

How to interact With CASAL

C.Marsh

09/06/2020

Introduction

The aim of this document is to demonstrate how to use the abm to simulate data, and then some R-code on how you would read that simulated data in to other packages, in this case CASAL.

System requirements

- The following R-libraries installed

```
library(casal)
```

```
library(ibm)
```

```
library(ggplot2)
```

```
library(Dataweighting) for if you are applying data-weighting methods in simulations.
```

```
library(casal)
library(ibm)
library(ggplot2)
library(DataWeighting)
```

Read in a Single run

When the abm to get an idea on model behaviour using the following command in terminal `ibm -r > run.log`. If you want the abm to simulate multiple datasets there are a few methods. Firstly simulate data based on one process model cycle. This will run from start year to final year only once, and then simulate multiple datasets with observation error but the same expected value. This is done using the command `ibm -s 10 > sim.out`. Remember that you will need to include the `@reportblock` for simulated data **Note** you want to write each observation to a separate file, look in `reports.ibm` for an example. The alternative is to simulate a single dataset after each model cycle with multiple model cycles. This is done by supplying an input file `ibm -s 1 -i pars.txt > sim_multi.out`, the disadvantage to this, is it can be a lot slower if it takes 1-2 minutes to complete a model cycle then 10 simulations will take 10 minutes, but adds process variability. This will be needed for the tag-recapture observations as well.

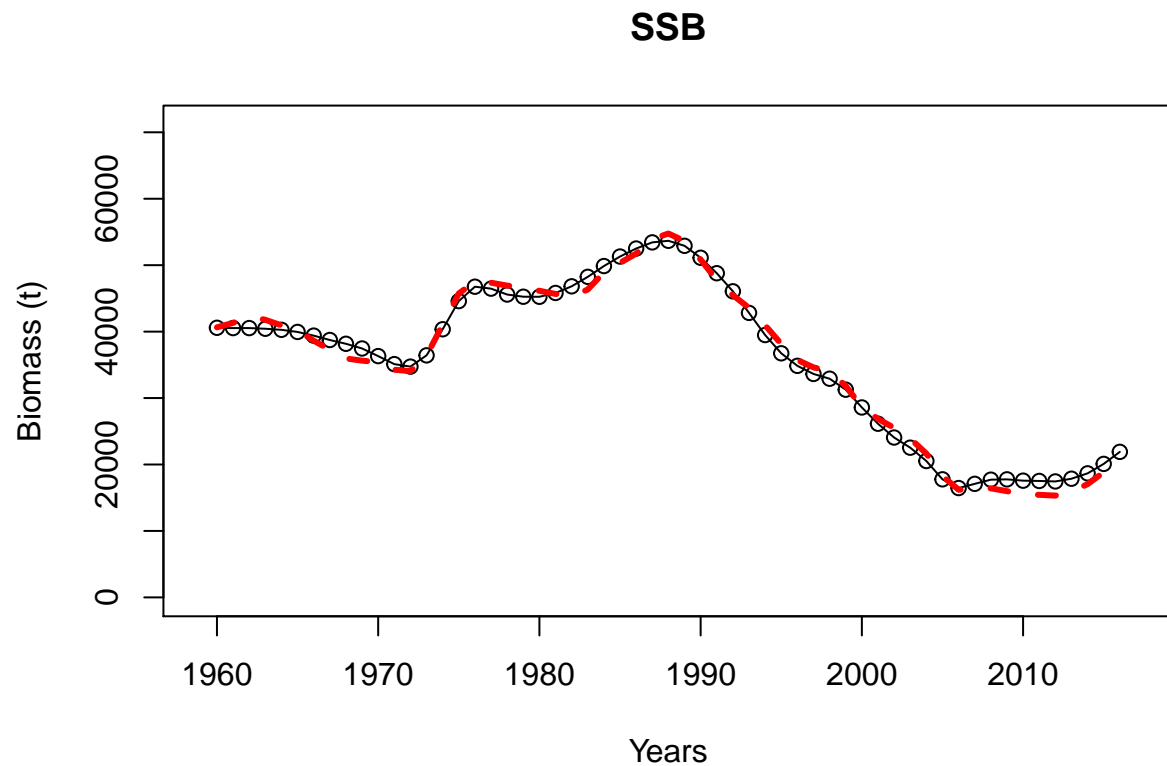
```
# this will create a list-like object called single_run
single_run <- extract.run(file = file.path("../", "abm", "run.log"))
# list of reports
names(single_run)
```

```
## [1] "init_1"          "init_2"          "age_freq"
## [4] "agents"          "Fishing"         "Rec"
## [7] "chatTANsel"      "chatFwestSel"    "chatFeastSel"
## [10] "derived_quants"  "model_attributes" "chatTANbiomass"
## [13] "chatTANage"      "chatOBSwst"      "chatOBSest"
## [16] "F_age"           "warnings_encounted" "model_run_time"
```

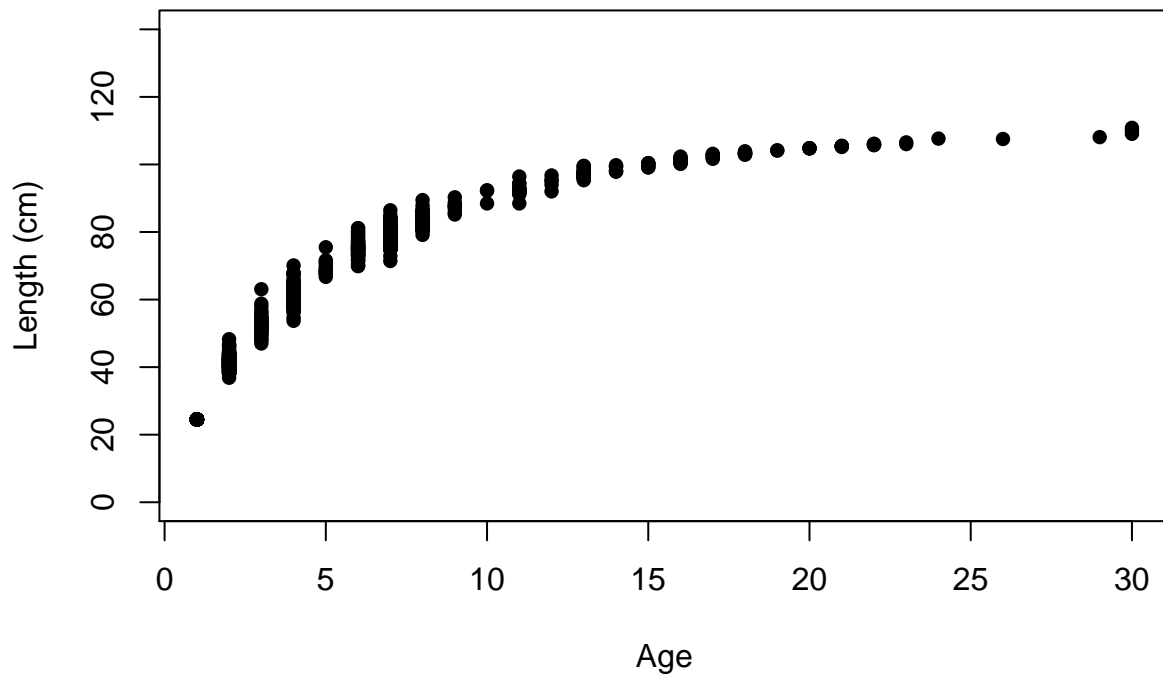
```
with_time_vary_m <- extract.run(file = file.path("../", "abm", "run_time_vary.out"))
```

```
plot.derived_quantities(single_run, report_label = "derived_quants")
```

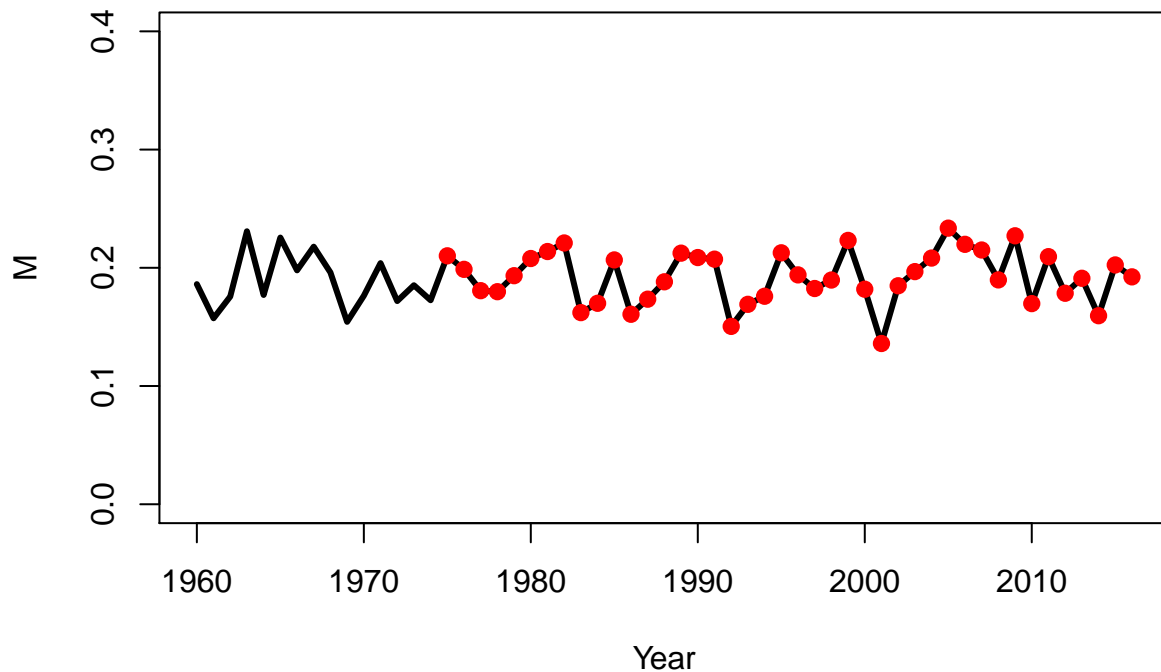
```
time_vary_ssb = with_time_vary_m$derived_quants$SSB
lines(names(time_vary_ssb$values), time_vary_ssb$values, lwd = 3, col = "red", lty = 2)
```



```
## plot age-length
plot(single_run$agents$`1975`$values$age, single_run$agents$`1975`$values$length,
     ylim = c(0, 140), xlab = "Age", ylab = "Length (cm)", pch = 16)
```



```
## check time-vary M applied correctly
plot(with_time_vary_m$time_vary$natMort$year, with_time_vary_m$time_vary$natMort$Value,
      type = "l", lwd = 3, ylim = c(0, 0.4), xlab = "Year", ylab = "M")
years = as.numeric(names(with_time_vary_m$agents)[names(with_time_vary_m$agents) !=
"year"]))
for (i in 1:length(years)) {
  this_ndx = with_time_vary_m$agents[[which(names(with_time_vary_m$agents) == years[i])]]
  points(years[i], mean(this_ndx$values$M), col = "red", pch = 16, cex = 1.2)
}
```



```
## plot observation over the time-horizon obs_time_series =
## plot_obs_timeseries(model_years = single_run$model_attributes$model_years,
## ibm_model = single_run) obs_time_series
```

Run a ReEstiamtion

```
# Get location of simulated data
abm <- extract.run(file = file.path("../", "abm", "run.log"))

sim_obs = file.path("../", "abm", "sim")
sim_file_names = unique(sapply(strsplit(list.files(sim_obs), split = "\\."), "[",
1))
extensions = unique(sapply(strsplit(list.files(sim_obs), split = "\\."), "[", 2))
N_sims = length(extensions)
SSBs = list()
output_list = list()

## read in Casal estimation csl file
cas_mpd = casal::extract.csl.file(file = file.path("../", "CASAL", "estimation.csl"))
## make a copy for safe keepings casal::write.csl.file(cas_mpd, file =
## file.path('../', 'CASAL', 'estimation_orig.csl'))

sim_chat_bio_ndx = which(sim_file_names %in% "chatTANbiomass")
sim_chat_age_ndx = which(sim_file_names %in% "chatTANage")
sim_east_ndx = which(sim_file_names %in% "chatOBSest")
```

```

sim_west_ndx = which(sim_file_names %in% "chatOBSwst")

##### Main loop this will have to be manually edited
abm_ages = abm$model_attributes$ages

for (sim in 1:N_sims) {
  # Chat bio
  sim_ibm = ibm::extract.ibm.file(file = file.path(sim_obs, paste0(sim_file_names[sim_chat_bio_ndx],
    ".", extensions[sim])))
  year_ndx = match(sim_ibm$`observation[chatTANbiomass_sim]`$years$value, names(cas_mpd$`relative_abun
  for (i in 1:length(year_ndx)) cas_mpd$`relative_abundance[chatTANbiomass]`[[year_ndx[i]]] = sim_ibm
  # Chat age
  sim_ibm = ibm::extract.ibm.file(file = file.path(sim_obs, paste0(sim_file_names[sim_chat_age_ndx],
    ".", extensions[sim])))
  year_ndx = match(sim_ibm$`observation[chatTANage_sim]`$years$value, names(cas_mpd$`proportions_at[ch
  max_age = as.numeric(cas_mpd$`proportions_at[chatTANage]`$max_class)
  min_age = as.numeric(cas_mpd$`proportions_at[chatTANage]`$min_class)
  for (i in 1:length(year_ndx)) {
    cas_mpd$`proportions_at[chatTANage]`[[year_ndx[i]]] = as.numeric(sim_ibm$`observation[chatTANage
    ## truncate
    if (cas_mpd$`proportions_at[chatTANage]`$plus_group %in% c("1", "TRUE", "T",
      "True"))
      cas_mpd$`proportions_at[chatTANage]`[[year_ndx[i]]][max_age] = sum(cas_mpd$`proportions_at[
    cas_mpd$`proportions_at[chatTANage]`[[year_ndx[i]]] = cas_mpd$`proportions_at[chatTANage]`[[year
    cas_mpd$`proportions_at[chatTANage]`[[year_ndx[i]]] = cas_mpd$`proportions_at[chatTANage]`[[year
  }
  # Chat west
  sim_ibm = ibm::extract.ibm.file(file = file.path(sim_obs, paste0(sim_file_names[sim_west_ndx],
    ".", extensions[sim])))
  year_ndx = match(sim_ibm$`observation[chatOBSwst_sim]`$years$value, names(cas_mpd$`catch_at[chatOBS
  max_age = as.numeric(cas_mpd$`catch_at[chatOBSwst]`$max_class)
  min_age = as.numeric(cas_mpd$`catch_at[chatOBSwst]`$min_class)
  for (i in 1:length(year_ndx)) {
    cas_mpd$`catch_at[chatOBSwst]`[[year_ndx[i]]] = as.numeric(sim_ibm$`observation[chatOBSwst_sim]
    ## truncate
    if (cas_mpd$`catch_at[chatOBSwst]`$plus_group %in% c("1", "TRUE", "T", "True"))
      cas_mpd$`catch_at[chatOBSwst]`[[year_ndx[i]]][max_age] = sum(cas_mpd$`catch_at[chatOBSwst]`
    cas_mpd$`catch_at[chatOBSwst]`[[year_ndx[i]]] = cas_mpd$`catch_at[chatOBSwst]`[[year_ndx[i]]][m
    cas_mpd$`catch_at[chatOBSwst]`[[year_ndx[i]]] = cas_mpd$`catch_at[chatOBSwst]`[[year_ndx[i]]]/s
  }
  # Chat east
  sim_ibm = ibm::extract.ibm.file(file = file.path(sim_obs, paste0(sim_file_names[sim_east_ndx],
    ".", extensions[sim])))
  year_ndx = match(sim_ibm$`observation[chatOBSest_sim]`$years$value, names(cas_mpd$`catch_at[chatOBS
  max_age = as.numeric(cas_mpd$`catch_at[chatOBSest]`$max_class)
  min_age = as.numeric(cas_mpd$`catch_at[chatOBSest]`$min_class)
  for (i in 1:length(year_ndx)) {
    cas_mpd$`catch_at[chatOBSest]`[[year_ndx[i]]] = as.numeric(sim_ibm$`observation[chatOBSest_sim]
    ## truncate
    if (cas_mpd$`catch_at[chatOBSest]`$plus_group %in% c("1", "TRUE", "T", "True"))
      cas_mpd$`catch_at[chatOBSest]`[[year_ndx[i]]][max_age] = sum(cas_mpd$`catch_at[chatOBSest]`

```

```

    cas_mpd$`catch_at[chatOBSEst]`[[year_ndx[i]]] = cas_mpd$`catch_at[chatOBSEst]`[[year_ndx[i]]][m
    cas_mpd$`catch_at[chatOBSEst]`[[year_ndx[i]]] = cas_mpd$`catch_at[chatOBSEst]`[[year_ndx[i]]]/s
  }
  ## write estimation.csl.
  casal::write.csl.file(cas_mpd, file = file.path("../", "CASAL", "estimation.csl"))

  ## estimate and data weighting
  setwd(file.path("../", "CASAL"))
  for (i in 1:5) {
    shell(paste0("cmd & casal -e > ReEstimateStand\\mpd_", sim, ".out 2> AD.out"))
    est_mpd = casal::extract.mpd(paste0("ReEstimateStand\\mpd_", sim, ".out"))
    cas_mpd_est_alt = casal::extract.csl.file(file = file.path("../", "CASAL",
      "estimation.csl"))
    error_ndx = grepl(names(cas_mpd_est_alt$`proportions_at[chatTANage]`), pattern = "N_")
    chat_age_weight = Method.TA1.8(est_mpd$fits$chatTANage, plotit = T)
    cas_mpd_est_alt$`proportions_at[chatTANage]`[error_ndx] = lapply(cas_mpd_est_alt$`proportions_a
      FUN = function(x) {
        as.numeric(x) * chat_age_weight
      })

    error_ndx = grepl(names(cas_mpd_est_alt$`catch_at[chatOBSEst]`), pattern = "N_")
    chat_west_weight = Method.TA1.8(est_mpd$fits$chatOBSEst, plotit = F)
    cas_mpd_est_alt$`catch_at[chatOBSEst]`[error_ndx] = lapply(cas_mpd_est_alt$`catch_at[chatOBSEst]
      FUN = function(x) {
        as.numeric(x) * chat_west_weight
      })

    error_ndx = grepl(names(cas_mpd_est_alt$`catch_at[chatOBSEst]`), pattern = "N_")
    chat_east_weight = Method.TA1.8(est_mpd$fits$chatOBSEst, plotit = F)
    cas_mpd_est_alt$`catch_at[chatOBSEst]`[error_ndx] = lapply(cas_mpd_est_alt$`catch_at[chatOBSEst]
      FUN = function(x) {
        as.numeric(x) * chat_east_weight
      })
    casal::write.csl.file(object = cas_mpd, file = file.path("../", "CASAL", "estimation.csl"))
  }
}

```