ISC/24/ALBWG-01/05

Brief instruction and developing plan of the SSfuture C++.[1]

Hirotaka Ijima and Yuichi Tsuda

Highly Migratory Resources Division, Fisheries Stock Assessment Center, Fisheries Resources Institute (FRI), Japan Fisheries Research and Education Agency. 2-12-4 Fukuura, Kanazawa-ku, Yokohama, Kanagawa, 236-8648, JAPAN

ijima_hirotaka69@fra.go.jp

[1] This working paper was submitted to the ISC Albacore Working Group Intercessional Workshop, 11-18 March 2024, held at the National Oceanic and Atmospheric Administration Southwest Fisheries Science Center, Victoria, Canada

Abstract

The ISC Albacore Working Group utilizes its proprietary software, SSfuture C++, for future projecting of the albacore tuna stock. This software consists of R functions capable of seamlessly integrating the results of Stock Synthesis 3 (SS3) and a future projection program implemented using Rcpp, ensuring consistency with SS3's future projection results. However, SS future C++ has yet to be publicly released. This document explains the usage, release process, and development plans for SSfuture C++ to maintain transparency in stock assessment.

Introduction

The International Scientific Committee for Tuna and Tuna-Like Species in the North Pacific Ocean (ISC) Pacific Bluefin Tuna Working Group (PBFWG) developed its own future projection program, SSfuture, to account for uncertainty in future stock recruitments (Ichinokawa 2012). An essential aspect of this software is its consistency with the Stock Synthesis 3 (SS3) results. Uncertainty in initial values is also addressed through bootstrapping, and both constant catch and constant fishing mortality management scenarios are available. On the other hand, the ISC Albacore Tuna Working Group (ALBWG) used SSfuture for future predictions in the stock assessments of 2011 and 2014. However, in the 2017 assessment, changes to the stock assessment model for sex-specific population dynamics compromised the consistency between SSfuture and SS3. To address this issue, Ijima et al. 2016 developed SSfuture C++ to accommodate sex-specific population dynamics. This software enhances efficiency in stock assessment through accelerated computation using Rcpp.

However, this program has yet to be publicly released, and specific usage instructions and transparency need improvement. In the latest stock assessment, updated SSfuture C++ was utilized for future projections under a constant fishing intensity scenario, but some issues need improvement in other options. Thus, this document provides detailed explanations of the release process of SSfuture C++, program description and usage instructions, identified issues, and future development plans.

Publication

We will publish SSfuture C++ programs on GitHub. While access rights will generally be set to "open," there may be instances where access is restricted to certain members depending on the version. The version used in the latest resource assessment is “ssfcpp_simple_2023.cpp”, and future numbering will follow the format ssfcpp.v2.xx.yy, where xx corresponds to the version of SS3 and yy signifies other updates. Initially, the program used in the latest stock assessment (ssfcpp_simple_2023.cpp) will be made public, with the next version numbered ssfcpp.v2.00.00.

Program Description and Usage

Management scenario

SS3 offers three methods for estimating initial condition uncertainty: maximum likelihood estimate for variance-covariance matrix estimation, bootstrap, and Markov Chain Monte Carlo (MCMC) for posterior distribution estimation. Although SSfuture C++ allows for future projections using results from all three methods, it's crucial to be cautious when using them because SS3 often produces different estimates across these methods (Stewart et al. 2013). Additionally, since the initial values for uncertainty estimation differ for each method, it's important to verify them thoroughly.

SSfuture C++ can handle both process errors and implementation errors in projections. The recruitment process is modeled using a Beverton-Holt relationship, assuming spawner-recruitment relationships, with process errors following a lognormal distribution. Furthermore, it can incorporate autocorrelation into the process errors. Fishing intensity changes across all time intervals can test implementation errors.

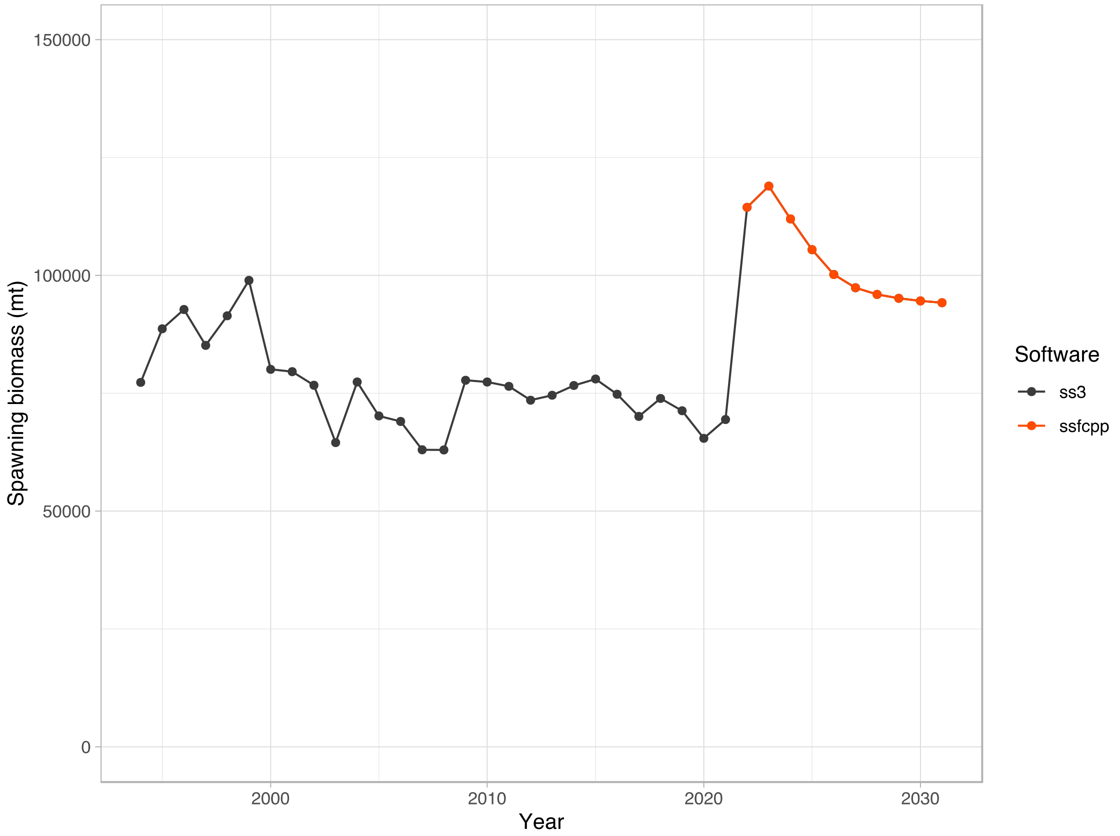
Management scenario

The original version of SSfuture C++ exhibited capabilities for conducting future projections under constant fishing intensity, constant catch, and simplistic Harvest Control Rules (HCR). However, the current version of SS future C++ exclusively accommodates fishing intensity management scenarios, which undergo annual adjustments. Fishing intensity aligns with the Spawning Potential Ratio (SPR) analysis outcomes within SS3. Within SS3, SPR computes the average selectivity across designated intervals and modulates the F multiplier accordingly. The F multiplier, in conjunction with selectivity, determines age-specific fishing mortality. Within SS future C++, the option exists to externally configure the computation of the F multiplier and selectivity independent of SS3, thereby enabling more adaptable future prognostications. Nonetheless, within the present stock assessment, both the F multiplier and selectivity derive from SS3 outcomes, necessitating the specification of appropriate SPR and selectivity intervals and subsequent rerunning of SS3 to acquire them.

Simple example: constant intensity scenario

The SS future C++ comprises an R function for importing results from SS3 and the Rcpp program for future projections. Given that the F multiplier and selectivity composing the fishing mortality at age (FAA) are assumed to vary annually, it is necessary to set the F multiplier and selectivity for the number of repetitions over the future projection period. Based on the 2023 stock assessment results, the deterministic constant fishing pressure scenario is as follows.

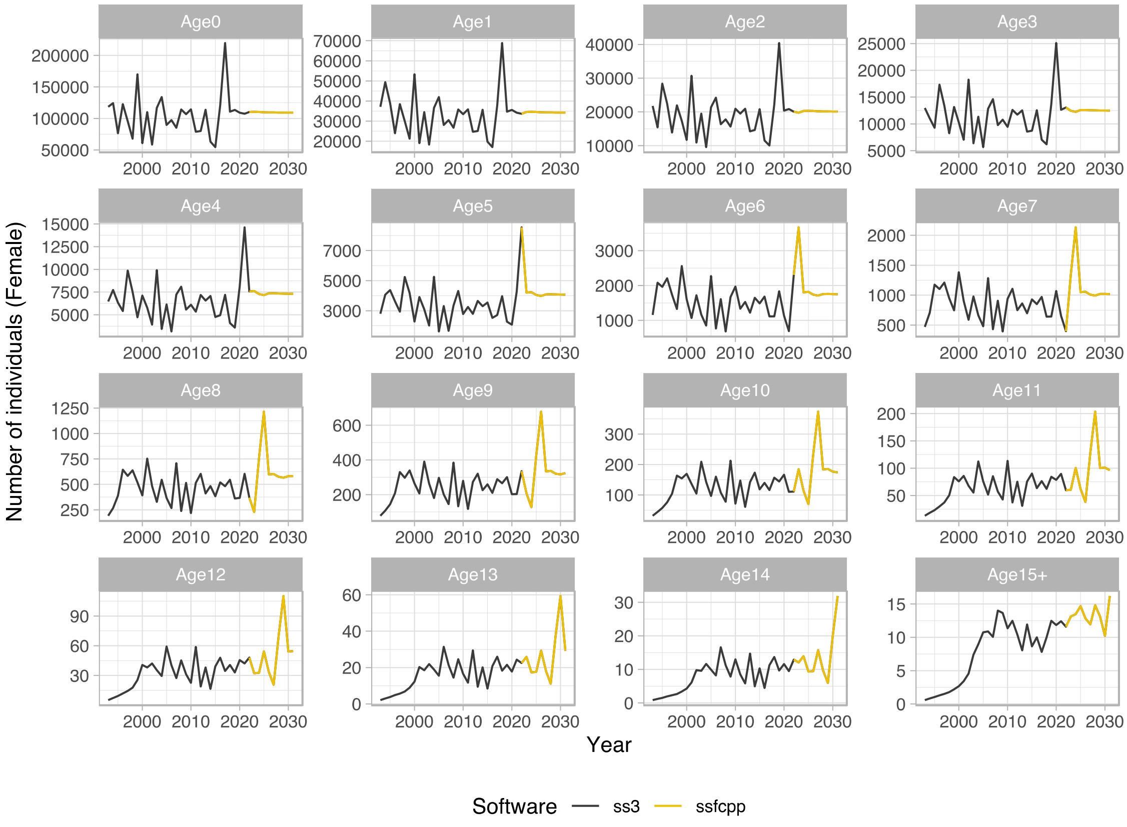
```
library(r4ss)
library(Rcpp)
library(tidyverse)
source("make_input_data.R") #R functions for ssfcpp
Sys.setenv("PKG_CXXFLAGS"="-std=c++0x")
sourceCpp("ssfcpp_constF.cpp")

replist = SS_output(dir = "2023npalbbasecasemodelfiles3")

# Make F multiplier vector using SS3 output
# (Option: fmsy, fspr, and bench) Start yr and end yr means the time period of selectivity
fmult = rep(NA_real_,10) # projection year X iteration number
for(i in 1:10) {
   fmult[i] = get_ref_point_f(replist, startyr=2018, endyr=2020, spr=0.3)$bench
}

# Make selectivity matrix using SS3 output
sel =matrix(NA_real_, nrow=10, ncol=(15+1)*4*2) #nrow: projection year X iteration number ncol:
(maxage+1) X (number of season) X (number of gender)
for (i in 1:10) {
   sel[i,] = get_sel(replist, startyr=2018, endyr=2020)$sel
}

# Set input data for ssfcpp
d = list(
   imax = 1, # Number of iterations
   tmax = 10, # Projection time
   SB0 = get_rec(replist)$SB0, # SSB0
   R0 = get_rec(replist)$R0, # R0
   h = get_rec(replist)$h, # Steepness
   recfun = 1, # Recruitment deviation pattern (1:deterministic, 2:simple deviation, 3:auto correlation)
   sigmar = get_rec(replist)$sigmar, # Sigma r
   rho = NA, # Autocorrelation parameter that is an option of recruitment when recfun=2.
   age = get_maa(replist)$age, # Max Age
   gender = get_maa(replist)$gender, # Gender type 1:one gender, 2:two gender
   spawn_flags = get_maa(replist)$birthseas, # Spawning season (0:no spawning, 1:spawning)
   int_n = get_naa(replist)$int_n, # Initial population number (quarter 1 in end year+1)
   maa = get_maa(replist)$maa, # Natural mortality at age
   waa_beg = get_maa(replist)$waa_beg, # Weight at age in the beginning of season
   waa_mid = get_maa(replist)$waa_mid, # Weight at age in the middle of season
   fec = get_maa(replist)$fec, # Maturity X Fecundity: SS3 configuration
   fmult = fmult, # F multiplier
   sel = sel # Selectivity
)

#Run SSfuture C++
set.seed(2024)
res_ssf = ssfcpp(d)

#SS3 results
res_ss = replist$sprseries

#Figure: Spawning biomass
ssb_ss = replist$timeseries %>% filter(Seas==2,Yr<=2031, Yr>=1994) %>%
              select(Yr,SpawnBio) %>%
              mutate(Software="ss3")

ssb_ssf = data.frame(Yr=rep(2022:2031),SpawnBio=res_ssf$sb_qt2[1,],Software="ssfcpp")

ssb = bind_rows(ssb_ss,ssb_ssf)

g = ggplot(data=ssb)
g = g + theme_light()
g = g + geom_line(aes(x=Yr,y=SpawnBio,color=Software))
g = g + geom_point(aes(x=Yr,y=SpawnBio,color=Software))
g = g + scale_colour_manual(values=c(ss3="#3B3B3BFF", ssfcpp="#ff4b00"))
g = g + ylim(c(0,150000))
g = g + xlab("Year") + ylab("Spawning biomass (mt)")
g
```



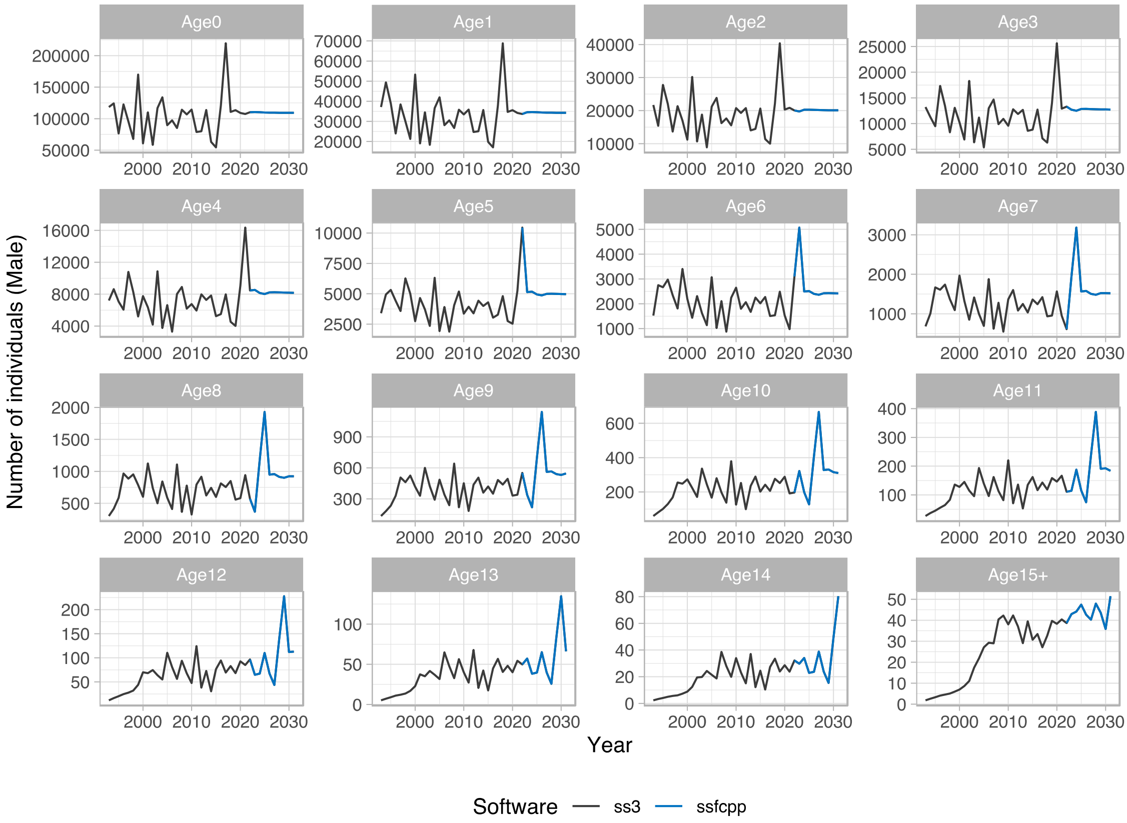
Year

```
#Figure: Population number at quarter 2
n_ss3 = replist$natage %>% filter(`Beg/Mid`=="B",Yr>=1993, Seas==2) %>%
select(-Area,-Bio_Pattern,-BirthSeas,-Settlement,-Platoon,-Morph,-Time,-Era,-`Beg/Mid`) %>%
rename(Age0=`0`,Age1=`1`,Age2=`2`,Age3=`3`,Age4=`4`,Age5=`5`,Age6=`6`,Age7=`7`,Age8=`8`,Age9=`9`,
Age10=`10`,Age11=`11`,Age12=`12`,Age13=`13`,Age14=`14`,`Age15+`=`15`) %>%
pivot_longer(col=c(-Sex,-Yr,-Seas), names_to="Age", values_to="Number") %>%
mutate(Software="ss3")

n_ssf = data.frame(res_ssf$N_qt2) %>%
mutate(Sex=c(rep(1,16),rep(2,16)), Age = factor(c(rep(0:15),rep(0:15)),labels=c("Age0", "Age1", "Age2",
"Age3", "Age4", "Age5", "Age6", "Age7", "Age8", "Age9", "Age10", "Age11", "Age12", "Age13", "Age14",
"Age15+"))) %>%
pivot_longer(col=c(-Sex,-Age), names_to="Yr", values_to="Number") %>%
mutate(Yr=rep(2022:2031,32), Software="ssfcpp")

n = bind_rows(n_ss3, n_ssf) %>% mutate(Age=factor(Age, levels=c("Age0", "Age1", "Age2", "Age3",
"Age4", "Age5", "Age6", "Age7", "Age8", "Age9", "Age10", "Age11", "Age12", "Age13", "Age14",
"Age15+")))

n_f = n %>% filter(Sex==1)
```

```
g = ggplot(data=n_f)
g = g + theme_light()
g = g + theme(legend.position = "bottom")
g = g + geom_line(aes(x=Yr, y=Number, group=Software, color=Software))
g = g + scale_colour_manual(values=c(ss3="#3B3B3BFF", ssfcpp="#EFC000FF"))
g = g + facet_wrap(vars(Age), scales = "free")
g = g + xlab("Year") + ylab("Number of individuals (Female)")
g = g + labs(color="Software")
g
```



```
n_m = n %>% filter(Sex==2)

g = ggplot(data=n_m)
g = g + theme_light()
g = g + theme(legend.position = "bottom")
g = g + geom_line(aes(x=Yr, y=Number, group=Software, color=Software))
g = g + scale_colour_manual(values=c(ss3="#3B3B3BFF", ssfcpp="#0073C2FF"))
g = g + facet_wrap(vars(Age), scales = "free")
g = g + xlab("Year") + ylab("Number of individuals (Male)")
g = g + labs(color="Software")
g
```



Difficulties and Future Development Plans

Age Length Key (ALK) and Maturity at Age (MAA)

Further modifications are needed for SS future C++ to accommodate the flexible configurations of SS3. Notably, key information concerning age and length (Age length key, ALK) and maturity at age (Maturity at age, MAA) are significantly influenced by the settings of SS3, necessitating accurate incorporation of their outcomes. Differences in spawning months lead to variations in ALK for calculating spawning stock biomass (SSB). While MAA can be configured based on age or length, aligning it with the appropriate ALK is imperative when length-based settings are applied. Moreover, it is essential to examine how SS3 handles the population dynamics of species such as skipjack and yellowfin tuna, which spawn throughout the year, and adjust accordingly.

Constant catch scenario

The latest SS3 can execute the constant catch future projection. We will compare the result of the constant catch management scenario with the SS future C++. If any issues arise, we will rectify the program.

Initial condition

Currently, the ISC ALBWG utilizes the output results of the variance-covariance matrix from SS3 to construct a multivariate normal (MVN) distribution, generating initial values as random numbers. However, SS3's inability to output covariance matrices for F multiplier and selectivity compromises the precision of uncertainty settings. As a current measure, the NAA constructs the MVN outputs with random numbers and recalculates the F multiplier to reproduce the catch with fixed selectivity.

Work plan

The ssfcpp.v2.00.00 release aims to address issues related to ALK and MAA, enabling the execution of the catch-based future projections. Its functionality will be validated using resource assessment results from MLS, BET, and ALB. Subsequently, ssfcpp.v2.00.01 will be released, incorporating a simple Harvest Control Rule (HCR) and undergoing additional analyses for academic publication.

References

Ichinokawa, M., 2012. Operating manual and detailed algorithms for conducting stochastic future projections with R packages of'ssfuture'. Nat. Res. Inst. Far Seas Fish., Shimizu, Shizuoka, Japan, Available at: http://cse. fra. affrc. go. jp/ichimomo/Tuna/projection_manual_v0, 4.

Ijima, H., Sakai, O., Akita, T., and Kiyofuji, H. 2016. New future projection program for North Pacific albacore tuna (Thunnus alalunga): considering two-sex age-structured population dynamics. ISC/16/ALBWG-02/06. Work. Doc. Submitt. to ISC Albacore

Stewart, I.J., Hicks, A.C., Taylor, I.G., Thorson, J.T., Wetzel, C. and Kupschus, S., 2013. A comparison of stock assessment uncertainty estimates using maximum likelihood and Bayesian methods implemented with the same model framework. Fisheries Research, 142, pp.37-46.