

Stock Assessment Modeling using JABBA

Dr. Luiz Ribas

Introduction

This document provides a walkthrough of the analytical framework used for modeling fish stock sustainability in the Itaipu Reservoir, in line with the analyses presented in the publication "*Evaluating 36 years of fishing sustainability in a large reservoir*" (Ribas et al., 2025). We use the R package **JABBA** (Just Another Bayesian Biomass Assessment) to fit surplus production models (SPMs) and evaluate Maximum Sustainable Yield (MSY) metrics.

Methodological Background

JABBA is a Bayesian state-space implementation of surplus production models designed for data-limited fisheries. It allows for flexible specification of prior distributions and integration of CPUE data and catch series into robust assessments of fishing sustainability. The method was introduced and validated in the following papers:

Winker, H., Carvalho, F., & Kapur, M. (2018). JABBA: Just Another Bayesian Biomass Assessment. **Fisheries Research**, 204, 275–288.
<https://doi.org/10.1016/j.fishres.2018.03.010>

Winker, H., Carvalho, F., Thorson, J.T., et al. (2020). JABBA-Select: Incorporating life history and fisheries' selectivity into surplus production models. **Fisheries Research**, 222, 105355. <https://doi.org/10.1016/j.fishres.2019.105355>

Environment Setup

First, load the required packages in R environment:

```
library(devtools)
library(usethis)
library(openxlsx)
library(readxl)
library(tidyverse)
library(JABBA)
library(rjags)
```

Then define the working directory and load your dataset:

```
setwd("path/to/your/folder")
data <- readxlsx("yourdata_file.xlsx") # Replace with your actual file
name
```

The dataset must include a time series with columns like:

- `Year`: integer year of observation
- `Biomass`: observed biomass (or catch, depending on setup)
- `CPUE`: catch per unit effort

```
biomass <- data.frame(year = data$Year, biomass = data$Biomass)
cpue <- data.frame(year = data$Year, cpue = data$CPUE)
```

Model Construction

Construct JABBA model inputs for various model structures and priors:

```
jbinput2 = buildjabba(catch = biomass, cpue = cpue, model.type =
"Pella")
jbinput4 = buildjabba(catch = biomass, cpue = cpue, model.type =
"Pella", psi.prior = c(1, 0.2))
jbinput6 = buildjabba(catch = biomass, cpue = cpue, model.type =
"Pella", igamma = c(0.001, 0.001), psi.prior = c(1, 0.2))
```

... Add Fox and Schaefer types similarly

```
modellist <- list(jbinput2, jbinput4, jbinput6)
```

Model Fitting and Selection

Fit each model and extract DIC for model comparison:

```
model_j_stats <- matrix(NA, nrow = length(model_list), ncol = 2)
for (j in 1:length(model_list)) {
  model_j = fit_jabba(model_list[[j]], quickmcmc = TRUE)
  model_j_stats[j,] <- c(j, model_j$stats["DIC", "value"])
}
model_selected <- model_list[[which.min(model_j_stats[,2])]]
```

Final fit with MCMC:

```
modelfinal <- fitjabba(model_selected, quickmcmc = FALSE, ni = 30000,  
nt = 5, nb = 5000, nc = 5)
```

Plotting and Diagnostics

```
jbplot_trj(model_final, type = c("B", "F", "BBmsy", "FFmsy"))  
jbplot_spphase(model_final)  
jbplot_kobe(model_final)  
jbplot_biplot(model_final)  
jbplot_runstest(model_final)  
jbplot_ppdist(model_final)  
jbplot_mcmc(model_final)
```

Retrospective Analysis

```
hc1 <- hindcastjabba(modelselected, modelfinal, peels = 1:5)  
jbplotretro(hc1)
```

Forecasting

```
fw3 <- fw_jabba(model_final, nyears = 10, imp.yr = 1, quant = "Catch",  
type = "msy", stochastic = TRUE)  
jbplot_ensemble(fw3)
```

Reference to Published Study

This workflow replicates procedures from:

Ribas, L., Piana, P.A., Henn, C. et al. Evaluating 36 years of fishing sustainability in a large reservoir. **Scientific Reports** 15, 3748 (2025).
<https://doi.org/10.1038/s41598-025-88209-8>

This markdown serves as a replicable guide for stock assessment using **JABBA** under data-limited scenarios.