Extracting purse-seine catch and length compostion data for bigeye in 2000-2021

Haikun Xu

2022-06-10

This example code demonstrates how to extract the purse-seine catch and length composition data for the stock assessment of bigeye tuna in the eastern Pacific Ocean. Data are extracted for bigeye between 2000 and 2021 based on the R package *BSE* (version 1.0.0). The package can be installed using devtools::install\_github('HaikunXu/BSE',ref='main'). Fishery definition for this data extraction is based on the benchmark assessment conducted in 2020.

* Step 1: set up some directories and parameters for the extraction

# devtools::install\_github('HaikunXu/BSE',ref='main')   
library(BSE)  
  
# Load the base files (please ask Haikun to get those data)  
raw\_data\_dir <- "D:/OneDrive - IATTC/IATTC/2022/BSE stuff from Cleridy/spp comp programs\_from 2000/Raw data extractions/"  
# the directory where output will be saved  
save\_dir <- "D:/OneDrive - IATTC/IATTC/2022/BSE stuff from Cleridy/BET/"  
yr.start <- 2000  
yr.end <- 2021  
Species <- "BET"  
grow.increments <- grow.increments.betyftskj # the growth increment matrix

* Step 2: lightly process the raw data so that they can be easily used in rest of steps

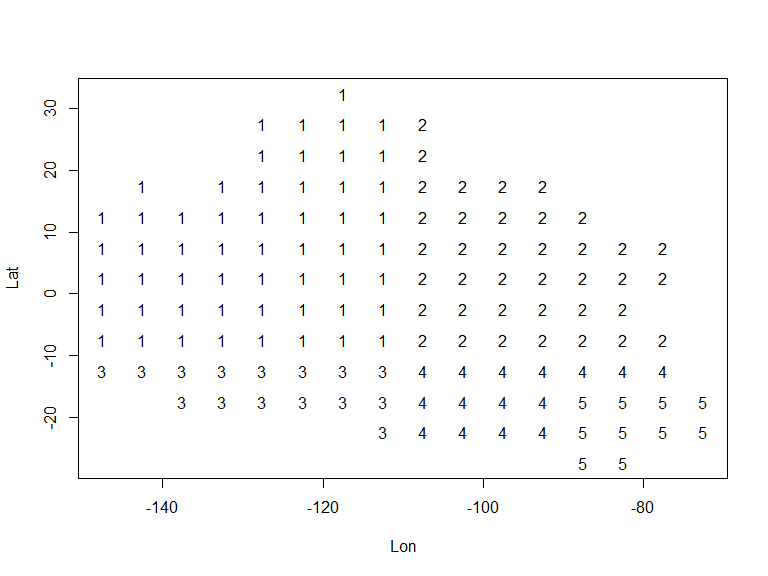
# Get the total unloads for the PS fleet  
total.unlds <- read.unloads.f(raw\_data\_dir,"Unloading2000-2021.txt",yr.start,yr.end)  
# Get the CAE+IDM data  
cae <- read.cae.f(raw\_data\_dir,"CAE-LatLon2000-2021.txt",yr.start,yr.end)  
# Get the length-frequency data (length in millimeters)  
lfmm <- read.lfmmdata.f(raw\_data\_dir,"LengthMM2000-2021.txt")  
# Get the grouped length-frequency output  
lfgrpd <- read.lengthfreq.f(raw\_data\_dir,"LengthFreq2000-2021.txt")

* Step 3: compile the OBJ catch and composition data for BET

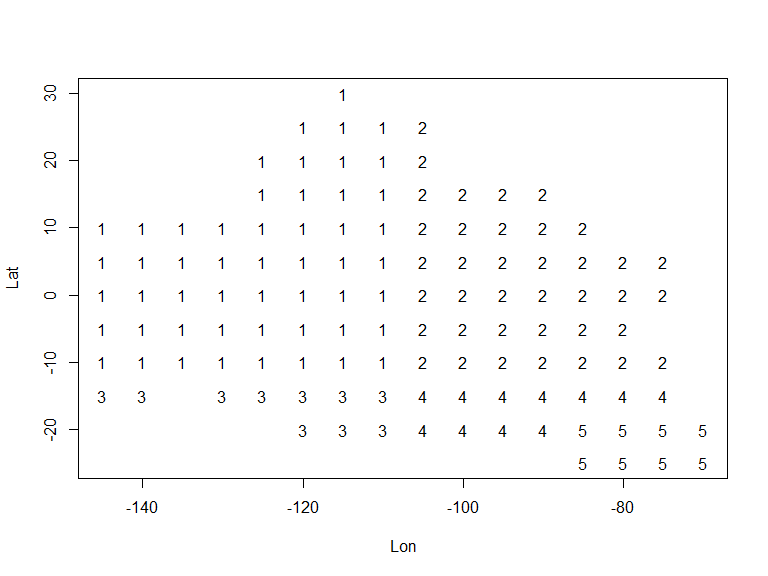
PS <- "OBJ"  
# area.substitution.mat <- area.substitution.mat.BET.movemodel.ALL # for OBJ  
  
area.substitution.mat <- matrix(c(1,3,2,4,5,  
 2,4,1,3,5,  
 3,1,4,2,5,  
 4,5,2,3,1,  
 5,4,2,3,1),  
 ncol = 5, byrow = TRUE) # for OBJ  
  
cae.stratflg <- create.strat.flg.f(cae$latc5,cae$lonc5,is.lwrght=F,cae$month,cae$setype,cae$class,PS=PS,Species=Species)  
lfgrpd.stratflg <- create.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,is.lwrght=T,floor(lfgrpd$moda/100),lfgrpd$setype,lfgrpd$class,PS=PS,Species=Species)

Check the strata definition for OBJ in both cae and lf data sets to make sure that they are correct

check.strat.flg.f(cae$latc5,cae$lonc5,cae.stratflg)



check.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,lfgrpd.stratflg)



Loop through every year between yr.start and yr.end to get catch and composition data for BET in the OBJ fishery

for(year in yr.start:yr.end) {  
 # print(paste0("Year: ",year))  
   
 # print("Step 1: get well estimates")  
 well.estimates <- well.estimates.f(lfgrpd[lfgrpd$year.firstset==year,],lfmm)  
   
 # print("Step 2: get catch estimates")  
 catch.estimates <- get.catch.estimates.f(cae,cae.stratflg,total.unlds,lfgrpd,lfgrpd.stratflg,lfmm,year,2,well.estimates,area.substitution.mat,grow.increments,PS=PS,Species=Species)  
   
 # print("Step 3: get fishery estimates")  
 fishery.estimates <- fishery.estimates.f(catch.estimates$stratum.estimates.withsamps,catch.estimates$stratum.estimates.NOsamps,year,PS=PS,Species=Species)  
   
 assign(paste0("fishery.estimates.", year), fishery.estimates, pos=1)  
}  
# save middle-step data as a record  
save(list=objects(pat="fishery.estimates"),file=paste0(save\_dir,"BET\_",PS,"\_2000-2021.RData"))

Get final OBJ catch and comp output for the stock assessment

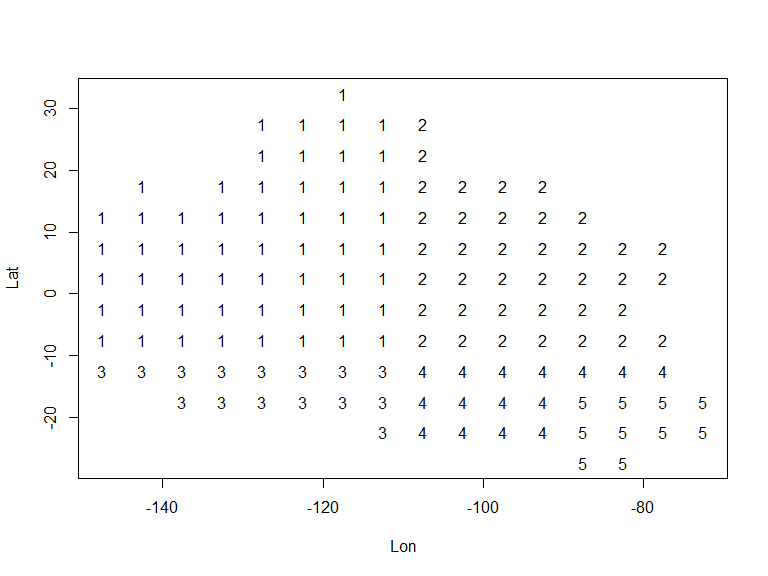
BET.OBJ.Catch.20002021<-compile.catch.output.f(yr.start,yr.end,PS=PS,Species=Species,c("A1","A2","A3","A4","A5")) # five OBJ fisheries  
BET.OBJ.Comp.20002021<-compile.sizecomps.output.f(yr.start,yr.end,PS=PS,Species=Species)

* Step 4: compile the NOA catch and composition data for BET

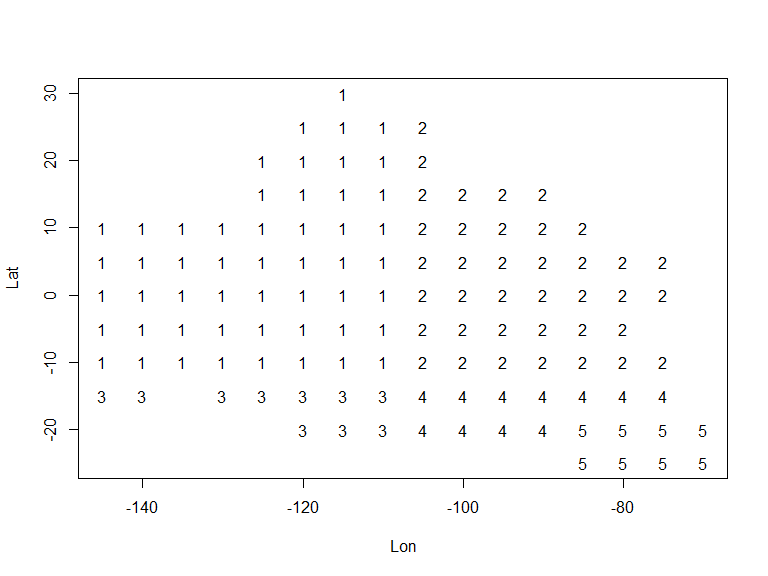
PS <- "NOA"  
  
cae.stratflg <- create.strat.flg.f(cae$latc5,cae$lonc5,is.lwrght=F,cae$month,cae$setype,cae$class,PS=PS,Species=Species)  
lfgrpd.stratflg <- create.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,is.lwrght=T,floor(lfgrpd$moda/100),lfgrpd$setype,lfgrpd$class,PS=PS,Species=Species)

Check the strata definition for NOA in both cae and lf data sets to make sure that they are correct

check.strat.flg.f(cae$latc5,cae$lonc5,cae.stratflg)



check.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,lfgrpd.stratflg)



Loop through every year between yr.start and yr.end to get catch and composition data for BET in the NOA fishery

for(year in yr.start:yr.end) {  
 # print(paste0("Year: ",year))  
   
 # print("Step 1: get well estimates")  
 well.estimates <- well.estimates.f(lfgrpd[lfgrpd$year.firstset==year,],lfmm)  
   
 # print("Step 2: get catch estimates")  
 catch.estimates <- get.catch.estimates.f(cae,cae.stratflg,total.unlds,lfgrpd,lfgrpd.stratflg,lfmm,year,2,well.estimates,area.substitution.mat,grow.increments,PS=PS,Species=Species)  
   
 # print("Step 3: get fishery estimates")  
 fishery.estimates <- fishery.estimates.f(catch.estimates$stratum.estimates.withsamps,catch.estimates$stratum.estimates.NOsamps,year,PS=PS,Species=Species)  
   
 assign(paste0("fishery.estimates.", year), fishery.estimates, pos=1)  
}  
# save middle-step data as a record  
save(list=objects(pat="fishery.estimates"),file=paste0(save\_dir,"BET\_",PS,"\_2000-2021.RData"))

Get final NOA catch and comp output for the stock assessment

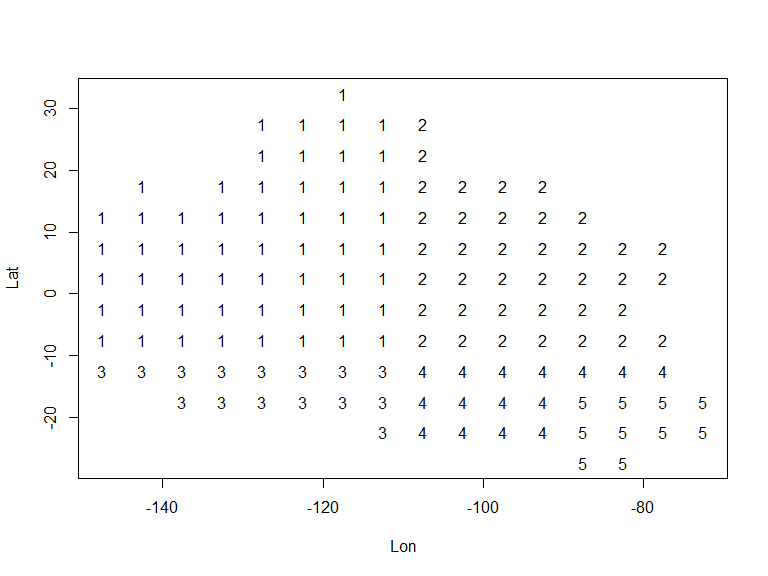
BET.NOA.Catch.20002021<-compile.catch.output.f(yr.start,yr.end,PS=PS,Species=Species,c("A1","A2","A3","A4","A5")) # five NOA fisheries  
BET.NOA.Comp.20002021<-compile.sizecomps.output.f(yr.start,yr.end,PS=PS,Species=Species)

* Step 5: compile the DEL catch and composition data for BET

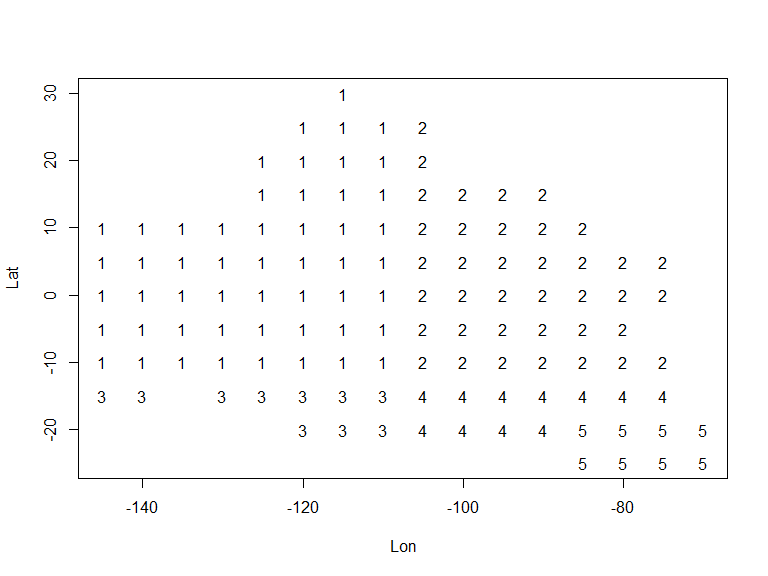
PS <- "DEL"  
  
cae.stratflg <- create.strat.flg.f(cae$latc5,cae$lonc5,is.lwrght=F,cae$month,cae$setype,cae$class,PS=PS,Species=Species)  
lfgrpd.stratflg <- create.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,is.lwrght=T,floor(lfgrpd$moda/100),lfgrpd$setype,lfgrpd$class,PS=PS,Species=Species)

Check the strata definition for DEL in both cae and lf data sets to make sure that they are correct

check.strat.flg.f(cae$latc5,cae$lonc5,cae.stratflg)



check.strat.flg.f(lfgrpd$lat.5deg,lfgrpd$lon.5deg,lfgrpd.stratflg)



Loop through every year between yr.start and yr.end to get catch and composition data for BET in the DEL fishery

for(year in yr.start:yr.end) {  
 # print(paste0("Year: ",year))  
   
 # print("Step 1: get well estimates")  
 well.estimates <- well.estimates.f(lfgrpd[lfgrpd$year.firstset==year,],lfmm)  
   
 # print("Step 2: get catch estimates")  
 catch.estimates <- get.catch.estimates.f(cae,cae.stratflg,total.unlds,lfgrpd,lfgrpd.stratflg,lfmm,year,2,well.estimates,area.substitution.mat,grow.increments,PS=PS,Species=Species)  
   
 # print("Step 3: get fishery estimates")  
 fishery.estimates <- fishery.estimates.f(catch.estimates$stratum.estimates.withsamps,catch.estimates$stratum.estimates.NOsamps,year,PS=PS,Species=Species)  
   
 assign(paste0("fishery.estimates.", year), fishery.estimates, pos=1)  
}  
# save middle-step data as a record  
save(list=objects(pat="fishery.estimates"),file=paste0(save\_dir,"BET\_",PS,"\_2000-2021.RData"))

Get final DEL catch and comp output for the stock assessment

BET.DEL.Catch.20002021<-compile.catch.output.f(yr.start,yr.end,PS=PS,Species=Species,c("A1","A2","A3","A4","A5")) # tive DEL fisheries  
BET.DEL.Comp.20002021<-compile.sizecomps.output.f(yr.start,yr.end,PS=PS,Species=Species)

* Step 6: save all results for BET as csv files

write.csv(BET.OBJ.Catch.20002021,file=paste0(save\_dir,"BET.OBJ.Catch.20002021.csv"),row.names = FALSE)  
write.csv(BET.OBJ.Comp.20002021,file=paste0(save\_dir,"BET.OBJ.Comp.20002021.csv"),row.names = FALSE)  
write.csv(BET.NOA.Catch.20002021,file=paste0(save\_dir,"BET.NOA.Catch.20002021.csv"),row.names = FALSE)  
write.csv(BET.NOA.Comp.20002021,file=paste0(save\_dir,"BET.NOA.Comp.20002021.csv"),row.names = FALSE)  
write.csv(BET.DEL.Catch.20002021,file=paste0(save\_dir,"BET.DEL.Catch.20002021.csv"),row.names = FALSE)  
write.csv(BET.DEL.Comp.20002021,file=paste0(save\_dir,"BET.DEL.Comp.20002021.csv"),row.names = FALSE)