

Gizzard shad biomass index development

Sean Hardison

2024-04-23

```
# Librararies and data

knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,
                      fig.align = "center")

library(tidyverse)
library(sf)
library(DHARMA)
library(tsibble)
library(lubridate)
library(INLA)
library(raster)
library(sdmTMB)
library(patchwork)

# load ChesMMAp data----
load(here::here("data/chesmmap.rdata"))
rm(count) # not using count data in analysis

# load harvest data----
load(here::here("data/processed_landings.rdata"))

# raster for projecting over
rast_res <- 2.5
r <- raster(extent(chesmmap_poly),
            res = rast_res,
            crs = crs(ncrs))

# df from raster
gridded_cb <- chesmmap_poly %>%
  fasterize::fasterize(.,r) %>%
  dream::rst_to_tib()

# ggplotly(
#   ggplot(gridded_cb)+
#     geom_raster(aes(y = latitude, x = longitude)) +
#     geom_sf(data = chesmmap_poly, fill = "transparent")
# )

# create prediction grid for biomass indices
create_pred_df <- F
if (create_pred_df){
  pred_df_glm <- expand_grid(month = c(3,5,7,9,11),
```

```

        year = 2002:2018,
        gridded_cb) %>%
mutate(month2 = ifelse(month == 3, "MAR",
                        ifelse(month == 5, "MAY",
                                ifelse(month == 7, "JUL",
                                        ifelse(month == 9, "SEP",
                                                ifelse(month == 11, "NOV",
                                                        NA))))),

        yseas = factor(paste(year, month2)),
        ymon = yearmonth(paste(year, month2)),
        ymon = as.numeric(ymon),
        fmonth = factor(month),
        year2 = factor(year),
        season = month2,
        metacomm = ifelse(latitude > 4200.327, "MD", "VA"))
save(pred_df_glmm, file = here::here("data/prediction_grid.rdata"))
} else {
  load(file = here::here("data/prediction_grid.rdata"))
}

# nsims for simulation step
nsims <- 2000

```

Gizzard shad

Data spatial footprint

```

selected_species <- "gizzard shad"
ss <- "gz"

# data processing----
bio2 <- bio %>%
  filter(common %in% selected_species) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = 4326) %>%
  st_transform(crs = st_crs(ncrs)) %>%
  mutate(stat_area = st_nearest_feature(., va_mgmt)) %>%
  mutate(biomass = biomass/(areasw/1e6),
         presence = ifelse(biomass == 0, 0, 1),
         location = plyr::mapvalues(stat_area,
                                     from = 1:7,
                                     to = c("025", "027", "029",
                                              "CBUW", "CBUE",
                                              "CBLW", "CBLE"))) %>%
  dream::sfc_as_cols(names = c("longitude", "latitude")) %>%
  st_set_geometry(NULL) %>%
  mutate(day = day(date),
         month = month(date),
         fmonth = factor(month),
         season = factor(season, levels = c("MAR", "MAY", "JUL",
                                             "SEP", "NOV")),
         season_char = as.character(season),
         cruise = factor(cruise),

```

```

    year2 = factor(year),
    ymon = as.numeric(yearmonth(date)),
    yseas = factor(paste(year, season)) %>%
mutate(metacomm = ifelse(stat_area %in% 1:3, "MD", "VA"))

# model fitting----
process <- T

p <-
  chesmmmap_poly %>%
  as("sf")
ex <- extent(p)
ex[4] <- max(bio2 %>% filter(biomass > 0) %>% pull(latitude))
ex[3] <- min(bio2 %>% filter(biomass > 0) %>% pull(latitude))

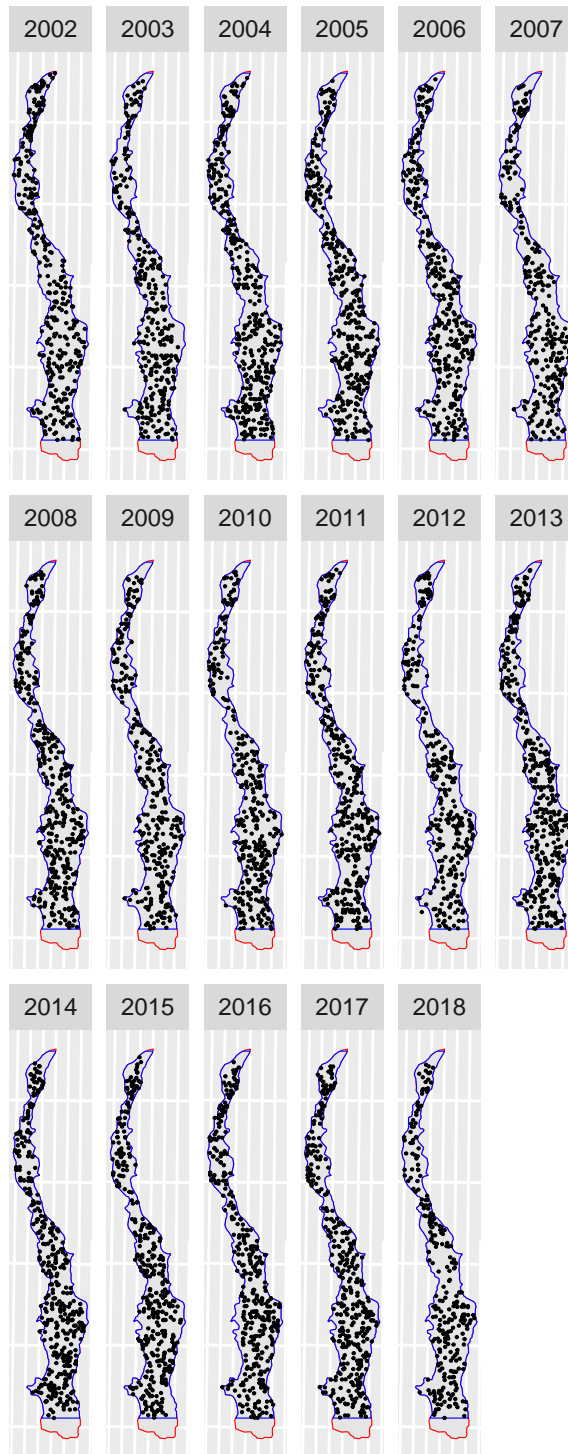
pred_df_glmm_filt <- pred_df_glmm %>%
  filter(latitude <= ex[4],
         latitude >= ex[3])

chesmmmap_poly2 <- st_crop(p, ex) %>% as_Spatial()
bio3 <- bio2 %>% filter(latitude <= ex[4],
                      latitude >= ex[3])

ggplot() +
  geom_sf(data = chesmmmap_poly, color = "red") +
  geom_sf(data = chesmmmap_poly2 %>% as("sf"),
         color = 'blue') +
  geom_point(data = bio3,
            aes(x = longitude, y = latitude),
            size = 0.1) +
  facet_wrap(~year, nrow = 3) +
  theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank()) +
  # geom_sf(data = t, color = "blue") +
  labs(title = selected_species)

```

gizzard shad

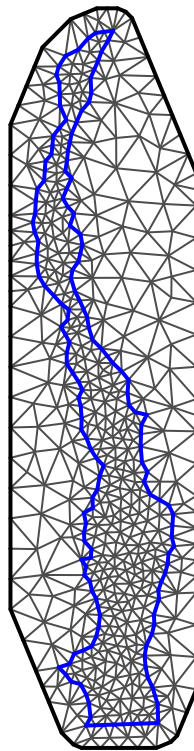


Mesh

```
max.edge <- 3
bound.outer <- 8
mesh <- inla.mesh.2d(boundary = chesmmap_poly2,
                     max.edge = c(1,5)*max.edge,
                     offset = c(max.edge, bound.outer),
                     cutoff = max.edge)

# mesh for sdmTMB----
cb_spde <- make_mesh(data = bio3,
                     xy_cols = c("longitude", "latitude"),
                     mesh = mesh)

plot(mesh)
```



Fit model

```
hurld_mod <- sdmTMB(biomass ~ 0 +
                    season +
                    year2 +
                    s(month, year2, bs = 'fs', k = 4),
                    family = sdmTMB::tweedie(),
                    time = "year",
                    spatial = "on",
                    spatiotemporal = "IID",
                    reml = F,
                    control = sdmTMBcontrol(nlminb_loops = 2),
                    mesh = cb_spde,
                    data = bio3)

sanity(hurld_mod)
```

Residuals

```
set.seed(123)
if (process){
  samps <- sdmTMBextra::predict_mle_mcmc(hurld_mod, mcmc_iter = 201, mcmc_warmup = 200)
  mcmc_res <- residuals(hurld_mod, type = "mle-mcmc", mcmc_samples = samps)
  save(samps, mcmc_res,
       file = here::here(paste0("data/model_output/mcmc_res_tw_",ss, ".rdata")))
} else {
  load(here::here(paste0("data/model_output/mcmc_res_tw_",ss, ".rdata")))
  stats::qqnorm(mcmc_res, main = paste(selected_species, "MCMC residuals, fam == tweedie"))
  stats::qqline(mcmc_res)
}
```

```
##
## SAMPLING FOR MODEL 'tmb_generic' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.058283 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 582.83 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 201 [  0%] (Warmup)
## Chain 1: Iteration:  20 / 201 [  9%] (Warmup)
## Chain 1: Iteration:  40 / 201 [ 19%] (Warmup)
## Chain 1: Iteration:  60 / 201 [ 29%] (Warmup)
## Chain 1: Iteration:  80 / 201 [ 39%] (Warmup)
## Chain 1: Iteration: 100 / 201 [ 49%] (Warmup)
## Chain 1: Iteration: 120 / 201 [ 59%] (Warmup)
## Chain 1: Iteration: 140 / 201 [ 69%] (Warmup)
## Chain 1: Iteration: 160 / 201 [ 79%] (Warmup)
## Chain 1: Iteration: 180 / 201 [ 89%] (Warmup)
## Chain 1: Iteration: 200 / 201 [ 99%] (Warmup)
## Chain 1: Iteration: 201 / 201 [100%] (Sampling)
## Chain 1:
```

```
## Chain 1: Elapsed Time: 9092.3 seconds (Warm-up)
## Chain 1: 46.189 seconds (Sampling)
## Chain 1: 9138.49 seconds (Total)
## Chain 1:
```

Generate predictions

```
# VA-----
index_va <- lapply(c(3,5,7,9,11), \(m) {
  nd <- pred_df_glmm_filt %>%
    filter(month == m,
           metacomm == "VA",
           year %in% unique(bio3$year))
  p <- predict(hurd_mod, newdata = nd,
              return_tmb_object = TRUE)
  ii <- get_index(p, bias_correct = TRUE, area = 2.5^2)
  ii$month <- m
  ii
})

pred_out_seas_va <- do.call(rbind, index_va) %>%
  mutate(ymon = as.numeric(yearmonth(paste(year, month, sep = "-"))),
         common = selected_species,
         metacomm = "VA") %>%
  arrange(ymon)

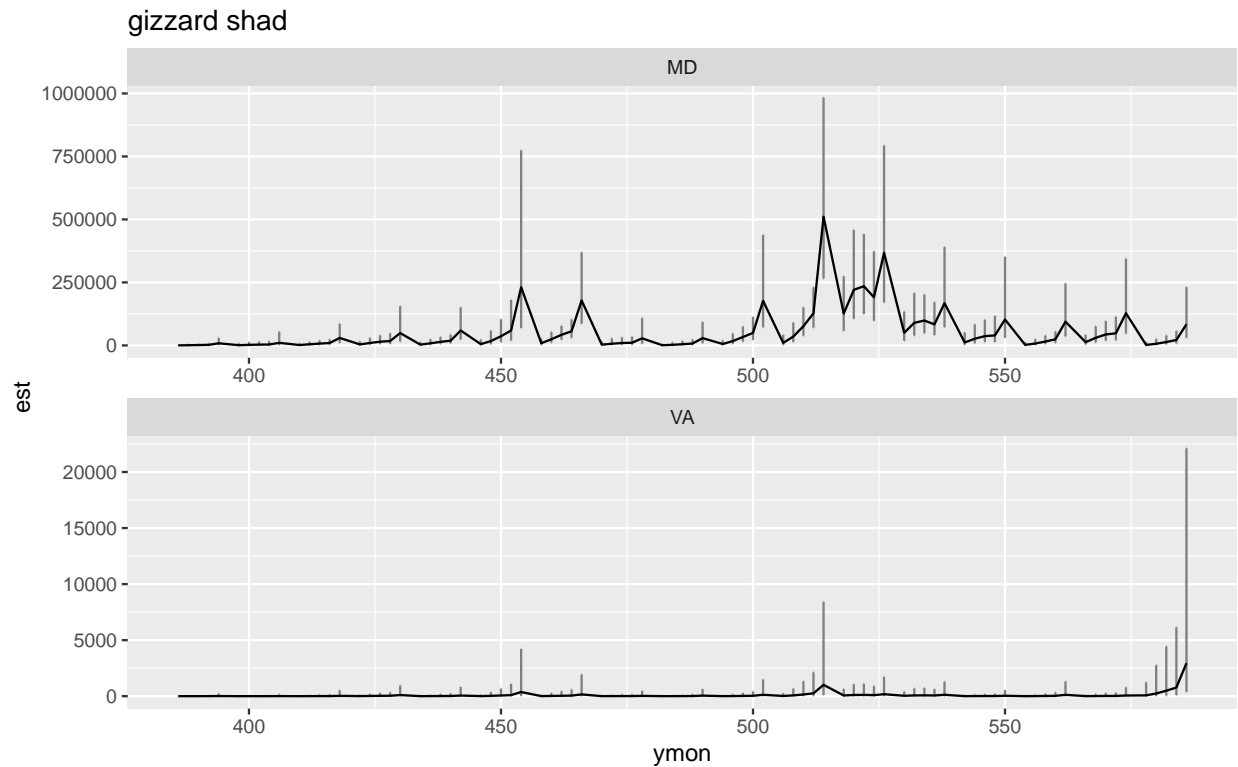
# MD-----
index_md <- lapply(c(3,5,7,9,11), \(m) {
  nd <- pred_df_glmm_filt %>%
    filter(month == m,
           metacomm == "MD",
           year %in% unique(bio3$year))
  p <- predict(hurd_mod, newdata = nd,
              return_tmb_object = TRUE)
  ii <- get_index(p, bias_correct = T, area = 2.5^2)
  ii$month <- m
  ii
})

pred_out_seas_md <- do.call(rbind, index_md) %>%
  mutate(ymon = as.numeric(yearmonth(paste(year, month, sep = "-"))),
         common = selected_species,
         metacomm = "MD") %>%
  arrange(ymon)

# combine-----
pred_out_seas <- bind_rows(pred_out_seas_va,
                           pred_out_seas_md)
```

Visualize predictions

```
ggplot(pred_out_seas) +  
  geom_errorbar(aes(ymax = upr, ymin = lwr, x = ymon),  
    width= 0.1, color = "gray50") +  
  geom_line(aes(y = est, x = ymon)) +  
  facet_wrap(~metacomm, scales = "free", nrow = 2) +  
  labs(title = selected_species)
```



```
## save outputs  
  
pred_out_seas_gz <- pred_out_seas %>%  
  as_tibble()  
save(pred_out_seas_gz,  
  file = here::here(paste0("data/model_output/pred_out_seas_",ss,"_sim.rdata"))
```