DLMtool **CHEAT SHEET 1**

Getting Started

Install package: install.packages('DLMtool')

User Guide: userguide()

Cheat Sheets: cheatsheets()

Website: http://datalimitedtoolkit.org

Report Issues: https://github.com/DLMtool/DLMtool/issues

Acronyms

DLMtool	Data-Limited Methods Toolkit
MP	Management Procedure
MSE	Management Strategy Evaluation
OM	Operating Model
PM	Performance Metrics

Main Object Classes

Class	Contents
Stock	Biological Properties
Fleet	Exploitation Properties
Obs	Observation Error
Imp	Implementation Error
OM	Operating Model
MSE	Management Strategy Evaluation Results
MP	Management Procedure

OM = Stock + Fleet + Obs + Imp

Find Available Objects: avail('Object Class')

e.g. avail('Stock')
avail('Fleet')
avail('Obs')
...
avail('MP')

More Objects: DLMextra()

Slot Names: slotNames('Object Class')

e.g. slotNames('Stock')

Create New OM

Blank OM: OM <- new('OM')

New OM from available objects: OM <- new('OM', 'Stock', 'Fleet', 'Obs', 'Imp')

e.g. OM <- new('OM', Albacore, Generic_Fleet, Generic_Obs, Perfect_Imp)

Initialize Excel OM and OM Report: OMinit('myOM')

Import OM from Excel: OM <- XL2OM('myOM')

Generate OM Report: OMdoc()

Customize OM

Sketch Historical Fishing: Fleet/OM <- ChooseEffort(Fleet/OM)

Sketch Selectivity: Fleet <- ChooseSelect(Fleet, FstYr = ...)

Sketch Age Specific *M*: OM <- ChooseM(OM)

Sketch Length Specific *M*: OM <- ChooseM(OM, 'Length')

Predicting Life-History Parameters:

e.g. OM <- new('OM')
OM@Species <- 'Scomber japonicus'
OM <- LH2OM(OM)

Custom Parameters:

e.g. OM <- new('OM', Albacore, Generic_Fleet, Generic_Obs, Perfect_Imp)
OM@cpars$M <- rlnorm(OM@nsim, log(0.2), 0.05)

Remove Process and Observation Error: OM <- tinyErr(OM)

Replace OM Component: OM <- Replace(OM, Blue_shark)

Examine OM

Plot OM Components:

e.g. plot(Albacore)
plot(Generic_Fleet)

Plot OM: plot(OM)

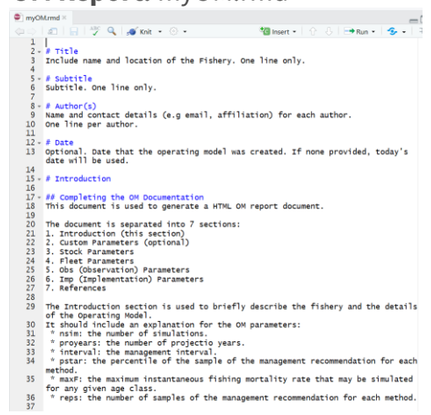
Plot Existing MPA: plotMPA(OM)

Plot *M*: plotM(OM)

Plot Selectivity: plotSelect(OM)

OM Excel: myOM.xlsx

OM Report: myOM.rmd



Management Procedures

MP Type	Returns
Output	TAC (total allowable catch)
Input	TAE, SL, Spatial (total allowable effort, size limit, spatial closure)
Mixed	Combination of Output and Input
Reference	TAC (assuming perfect data)

Find MP type: MPtype()

e.g. MPtype(c('AvC', 'curE', 'matlenlim', 'FMSYref'))

Run MSE

Run MSE: runMSE()

e.g. MSE <- runMSE(OM,
MPs=c('AvC', 'curE', 'matlenlim', 'FMSYref'))

Run MSE in parallel: runMSE(parallel=TRUE)

e.g. MSE <- runMSE(OM,
MPs=c('AvC', 'curE', 'matlenlim', 'FMSYref'),
parallel=TRUE)

Check Convergence: Converge(MSE)

Run Historical Simulations:

Hist <- runMSE(OM, Hist=TRUE)

DLMtool **CHEAT SHEET 2**

Performance Metrics

Available PM Functions: avail('PM')

Calculate PM:

e.g. MSE <- runMSE()
P50(MSE)

Example Custom PM:

Calculate Probability F < 2 x FMSY in first 5 years:

```
myPM <- function(MSEobj=NULL, Ref=2, Yrs=5) {
  Yrs <- ChkYrs(Yrs, MSEobj) # validate years
  PMobj <- new('PMobj') # create empty PM object
  PMobj@Name <- paste0('Probability F/FMSY < ', Ref) # name of PM
  PMobj@Caption <- paste0('Probability F/FMSY < ', Ref) # caption
  PMobj@Stat <- MSEobj@F_FMSY[, , Yrs[1]:Yrs[2]] # statistic
  PMobj@Ref <- Ref # save Reference
  PMobj@Prob <- calcProb(PMobj@Stat < PMobj@Ref, MSEobj) # prob.
  PMobj@Mean <- calcMean(PMobj@Prob) # average prob.
  PMobj@MPs <- MSEobj@MPs # record MPs
  PMobj  # return PM object
}
class(myPM) <- 'PM' # assign to class 'PM'
summary(MSE, 'myPM') # calculate performance
TradePlot(MSE, 'myPM', 'P50') # trade-off plot with new PM
```

Examine MSE Results

Summary Results: summary(MSE)

Trade-Off Plots: TradePlot(MSE, PMs)

e.g. TradePlot(MSE, 'P50', 'AAVY')
Tplot(MSE)
Tplot2(MSE)
NOAA_plot(MSE)

Projection Plots:

Pplot(MSE)
Pplot2(MSE)

Kobe Plot: Kplot(MSE)

Value of Information:

VOI(MSE)
VOI2(MSE)
VOIplot(MSE)
VOIplot2(MSE)

Other Plots:

COSEWIC_Hplot(MSE)
Cplot(MSE)
DFO_plot(MSE)
DFO_plot2(MSE)
DFO_proj(MSE)
IOTC_plot(MSE)
PWhisker(MSE)
wormplot(MSE)

Subset MSE

Subset by MP: MSE2 <- Sub(MSE, MPs= ...)

e.g. MSE <- runMSE()
stats <- summary(MSE)
accept <- which(stats$P50 > 0.7)
acceptMPs <- stats[accept, 'MP')
subMSE <- Sub(MSE, MPs=acceptMPs)

Subset by Simulation: MSE2 <- Sub(MSE, sims= ...)

e.g. below <- MSE@OM$M < median(MSE@OM$M)
subMSE <- Sub(MSE, sims=below)

Fishery Data Object

Example Data: avail('Data')

Blank Data: Data <- new('Data')

Initialize Data Excel: DataInit()

Import Data from Excel: Data <- XL2Data()

Plot Data: summary(Data)

Write Data: Data2csv(Data, 'Data.csv')

Evaluating OM

Compare Simulated and Actual Data:

Turing(OM, Data)

Management Procedures

Available MPs: Can(Data)

Unavailable MPs: Cant(Data)

Feasible MPs: ?Fease

e.g. All Management Options: Fease(Data) = Can(Data)

TAC Only: Fease(Data, TAE=FALSE,
SL=FALSE, Spatial=FALSE)

Size Reg. Only: Fease(Data, TAC=FALSE,
TAE=FALSE, Spatial=FALSE)

Custom MPs

Averaging MPs: myMP <- makeMeanMP(MP Names)

e.g. avgMP <- makeMeanMP(c('BK', 'DBSRA', 'Fadapt', 'Rcontrol')
MSE <- runMSE(DLMtool::testOM, MPs=c ('BK', 'DBSRA',
'Fadapt', 'Rcontrol', 'avgMP')
Tplot(MSE)

Pseudo-Code to create new MP:

```
AvCatchMP <- function(x, Data, reps=100, plot=FALSE) {
  AvC <- Data@AvC[x] # access element x from Data object slot
  ...
  Rec <- new('Rec') # create object of class Rec
  # slotNames("Rec")
  Rec@TAC <- AvC # populate one or more Rec slots
  Rec # return Rec object
}
class('AvCatchMP') <- 'MP'
```

Apply MPs

Apply MP: runMP(Data, 'MP Name')

e.g. All Available MPs: runMP(Atlantic_mackerel)
TAC <- runMP(Atlantic_mackerel, 'AvC')@TAC

Plot TACs:

e.g. Atlantic_mackerel <- runMP(Atlantic_mackerel)
boxplot(Atlantic_mackerel)

Posterior Predicted Data

Generate Predicted Data from MP application:

e.g. MSE <- runMSE(MPs="DCAC", PPD=TRUE)
Predicted_Ind <- MSE@Misc[[1]]@Ind
matplot(t(Predicted_Ind), type='l',
xlab='Projected Year', ylab='Index value')