Fitting Integrated Population Models to Lower Columbia River Chum Salmon Monitoring Data

Eric Buhle, Kale Bentley, Thomas Buehrens, Mark Scheuerell, Todd Hillson, and ...

2020-07-07

Contents

1 Overview **1**

2 Setup and data **1**

3 Retrospective models **5**

4 Forecasting **13**

1 Overview

Background on IPMs, outline of **salmonIPM**...

[eqns]

2 Setup and data

Load the packages we'll need...

```
options(device = ifelse(.Platform$OS.type == "windows", "windows", "quartz"))
options(mc.cores = parallel::detectCores(logical = FALSE) - 1)

library(salmonIPM)
library(rstan)
library(shinystan)
library(matrixStats)
library(tibble)
library(dplyr)
library(tidyr)
library(reshape2)
```

```
library(yarrr)
library(corrplot)
library(magicaxis)
library(zoo)
library(here)

if(file.exists(here("analysis","results","LCRchumIPM.RData")))
  load(here("analysis","results","LCRchumIPM.RData"))
```

Read in and manipulate the data...

```
# Mapping of location to population
location_pop <- read.csv(here("data","Location.Reach_Population.csv"),
                         header = TRUE, stringsAsFactors = TRUE) %>%
  rename(strata = Strata, location = Location.Reach, pop1 = Population1, pop2 = Population2)

# Mapping of disposition to hatchery vs. wild (i.e., broodstock vs. natural spawner)
disposition_HW <- read.csv(here("data","Disposition_HW.csv"),
                           header = TRUE, stringsAsFactors = TRUE) %>%
  rename(disposition = Disposition) %>% arrange(HW)

# Start dates of hatcheries associated with populations
hatcheries <- read.csv(here("data","Hatchery_Programs.csv"), header = TRUE, stringsAsFactors =

# Spawner abundance data
# Assumptions:
# (1) NAs in hatchery dispositions (incl. Duncan Channel) are really zeros
# (2) NAs in Duncan Creek from 2004-present are really zeros
# (3) All other NAs are real missing observations
# (4) When calculating the observation error of log(S_obs), tau_S_obs, assume
#     Abund.Mean and Abund.SD are the mean and SD of a lognormal posterior distribution
#     of spawner abundance based on the sample
spawner_data <- read.csv(here("data","Data_ChumSpawnerAbundance_2019-12-12.csv"),
                         header = TRUE, stringsAsFactors = TRUE) %>%
  rename(year = Return.Yr., strata = Strata, location = Location.Reach,
         disposition = Disposition, method = Method, S_obs = Abund.Mean, SD = Abund.SD) %>%
  mutate(pop = location_pop$pop2[match(location, location_pop$location)],
         disposition_HW = disposition_HW$HW[match(disposition, disposition_HW$disposition)],
         S_obs = replace(S_obs, is.na(S_obs) & disposition_HW == "H", 0),
         S_obs = replace(S_obs, is.na(S_obs) & pop == "Duncan_Creek" & year >= 2004, 0),
         tau_S_obs = sqrt(log((SD/S_obs)^2 + 1))) %>%
  select(year:location, pop, disposition, disposition_HW, method:tau_S_obs) %>%
  arrange(strata, location, year)

names_S_obs <- disposition_HW$disposition
names_B_take_obs <- disposition_HW$disposition[disposition_HW$HW == "H"]
```

```r
spawner_data_agg <- spawner_data %>% group_by(strata, pop, year, disposition) %>%
  summarize(S_obs = sum(S_obs), tau_S_obs = unique(tau_S_obs)) %>% ungroup() %>%
  pivot_wider(id_cols = c(strata, pop, year), names_from = disposition,
              values_from = c(S_obs, tau_S_obs), values_fill = list(S_obs = 0)) %>%
  add_column(S_obs = rowSums(select(., all_of(paste0("S_obs_", names_S_obs)))),
             tau_S_obs = rowSums(select(., all_of(paste0("tau_S_obs_", names_S_obs))), na.rm =
             B_take_obs = rowSums(select(., all_of(paste0("S_obs_", names_B_take_obs))))) %>%
  mutate(tau_S_obs = replace(tau_S_obs, tau_S_obs == 0, NA)) %>%
  select(-matches(paste(names_S_obs, collapse = "|"))) %>%
  as.data.frame()

# Spawner age-, sex-, and origin-frequency (aka BioData)
bio_data <- read.csv(here("data","Data_ChumSpawnerBioData_2019-12-12.csv"),
                     header = TRUE, stringsAsFactors = TRUE) %>%
  rename(year = Return.Yr., strata = Strata, location = Location.Reach,
         disposition = Disposition, origin = Origin, sex = Sex, age = Age, count = Count) %>%
  mutate(pop = location_pop$pop2[match(location, location_pop$location)],
         origin_HW = ifelse(origin == "Natural_spawner", "W", "H"),
         count = ifelse(is.na(count), 0, count)) %>%
  select(year:location, pop, disposition, origin, origin_HW, sex:count) %>%
  arrange(strata, location, year, origin, age, sex)

# age of wild spawners only
bio_data_age <- bio_data %>% filter(origin_HW == "W") %>%
  dcast(year + strata + pop ~ age, value.var = "count", fun.aggregate = sum)

bio_data_origin <- bio_data %>%
  dcast(year + strata + pop ~ origin_HW, value.var = "count", fun.aggregate = sum)

# Juvenile abundance data
# Assumptions:
# (1) Smolts from Duncan Channel represent all naturally produced offspring of spawners
#     in Duncan Creek (hence Duncan_Channel -> Duncan_Creek in location_pop)
# (2) Duncan_North + Duncan_South = Duncan_Channel, so the former two are redundant
#     (not really an assumption, although the equality isn't perfect in all years)
# (3) When calculating the observation error of log(M_obs), tau_M_obs, assume
#     Abund_Mean and Abund_SD are the mean and SD of a lognormal posterior distribution
#     of smolt abundance based on the sample
# (4) If Abund_SD == 0 (when Analysis=="Census": some years in Duncan_Creek and
#     Hamilton_Channel) treat as NA
juv_data <- read.csv(here("data", "Data_ChumJuvenileAbundance_2020-06-09.csv"),
                     header = TRUE, stringsAsFactors = TRUE) %>%
  rename(brood_year = Brood.Year, year = Outmigration.Year, strata = Strata,
         location = Location.Reach, origin = Origin, trap_type = TrapType,
         analysis = Analysis, partial_spawners = Partial.Spawners, raw_catch = RawCatch,
         M_obs = Abund_Median, mean = Abund_Mean, SD = Abund_SD,
         L95 = Abund_L95, U95 = Abund_U95, CV = Abund_CV, comments = Comments) %>%
  mutate(pop = location_pop$pop2[match(location, location_pop$location)],
```

```
        tau_M_obs = replace(sqrt(log((SD/mean)^2 + 1)), SD==0, NA)) %>%
  select(strata, location, pop, year, brood_year, origin:CV, tau_M_obs, comments) %>%
  arrange(strata, location, year)

# drop hatchery or redundant pops and cases with leading or trailing NAs in M_obs
head_noNA <- function(x) { cumsum(!is.na(x)) > 0 }
juv_data_incl <- juv_data %>% filter(pop %in% spawner_data$pop) %>%
  mutate(location = factor(location), pop = factor(pop, levels = levels(spawner_data$pop))) %>%
  group_by(pop) %>% filter(head_noNA(M_obs) & rev(head_noNA(rev(M_obs)))) %>% as.data.frame()

# Fish data formatted for salmonIPM
# Drop age-2 and age-6 samples (each is < 0.1% of aged spawners)
# Use A = 1 for now (so Rmax in units of spawners)
fish_data <- full_join(spawner_data_agg, bio_data_age, by = c("strata","pop","year")) %>%
  full_join(bio_data_origin, by = c("strata","pop","year")) %>%
  full_join(juv_data_incl, by = c("strata","pop","year")) %>%
  mutate(B_take_obs = replace(B_take_obs, is.na(B_take_obs), 0)) %>%
  rename_at(vars(contains("Age-")), list(~ paste0(sub("Age-","n_age",.), "_obs"))) %>%
  select(-c(n_age2_obs, n_age6_obs)) %>%
  rename(n_H_obs = H, n_W_obs = W) %>% mutate(A = 1, fit_p_HOS = NA, F_rate = 0) %>%
  mutate_at(vars(contains("n_")), ~ replace(., is.na(.), 0)) %>%
  select(strata, pop, year, A, S_obs, tau_S_obs, M_obs, tau_M_obs, n_age3_obs:n_W_obs,
         fit_p_HOS, B_take_obs, F_rate) %>% arrange(strata, pop, year)

# fill in fit_p_HOS
for(i in 1:nrow(fish_data)) {
  pop_i <- as.character(fish_data$pop[i])
  start_year <- ifelse(pop_i %in% hatcheries$pop,
                       min(hatcheries$start_brood_year[hatcheries$pop == pop_i]) + 1,
                       NA)
  fish_data$fit_p_HOS[i] <- ifelse((!is.na(start_year) & fish_data$year[i] >= start_year) |
                                     fish_data$n_H_obs[i] > 0, 1, 0)
}

# # drop cases with initial NAs in S_obs unless bio data is present
# fish_data <- fish_data %>% mutate(n_age = rowSums(select(., n_age2_obs:n_age6_obs))) %>%
#   group_by(pop) %>%  filter(head_noNA(S_obs) | cumsum(n_age) > 0) %>%
#   select(-n_age) %>% as.data.frame()

# subsets for models with specific stage structure
# spawner-spawner: drop cases with initial NAs in S_obs, even if bio data is present
fish_data_SS <- fish_data %>% group_by(pop) %>% filter(head_noNA(S_obs)) %>% as.data.frame()
# spawner-spawner: drop cases with initial NAs in M_obs, even if bio data is present
fish_data_SMS <- fish_data %>% group_by(pop) %>%
  filter(head_noNA(S_obs) | head_noNA(M_obs)) %>% as.data.frame()

# pad data with future years to generate forecasts
# use 5-year (1-generation) time horizon
```

```
fish_data_SMS_fore <- fish_data_SMS %>% group_by(pop) %>%
  slice(rep(n(), max(fish_data_SMS$year) + 5 - max(year))) %>%
  mutate(year = (unique(year) + 1):(max(fish_data_SMS$year) + 5),
         S_obs = NA, tau_S_obs = NA, M_obs = NA, tau_M_obs = NA,
         fit_p_HOS = 0, B_take_obs = 0, F_rate = 0) %>%
  mutate_at(vars(starts_with("n_")), ~ 0) %>%
  full_join(fish_data_SMS) %>% arrange(pop, year) %>%
  mutate(forecast = year > max(fish_data_SMS$year)) %>%
  select(strata:year, forecast, A:F_rate) %>% as.data.frame()
```

Let's look at the first few rows of `fish_data` to see the format **salmonIPM** expects...

```
head(fish_data_SMS)
```

```
   strata        pop year A S_obs  tau_S_obs M_obs tau_M_obs n_age3_obs n_age4_obs n_age5_obs n
1 Cascade Cascade_MS 2002 1  3160 0.15231109    NA        NA         101         114           7
2 Cascade Cascade_MS 2003 1  2866 0.05572085    NA        NA          19         448          26
3 Cascade Cascade_MS 2004 1  2324 0.08680695    NA        NA          75         203          50
4 Cascade Cascade_MS 2005 1   923 0.03711614    NA        NA           4          38           0
5 Cascade Cascade_MS 2006 1   869 0.02342601    NA        NA           1          41           0
6 Cascade Cascade_MS 2007 1   576 0.05358997    NA        NA          32         115          43
  B_take_obs F_rate
1         15      0
2          0      0
3          0      0
4          0      0
5          0      0
6          0      0
```

3 Retrospective models

Fit two-stage spawner-smolt-spawner models and explore output...

Density-independent

```
LCRchum_exp <- salmonIPM(fish_data = fish_data_SMS, ages = list(M = 1),
                         stan_model = "IPM_LCRchum_pp", SR_fun = "exp",
                         pars = c("B_rate_all","mu_Rmax","sigma_Rmax","rho_alphaRmax","Rmax"),
                         include = FALSE, log_lik = TRUE,
                         chains = 3, iter = 1500, warmup = 500,
                         control = list(adapt_delta = 0.99, max_treedepth = 13))
```

```
print(LCRchum_exp, prob = c(0.025,0.5,0.975),
      pars = c("alpha","rho_alphaRmax","phi_M","phi_MS","gamma","p","tau_M","tau_S",
               "p_HOS","S","M","s_MS","q","LL"),
      include = FALSE, use_cache = FALSE)
```

```
Inference for Stan model: IPM_LCRchum_pp.
3 chains, each with iter=1500; warmup=500; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=3000.

                   mean se_mean    sd      2.5%       50%     97.5% n_eff Rhat
mu_alpha           6.56    0.00  0.13      6.29      6.56      6.79  1453 1.00
sigma_alpha        0.29    0.00  0.10      0.13      0.28      0.53  1637 1.00
rho_phi_M          0.43    0.01  0.29     -0.25      0.49      0.83  1030 1.00
sigma_phi_M        0.42    0.00  0.13      0.20      0.41      0.74   984 1.00
sigma_M            0.42    0.00  0.05      0.32      0.41      0.53   886 1.00
mu_MS              0.00    0.00  0.00      0.00      0.00      0.00  1693 1.00
rho_phi_MS         0.58    0.01  0.20      0.09      0.61      0.85  1298 1.00
sigma_phi_MS       0.94    0.01  0.22      0.59      0.91      1.47  1414 1.00
sigma_MS           0.56    0.00  0.06      0.46      0.56      0.68  1075 1.00
mu_p[1]            0.20    0.00  0.02      0.17      0.20      0.24   988 1.00
mu_p[2]            0.75    0.00  0.02      0.71      0.75      0.78  1284 1.00
mu_p[3]            0.05    0.00  0.01      0.04      0.05      0.06   679 1.01
sigma_gamma[1]     0.18    0.01  0.14      0.01      0.15      0.50   534 1.00
sigma_gamma[2]     0.12    0.00  0.09      0.00      0.10      0.35   667 1.00
R_gamma[1,1]       1.00     NaN  0.00      1.00      1.00      1.00   NaN  NaN
R_gamma[1,2]       0.24    0.02  0.57     -0.90      0.35      0.98  1359 1.00
R_gamma[2,1]       0.24    0.02  0.57     -0.90      0.35      0.98  1359 1.00
R_gamma[2,2]       1.00    0.00  0.00      1.00      1.00      1.00  1084 1.00
sigma_p[1]         1.41    0.00  0.12      1.18      1.40      1.68   667 1.00
sigma_p[2]         0.78    0.00  0.08      0.63      0.78      0.96   778 1.00
R_p[1,1]           1.00     NaN  0.00      1.00      1.00      1.00   NaN  NaN
R_p[1,2]           0.70    0.00  0.07      0.55      0.71      0.82   763 1.00
R_p[2,1]           0.70    0.00  0.07      0.55      0.71      0.82   763 1.00
R_p[2,2]           1.00    0.00  0.00      1.00      1.00      1.00   135 1.00
mu_tau_M           0.07    0.00  0.01      0.05      0.07      0.09  2947 1.00
sigma_tau_M        1.08    0.00  0.13      0.86      1.06      1.38  2493 1.00
mu_tau_S           0.12    0.00  0.01      0.10      0.12      0.14  3837 1.00
sigma_tau_S        1.05    0.00  0.06      0.93      1.05      1.18  2798 1.00
lp__          -22340.10    1.35 35.07 -22410.24 -22339.82 -22269.64   674 1.00

Samples were drawn using NUTS(diag_e) at Mon Jul 06 23:42:39 2020.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Beverton-Holt

```
LCRchum_BH <- salmonIPM(fish_data = fish_data_SMS, ages = list(M = 1),
                        stan_model = "IPM_LCRchum_pp", SR_fun = "BH",
                        pars = "B_rate_all", include = FALSE, log_lik = TRUE,
                        chains = 3, iter = 1500, warmup = 500,
                        control = list(adapt_delta = 0.99, max_treedepth = 13))
```

```
print(LCRchum_BH, prob = c(0.025,0.5,0.975),
      pars = c("alpha","Rmax","phi_M","phi_MS","gamma","p","tau_M","tau_S",
               "p_HOS","S","M","s_MS","q","LL"),
      include = FALSE, use_cache = FALSE)
```

```
Inference for Stan model: IPM_LCRchum_pp.
3 chains, each with iter=1500; warmup=500; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=3000.
```

	mean	se_mean	sd	2.5%	50%	97.5%	n_eff	Rhat
mu_alpha	8.91	0.03	0.84	7.55	8.78	10.81	670	1.00
sigma_alpha	1.78	0.01	0.54	0.86	1.74	2.88	1889	1.00
mu_Rmax	13.11	0.02	0.52	12.09	13.09	14.16	1172	1.00
sigma_Rmax	1.60	0.01	0.38	1.00	1.56	2.53	1890	1.00
rho_alphaRmax	-0.41	0.01	0.30	-0.86	-0.46	0.26	824	1.00
rho_phi_M	0.05	0.01	0.43	-0.73	0.06	0.77	1434	1.00
sigma_phi_M	0.14	0.00	0.09	0.01	0.13	0.34	477	1.00
sigma_M	0.31	0.00	0.04	0.24	0.31	0.40	804	1.00
mu_MS	0.00	0.00	0.00	0.00	0.00	0.00	1727	1.00
rho_phi_MS	0.50	0.01	0.24	-0.07	0.54	0.83	1174	1.00
sigma_phi_MS	1.03	0.01	0.24	0.68	0.99	1.61	1292	1.00
sigma_MS	0.54	0.00	0.05	0.45	0.54	0.64	1153	1.00
mu_p[1]	0.20	0.00	0.02	0.16	0.20	0.24	1029	1.00
mu_p[2]	0.75	0.00	0.02	0.71	0.75	0.78	1228	1.00
mu_p[3]	0.05	0.00	0.01	0.04	0.05	0.06	930	1.00
sigma_gamma[1]	0.19	0.01	0.15	0.01	0.16	0.55	750	1.00
sigma_gamma[2]	0.15	0.01	0.13	0.00	0.13	0.45	510	1.01
R_gamma[1,1]	1.00	NaN	0.00	1.00	1.00	1.00	NaN	NaN
R_gamma[1,2]	0.26	0.02	0.57	-0.92	0.38	0.98	766	1.01
R_gamma[2,1]	0.26	0.02	0.57	-0.92	0.38	0.98	766	1.01
R_gamma[2,2]	1.00	0.00	0.00	1.00	1.00	1.00	1032	1.00
sigma_p[1]	1.49	0.00	0.13	1.24	1.48	1.75	752	1.00
sigma_p[2]	0.83	0.00	0.09	0.67	0.82	1.02	727	1.01
R_p[1,1]	1.00	NaN	0.00	1.00	1.00	1.00	NaN	NaN
R_p[1,2]	0.69	0.00	0.07	0.53	0.69	0.81	845	1.00
R_p[2,1]	0.69	0.00	0.07	0.53	0.69	0.81	845	1.00
R_p[2,2]	1.00	0.00	0.00	1.00	1.00	1.00	126	1.00
mu_tau_M	0.07	0.00	0.01	0.05	0.07	0.09	3513	1.00
sigma_tau_M	1.05	0.00	0.12	0.86	1.04	1.31	4325	1.00
mu_tau_S	0.11	0.00	0.01	0.10	0.11	0.13	3554	1.00
sigma_tau_S	0.97	0.00	0.06	0.87	0.97	1.09	1816	1.00
lp__	-22306.81	1.38	34.71	-22376.40	-22306.29	-22242.24	629	1.00

```
Samples were drawn using NUTS(diag_e) at Tue Jul 07 02:10:43 2020.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Ricker

```
LCRchum_Ricker <- salmonIPM(fish_data = fish_data_SMS, ages = list(M = 1),
                            stan_model = "IPM_LCRchum_pp", SR_fun = "Ricker",
                            pars = "B_rate_all", include = FALSE, log_lik = TRUE,
                            chains = 3, iter = 1500, warmup = 500,
                            control = list(adapt_delta = 0.99, max_treedepth = 13))
```

```
print(LCRchum_Ricker, prob = c(0.025,0.5,0.975),
      pars = c("alpha","Rmax","phi_M","phi_MS","gamma","p","tau_M","tau_S",
               "p_HOS","S","M","s_MS","q","LL"),
      include = FALSE, use_cache = FALSE)
```

```
Inference for Stan model: IPM_LCRchum_pp.
3 chains, each with iter=1500; warmup=500; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=3000.
```

	mean	se_mean	sd	2.5%	50%	97.5%	n_eff	Rhat
mu_alpha	7.06	0.00	0.16	6.75	7.06	7.36	1158	1.00
sigma_alpha	0.33	0.01	0.15	0.05	0.32	0.66	638	1.00
mu_Rmax	13.68	0.02	0.57	12.64	13.65	14.92	1120	1.00
sigma_Rmax	1.46	0.01	0.44	0.75	1.40	2.52	1022	1.00
rho_alphaRmax	-0.17	0.02	0.43	-0.84	-0.21	0.82	516	1.01
rho_phi_M	0.03	0.01	0.39	-0.71	0.03	0.74	1431	1.00
sigma_phi_M	0.20	0.00	0.10	0.02	0.19	0.43	667	1.00
sigma_M	0.32	0.00	0.04	0.24	0.32	0.41	1004	1.00
mu_MS	0.00	0.00	0.00	0.00	0.00	0.00	1660	1.00
rho_phi_MS	0.53	0.01	0.23	-0.04	0.57	0.83	1151	1.00
sigma_phi_MS	0.98	0.01	0.22	0.64	0.95	1.51	1581	1.00
sigma_MS	0.53	0.00	0.05	0.44	0.53	0.63	1258	1.00
mu_p[1]	0.20	0.00	0.02	0.17	0.20	0.23	1077	1.00
mu_p[2]	0.75	0.00	0.02	0.71	0.75	0.78	1418	1.00
mu_p[3]	0.05	0.00	0.01	0.04	0.05	0.06	970	1.00
sigma_gamma[1]	0.16	0.01	0.13	0.01	0.14	0.48	614	1.00
sigma_gamma[2]	0.12	0.00	0.10	0.00	0.10	0.37	880	1.00
R_gamma[1,1]	1.00	NaN	0.00	1.00	1.00	1.00	NaN	NaN
R_gamma[1,2]	0.22	0.01	0.57	-0.90	0.35	0.97	1544	1.00
R_gamma[2,1]	0.22	0.01	0.57	-0.90	0.35	0.97	1544	1.00
R_gamma[2,2]	1.00	0.00	0.00	1.00	1.00	1.00	995	1.00
sigma_p[1]	1.43	0.00	0.12	1.21	1.43	1.69	846	1.00
sigma_p[2]	0.79	0.00	0.08	0.64	0.79	0.97	969	1.00
R_p[1,1]	1.00	NaN	0.00	1.00	1.00	1.00	NaN	NaN
R_p[1,2]	0.70	0.00	0.07	0.55	0.71	0.83	1074	1.00
R_p[2,1]	0.70	0.00	0.07	0.55	0.71	0.83	1074	1.00
R_p[2,2]	1.00	0.00	0.00	1.00	1.00	1.00	102	1.00
mu_tau_M	0.07	0.00	0.01	0.05	0.07	0.10	2331	1.00
sigma_tau_M	1.09	0.00	0.13	0.87	1.08	1.38	1413	1.00

```
mu_tau_S            0.12    0.00  0.01      0.10      0.12      0.14  2290 1.00
sigma_tau_S         1.05    0.00  0.07      0.93      1.04      1.19  1547 1.00
lp__           -22338.86    1.24 33.45 -22405.94 -22337.71 -22274.49   728 1.00

Samples were drawn using NUTS(diag_e) at Tue Jul 07 04:54:30 2020.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

Model comparison based on LOO. Unhelpful because Pareto ks are too high, but appears to favor Beverton-Holt.

```
LL_LCRchum <- lapply(list(exp = LCRchum_exp, BH = LCRchum_BH, Ricker = LCRchum_Ricker),
                     loo::extract_log_lik, parameter_name = "LL", merge_chains = FALSE)

# Relative ESS of posterior draws of observationwise likelihood
r_eff_LCRchum <- lapply(LL_LCRchum, function(x) relative_eff(exp(x)))

# PSIS-LOO
LOO_LCRchum <- lapply(1:length(LL_LCRchum),
                      function(i) loo(LL_LCRchum[[i]], r_eff = r_eff_LCRchum[[i]]))
names(LOO_LCRchum) <- names(LL_LCRchum)

## Compare all three models
loo_compare(LOO_LCRchum)
```

```
       elpd_diff se_diff
BH       0.0       0.0
Ricker -14.7       9.4
exp    -14.9      10.6
```

```
## Exponential vs. Ricker
loo_compare(LOO_LCRchum[c("exp","Ricker")])
```

```
       elpd_diff se_diff
Ricker  0.0       0.0
exp    -0.2       7.1
```

```
## Exponential vs. Beverton-Holt
loo_compare(LOO_LCRchum[c("exp","BH")])
```

```
    elpd_diff se_diff
BH    0.0       0.0
exp -14.9      10.6
```

```
## Beverton-Holt vs. Ricker
loo_compare(LOO_LCRchum[c("BH","Ricker")])
```

```
       elpd_diff se_diff
BH       0.0       0.0
Ricker -14.7       9.4
```

Plot estimated spawner-smolt production curves and parameters for the Beverton-Holt model.

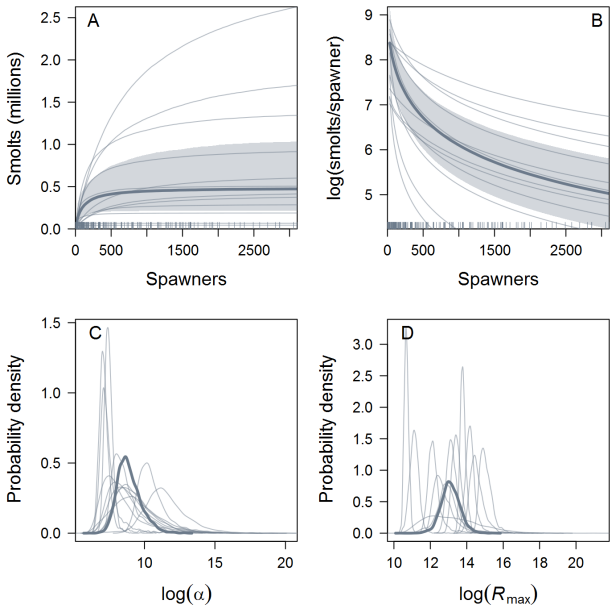

Figure 1: Estimated Beverton-Holt spawner-recruit relationship (A, B) and intrinsic productivity (C) and capacity (D) parameters for the multi-population IPM. Thin lines correspond to each of 12 populations of Lower Columbia chum salmon; thick lines represent hyper-means across populations. In (A, B), each curve is a posterior median and the shaded region represents the 90% credible interval of the hyper-mean curve (uncertainty around the population-specific curves is omitted for clarity).

The Beverton-Holt model is biologically plausible and appears to be supported by LOO, albeit with caveats, so let's tentatively proceed with that model for now. Here are the fits to the spawner data:

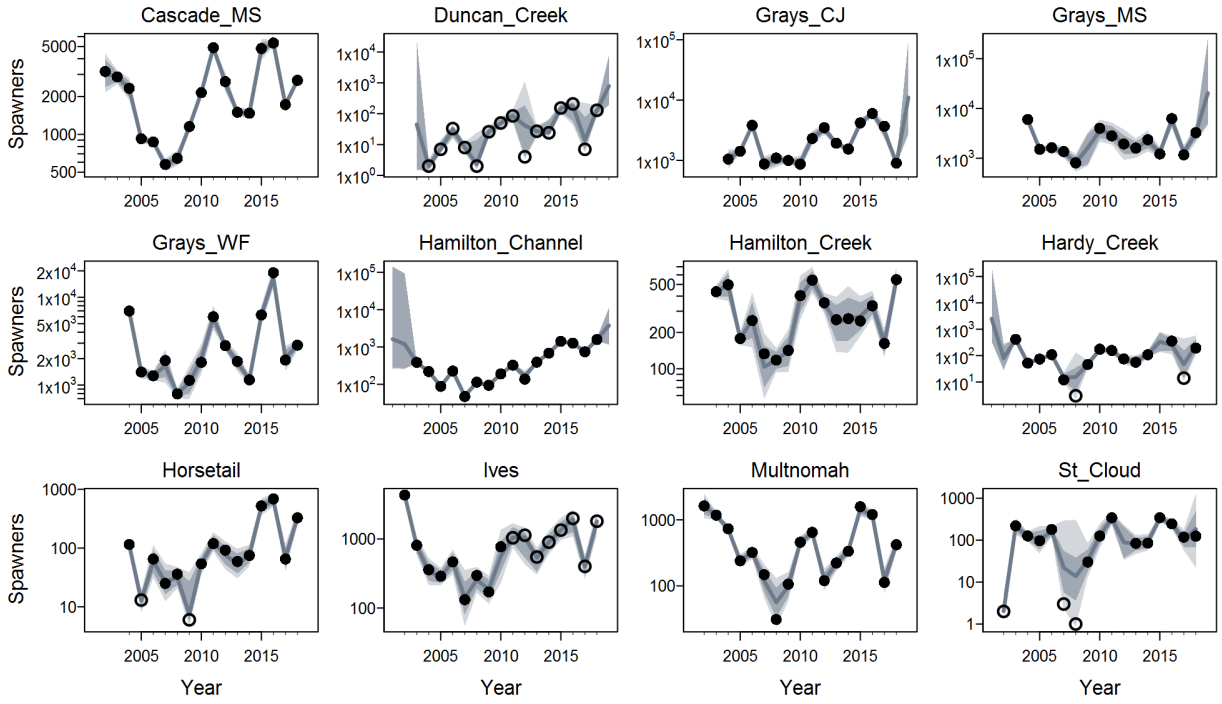

Figure 2: Observed (points) and estimated spawner abundance for Lower Columbia River chum salmon populations. Filled points indicate known observation error SD, while SD for open points is imputed. The posterior median (solid gray line) is from the multi-population IPM. Posterior 90% credible intervals indicate process (dark shading) and observation (light shading) uncertainty.

And here are the fits to the much sparser smolt data:

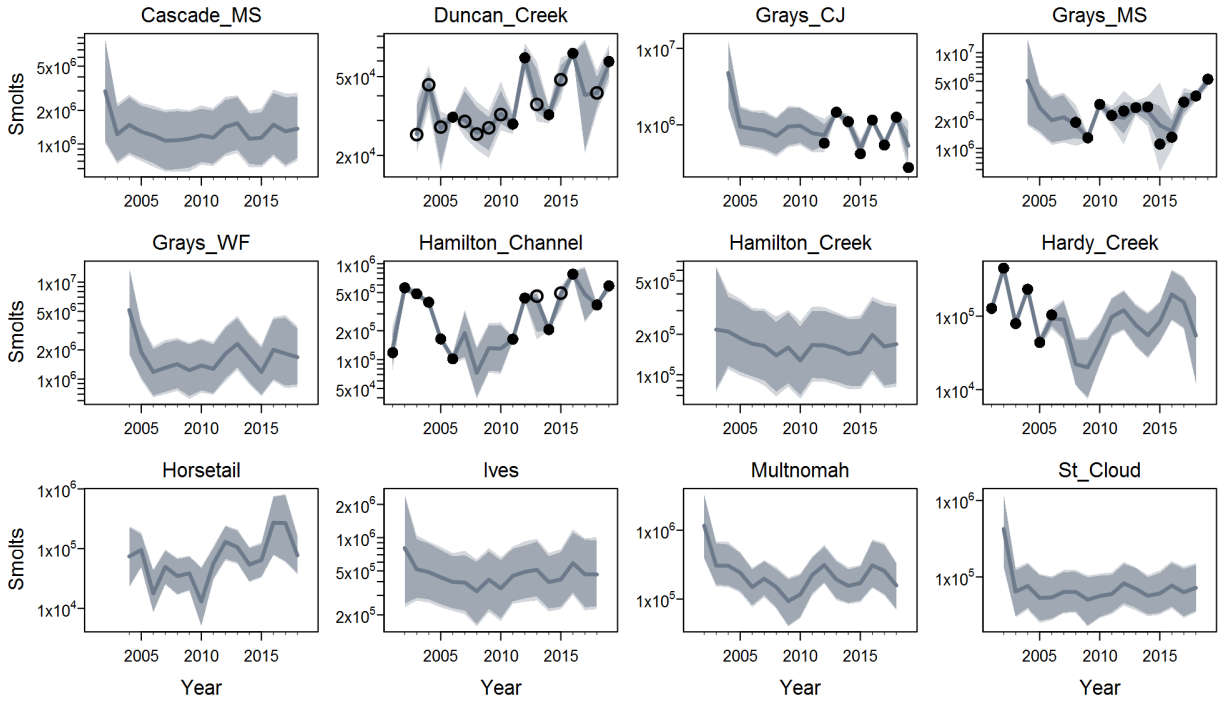

Figure 3: Observed (points) and estimated smolt abundance for Lower Columbia River chum salmon populations. Filled points indicate known observation error SD, while SD for open points is imputed. The posterior median (solid gray line) is from the multi-population IPM. Posterior 90% credible intervals indicate process (dark shading) and observation (light shading) uncertainty.

We can examine how the model partitions shared interannual fluctuations between the two life-stage transitions. . .

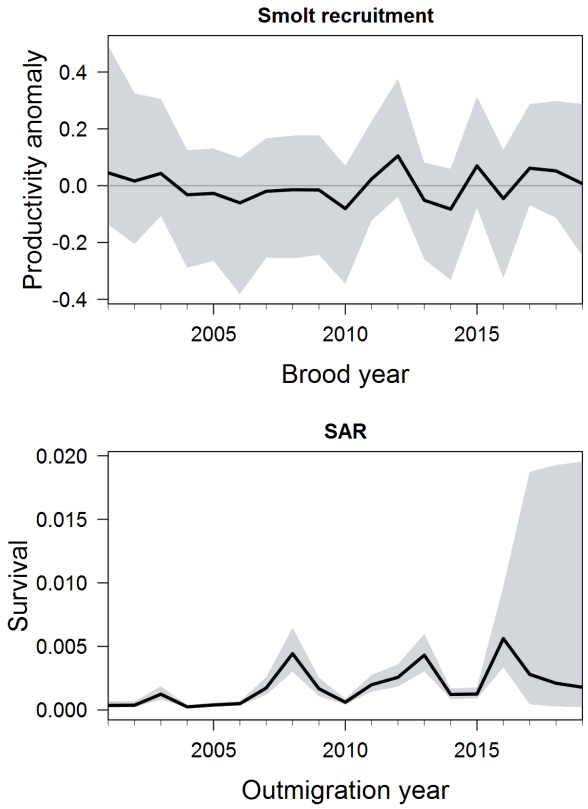

Figure 4: Estimates of shared (ESU-level) process errors from the multi-population IPM fitted to Lower Columbia River chum salmon data. The top panel shows the shared anomalies around the Ricker spawner recruit function, and the bottom panel shows the average SAR.

4 Forecasting

It is straightforward to use the IPM to generate forecasts of population dynamics...

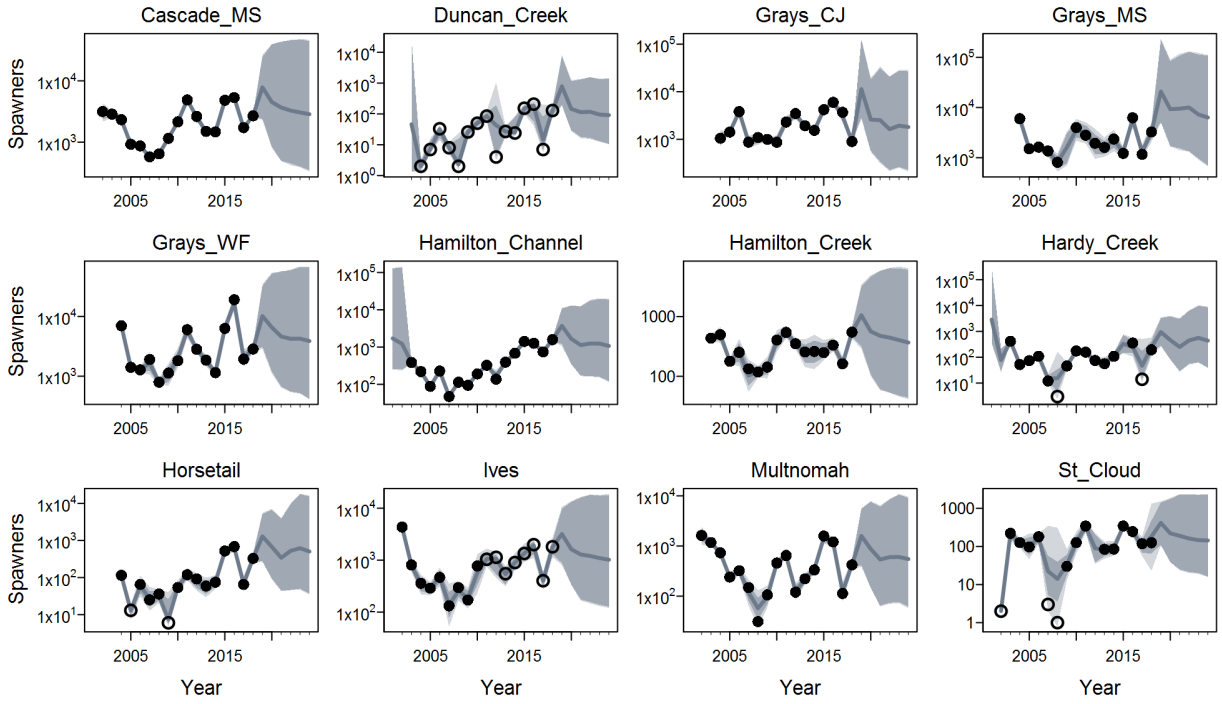

Figure 5: Observed (points) and estimated spawner abundance for Lower Columbia River chum salmon populations, including 5-year forecasts. Filled points indicate known observation error SD, while SD for open points is imputed. The posterior median (solid gray line) is from the multi-population IPM. Posterior 90% credible intervals indicate process (dark shading) and observation (light shading) uncertainty.

Of course we could also look at forecasts of smolts, or any other state variable. Here are the 2020 forecasts of wild spawners for each population...

Population	Estimate
Cascade_MS	4504 (850, 39237)
Duncan_Creek	145 (30, 1151)
Grays_CJ	2603 (572, 18013)
Grays_MS	9283 (1696, 84267)
Grays_WF	6556 (1203, 51771)
Hamilton_Channel	1660 (354, 12861)
Hamilton_Creek	562 (107, 4667)
Hardy_Creek	461 (83, 3790)
Horsetail	675 (95, 6737)
Ives	1618 (339, 13216)
Multnomah	869 (145, 7630)
St_Cloud	221 (39, 1748)
Total	32980 (8287, 241557)