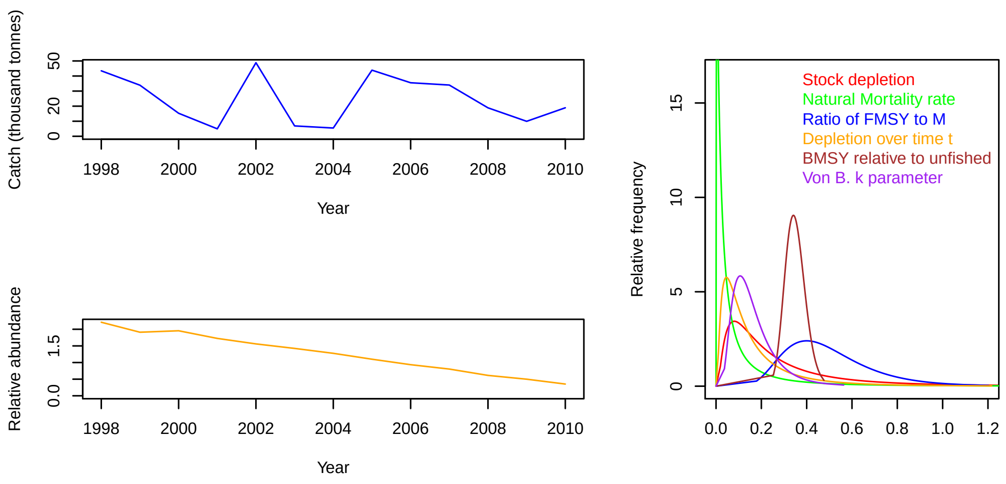
Alternatively you can create a blank DLM data object and fill the slots directly in R. E.g:

```
Madeup<-new('DLM')                                 #  Create a blank DLM object

## [1] "Couldn't find specified csv file in DLM/Data folder, blank DLM object created"

Madeup@Name<-'Test'                                #  Name it
Madeup@Cat<-matrix(20:11*rlnorm(10,0,0.2),nrow=1)  #  Generate fake catch data
Madeup@Units<-"Million metric tonnes"              #  State units of catch
Madeup@AvC<-mean(Madeup@Cat)                       #  Average catches for time t (DCAC)
Madeup@t<-ncol(Madeup@Cat)                         #  No. yrs for Av. catch (DCAC)
Madeup@Dt<-0.5                                     #  Depletion over time t (DCAC)
Madeup@Dep<-0.5                                    #  Depletion relative to unfished
Madeup@Mort<-0.1                                   #  Natural mortality rate
Madeup@Abun<-200                                   #  Current abundance
Madeup@FMSY_M<-0.75                                #  Ratio of FMSY/M
Madeup@AM<-3.5                                     #  Age at maturity
Madeup@BMSY_B0<-0.35                               #  BMSY relative to unfished
```

7.3 Working with DLM data objects

A generic summary function is available to visualize the data in a DLMdata object:

```
summary(Atlantic_mackerel)
```



You can see what methods can and can't be applied given your data and also what data are needed to get methods working:

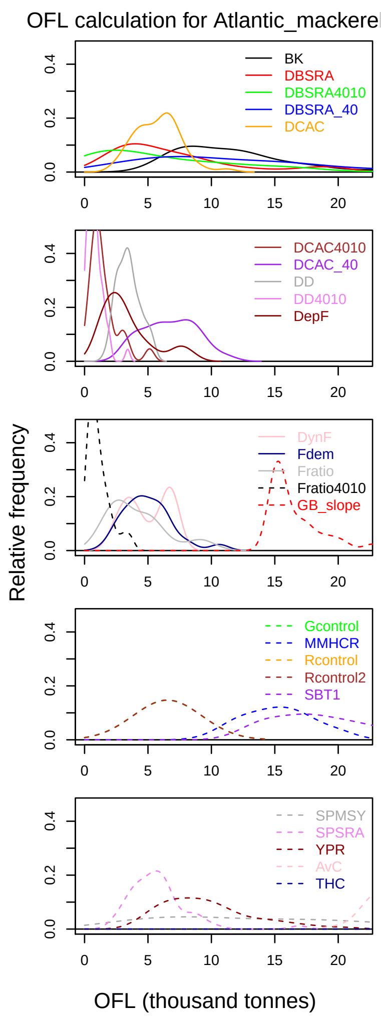
```
Can(Atlantic_mackerel)

##  [1] "BK"         "DBSRA"      "DBSRA4010"  "DBSRA_40"   "DCAC"
##  [6] "DCAC4010"   "DCAC_40"    "DD"         "DD4010"     "DepF"
## [11] "DynF"       "Fdem"       "Fratio"     "Fratio4010" "GB_slope"
## [16] "Gcontrol"   "MMHCR"      "Rcontrol"   "Rcontrol2"  "SBT1"
## [21] "SPMSY"      "SPSRA"      "YPR"        "AvC"        "THC"
## [26] "matsizlim"  "agelim5"    "area1MPA"   "area1_50"

Cant(Atlantic_mackerel)

##        [,1]         [,2]
##  [1,] "BK_CC"       "Insufficient data"
##  [2,] "BK_ML"       "Insufficient data"
##  [3,] "DBSRA_ML"    "Insufficient data"
##  [4,] "DCAC_ML"     "Insufficient data"
##  [5,] "FMSYref"     "Insufficient data"
##  [6,] "FMSYref50"   "Insufficient data"
##  [7,] "FMSYref75"   "Insufficient data"
##  [8,] "Fdem_CC"     "Insufficient data"
##  [9,] "Fdem_ML"     "Insufficient data"
## [10,] "Fratio_CC"   "Insufficient data"
## [11,] "Fratio_ML"   "Insufficient data"
## [12,] "GB_CC"       "Produced all NA scores"
## [13,] "GB_target"   "Produced all NA scores"
## [14,] "SBT2"        "Produced all NA scores"
## [15,] "SPSRA_ML"    "Insufficient data"
## [16,] "YPR_CC"      "Insufficient data"
## [17,] "YPR_ML"      "Insufficient data"
```

```
Needed(Atlantic_mackerel)

##  [1] "BK_CC: CAA"               "BK_ML: CAL"
##  [3] "DBSRA_ML: CAL"            "DCAC_ML: CAL"
##  [5] "FMSYref: OM"              "FMSYref50: OM"
##  [7] "FMSYref75: OM"            "Fdem_CC: CAA"
##  [9] "Fdem_ML: CAL"             "Fratio_CC: CAA"
## [11] "Fratio_ML: CAL"           "GB_CC: Cref"
## [13] "GB_target: Cref, Iref" "SBT2: Rec, Cref"
## [15] "SPSRA_ML: CAL"            "YPR_CC: CAA"
## [17] "YPR_ML: CAL"
```

Spatial methods and age-vulnerability methods can be MSE tested but are a management recommendation in themselves. DLM quota methods however can be calculated and plotted using getQuota() function:

```
Atlantic_mackerel<-getQuota(Atlantic_mackerel,reps=48)
```

```
plot(Atlantic_mackerel)
```



8 Efficacy test of an hypothetical Marine Protected Area

Marine Protected Areas (MPAs) have been suggested as a management tool for limiting the impact of overfishing. In this section we create a Marine Reserve (no fishing) and examine performance given different reserve size and exchange rates.

8.1 Adapting an existing Stock object

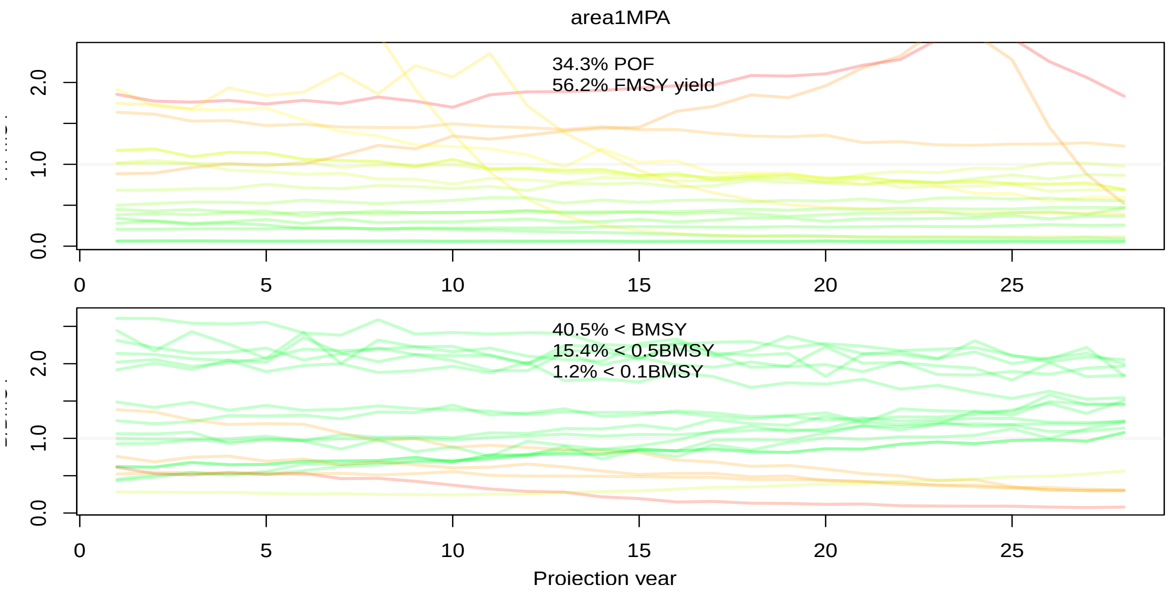
We use the Snapper stock object as a template and modify two variables, the probability of individuals staying in area 1 (the MPA) and the size of area 1 (the MPA):

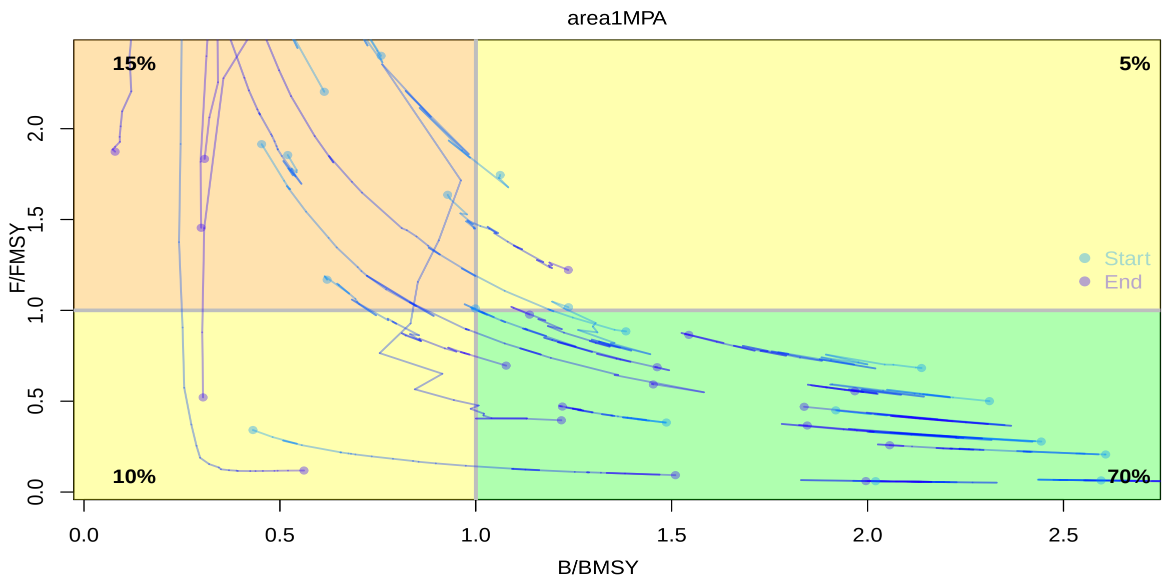
```
Rock<-Rockfish
Rock@Prob_staying<-c(0.9,0.999)
Rock@Frac_area_1<-c(0.05,0.5)
```

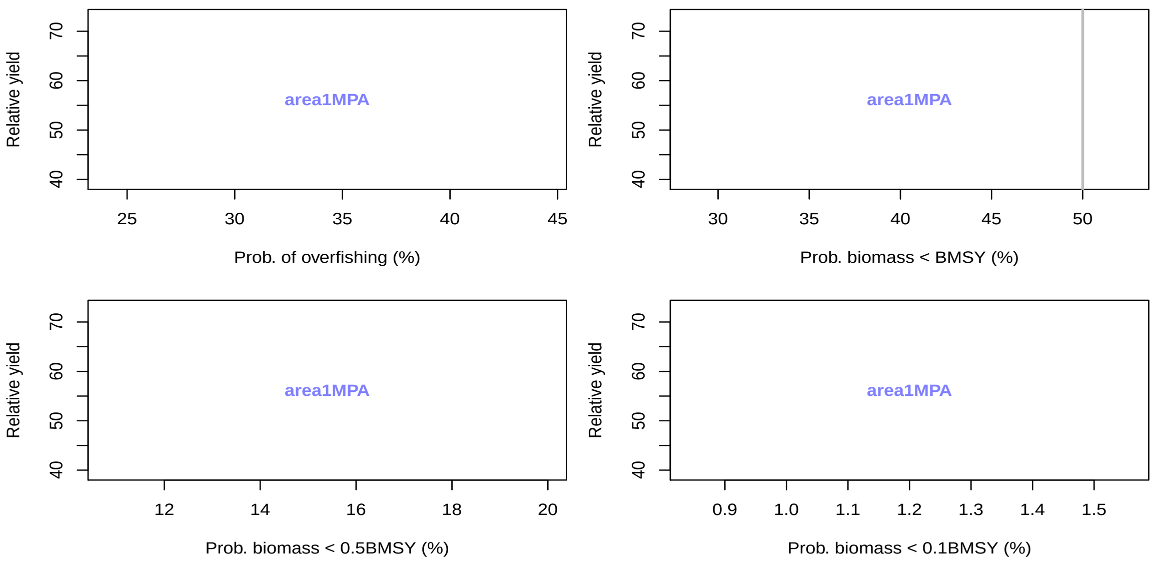
We now run the MSE for the area1MPA method (catches maintained at current levels in area 2 and set to zero in area 1).

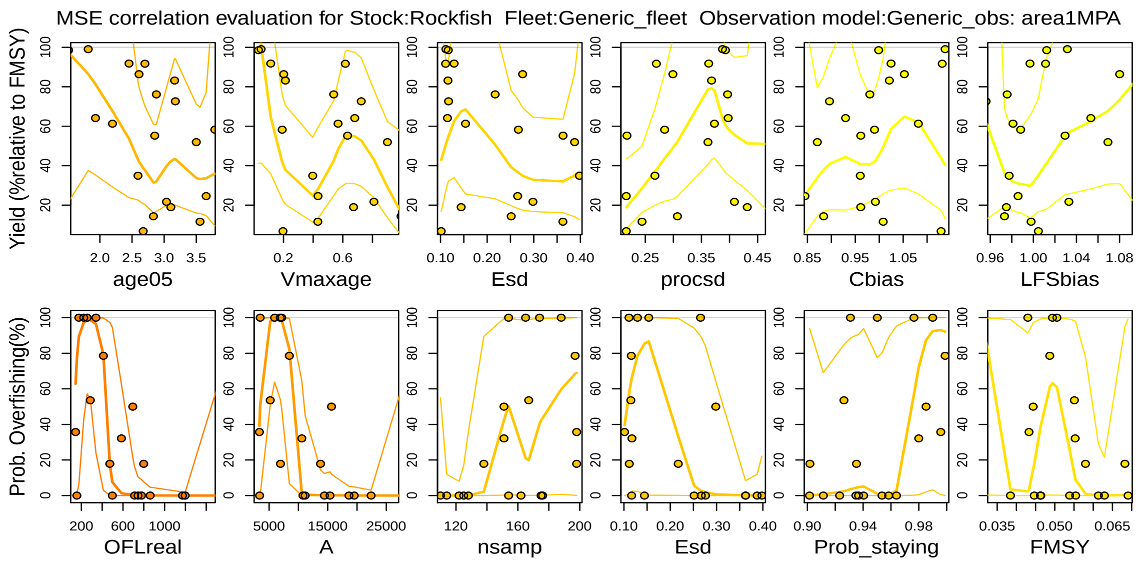
```
RockMPA<-runMSE(new('OM',Rock,Generic_fleet,Generic_obs),"area1MPA",nsim=20)

## [1] "Loading operating model"
## [1] "Optimizing for user-specified movement"
## [1] "Optimizing for user-specified depletion"
## [1] "Calculating historical stock and fishing dynamics"
## [1] "Calculating MSY reference points"
## [1] "Calculating reference yield - best fixed F strategy"
## [1] "Determining available methods"
## [1] "1/1 Running MSE for area1MPA"
## ............................

summary(RockMPA)

##     Method Yield       POF        P10        P50        P100
## 1 area1MPA 56.16 31.45 34.29 40.38 1.25 5.59 15.36 29.03 40.54 43.05
```

```
plot(RockMPA)
```


area1MPA
34.3% POF
56.2% FMSY yield
40.5% < BMSY
15.4% < 0.5BMSY
1.2% < 0.1BMSY
0.0
1.0
2.0
0
5
10
15
20
25
Projection year


area1MPA
15%
5%
10%
70%
Start
End
F/FMSY
B/BMSY
0.0
0.5
1.0
1.5
2.0
2.5





The sensitivity plots reveal that the fraction of the stock in area1 is critical to determining the effect of the MPA in terms of probability of overfishing but has much less impact on the long-term yield, despite reducing catches by an increasingly large amount as fraction in area 1 increases. In terms of yield, recruitment compensation (steepness, h) is a stronger determinant of yield than the size of the MPA. Interestingly the probability of individuals staying in area 1 does not feature in the top six most correlated variables for either yield or probability of overfishing.

9 Limitations

9.1 Idealised observation models for catch composition data

Currently DLMtool simulates catch-composition data from the true simulated catch composition data via a multinomial distribution and some effective sample size. This observation model may be unrealistically well behaved and favour those approaches that use these data.

9.2 Harvest control rules must be integrated into data-limited methods

In this version of DLMtool harvest control rules (e.g. the 40-10 rule) must be written into a data-limited method. There is currently no functionality whereby a given HCR can be applied to a given class of method. The reason for this is that it would require further subclasses. For example the 40-10 rule may be appropriate for the output of DBSRA but it would not be appropriate for some of the algorithmic management procedures such as DynF that already incorporate throttling of quota recommendations according to stock depletion.

9.3 Natural mortality rate at age

The current simulation assumes constant M with age. Age-specific M will be added soon.

9.4 Ontogenetic habitat shifts

Since the operating model simulated two areas, it is possible to prescribe a log-linear model that moves fish from one are to the other as they grow older simulating, for example, the ontogenetic shift of groupers from near shore waters to offshore reefs. Currently this feature is in development.

9.5 Implementation error

In this edition of DLMtool there is no implementation error. The only imperfection between a management recommendation and the simulated quota comes in the form of the MaxF argument that limits the maximum fishing mortality rate on any given age-class in the operating model. The default is 0.8 which is high for all but the shortest living fish species.

10 References

Carruthers, T.R., Punt, A.E., Walters, C.J., MacCall, A., McAllister, M.K., Dick, E.J., Cope, J. 2014. Evaluating methods for setting catch-limits in data-limited fisheries. Fisheries Research. 153, 48-68.

Costello, C., Ovando, D., Hilborn, R., Gains, S.D., Deschenes, O., Lester, S.E., 2012. Status and solutions for the world???s unassessed fisheries. Science. 338, 517-520. Deriso, R. B., 1980. Harvesting Strategies and Parameter Estimation for an Age-Structured Model. Can. J. Fish. Aquat. Sci. 37, 268-282.

Dick, E.J., MacCall, A.D., 2011. Depletion-Based Stock Reduction Analysis: A catch-based method for determining sustainable yields for data-poor fish stocks. Fish. Res. 110, 331-341.

Geromont, H.F. and Butterworth, D.S. 2014. Complex assessment or simple management procedures for efficient fisheries management: a comparative study. ICES J. Mar. Sci. doi:10.1093/icesjms/fsu017

MacCall, A.D., 2009. Depletion-corrected average catch: a simple formula for estimating sustainable yields in data-poor situations. ICES J. Mar. Sci. 66, 2267-2271.

Restrepo, V.R., Thompson, G.G., Mace, P.M., Gabriel, W.L., Low, L.L., MacCall, A.D., Methot, R.D., Powers, J.E., Taylor, B.L., Wade, P.R., Witzig, J.F.,1998. Technical Guidance On the Use of Precautionary Approaches to Implementing National Standard 1 of the Magnuson-Stevens Fishery Conservation and Management Act. NOAA Technical Memorandum NMFS-F/SPO-31. 54 pp.