Evaluate performance of POP-based estimator with known ages

Felix T Petersma

2022-07-17

Introduction

The initial negative log-likelihood in `Rcpp` with unknown ages of parents and offspring did not produce the desired results. The better understand what might be wrong, I decided to start with an estimator that uses known ages directly. Once that one would work well enough, I could build from there by first allowing for unknown age of the offspring, followed by unknown age of the parent. I simulated 100 populations in the scripts `source/fitting/GR_sims_mix.R`. These simulated data sets are stored in `data/100_sims_dfs_suff.RData`. Here, I will load these data and fit the simple POP-based CKMR model to them.

Fit the model

```
## Load data
load("../data/100_sims_dfs_suff_correct.RData")

## Load libraries
library(Rcpp)
library(parallel)
library(pbapply)

## Source cpp script
sourceCpp("../source/fitting/nllCKMRcppAgeKnown.cpp")

## Fit 100 models
result_list <- pblapply(dfs_suff, function(df) {
  par <- list(
    # phi = boot::logit(0.87), # same as plogis(0.9) -- boot::inv.logit() is qlogis()
    N_t0_m = log(500),
    r = log(1.03),
    N_t0_f = log(500))
  # sigma_vbgf = log(2))

  df_select <- df[, ]

  dat <- list(alpha_m = 10,
              alpha_f = 12,

              # r = log(1.01),
              # sigma_vbgf = log(0.000001),
              phi = boot::logit(0.87),

              max_age = 19,
              t0 = 140,
```

```
              # vbgf_l_inf = 175,
              # vbgf_k = 0.1,
              # vbgf_t0 = -3.5,
              s1 = df_select$indiv_1_sex,
              s2 = df_select$indiv_2_sex,
              c1 = df_select$indiv_1_capture_year,
              c2 = df_select$indiv_2_capture_year,
              a1 = df_select$indiv_1_age,
              a2 = df_select$indiv_2_age,
              pair_found = df_select$pop_found,
              cov_combo_freq = df_select$covariate_combo_freq,
              n = nrow(df_select))

  res <- nlminb(start = par,
                objective = nllPOPCKMRcppAgeKnown,
                dat = dat,
                control = list(trace = 0))

  return(res)

})
```

Evaluate the fit

We have several ways of exploring the fits. Some of the most important ones are the plots

```
## Extract male and female abundance in reference year 140
N_male <- sapply(result_list, function(res) {
  exp(res$par["N_t0_m"])
})
N_female <- sapply(result_list, function(res) {
  exp(res$par["N_t0_f"])
})
## Extract r
r_est <- sapply(result_list, function(res) {
  exp(res$par["r"])
})

## Get the summaries
summary(N_male)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   547.0   820.4  1012.6  1086.5  1286.1  2458.3
```

```
summary(N_female)
```

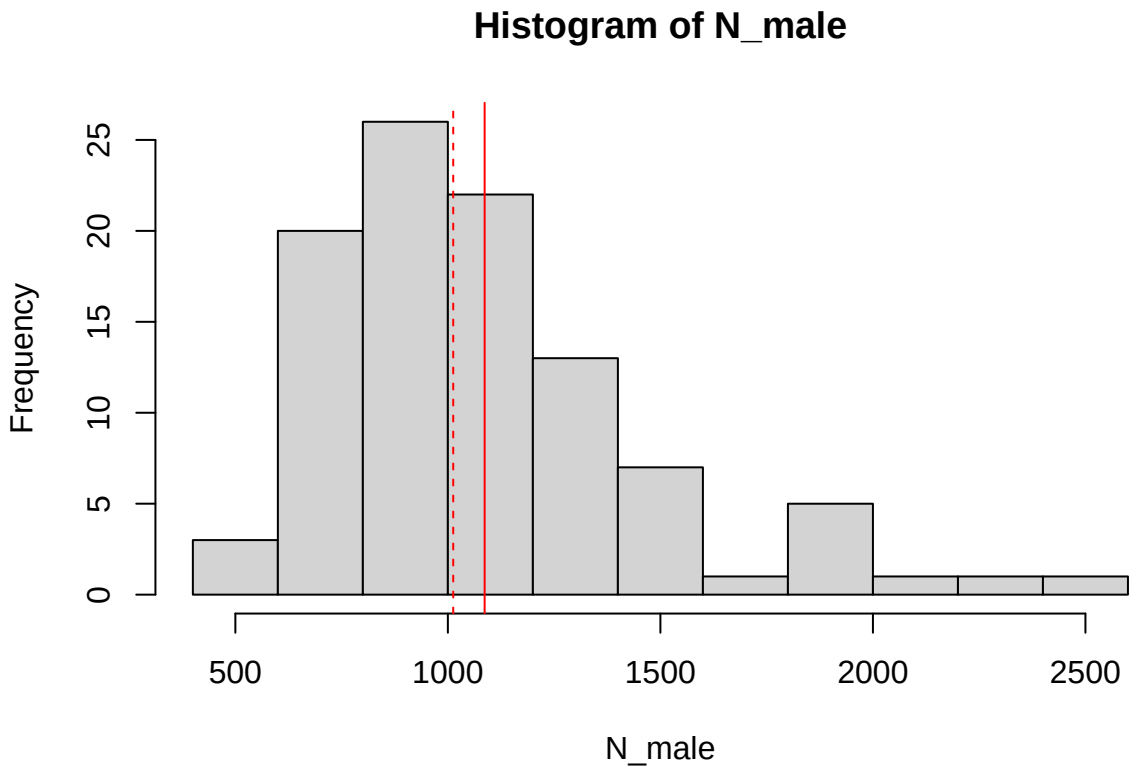
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   325.7   448.9   568.7   617.5   736.9  1452.2
```

```
summary(r_est)
```

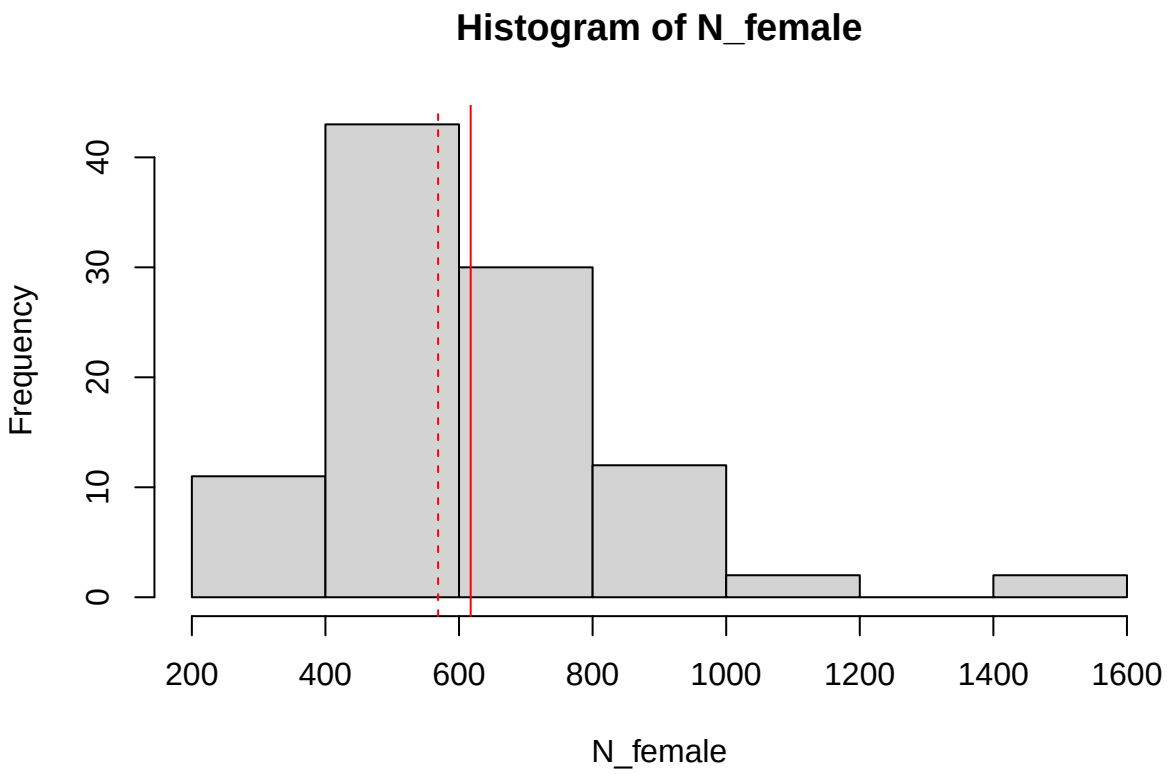
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8196  0.9628  1.0047  1.0145  1.0782  1.2186
```

```
## Create histograms
hist(N_male);
```

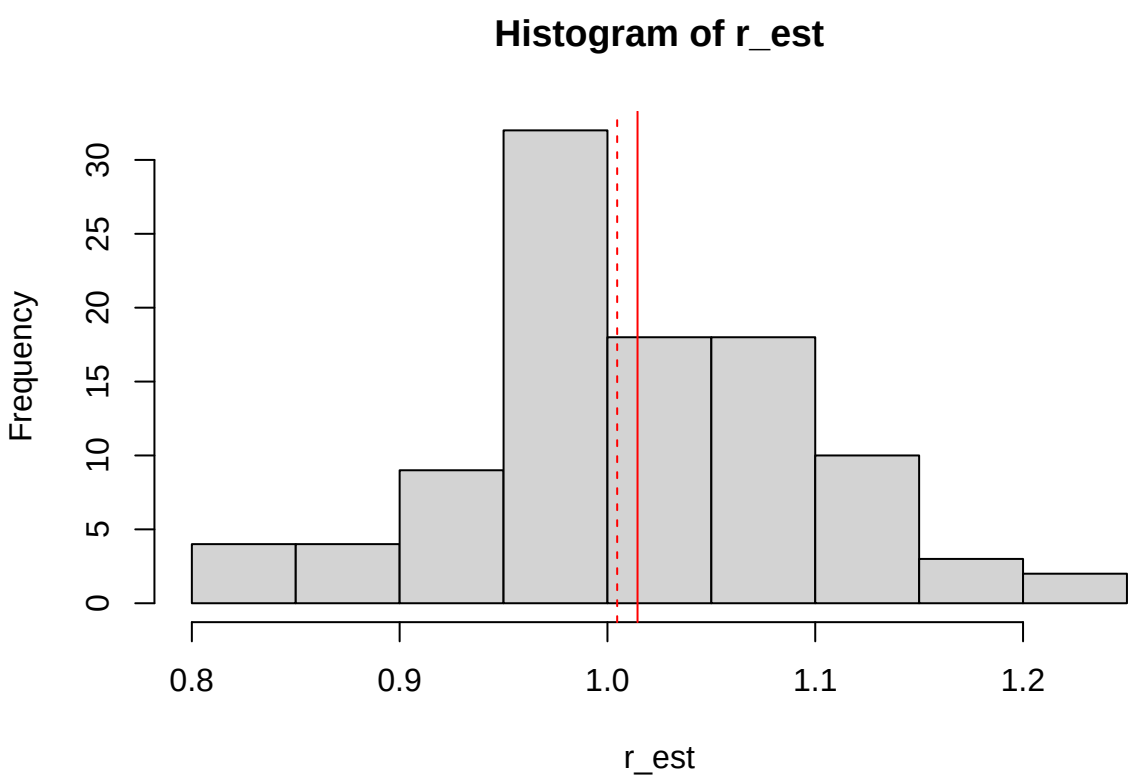
```
abline(v = mean(N_male), col = "red");
abline(v = median(N_male), col = "red", lty = "dashed")
```




```
hist(N_female);
abline(v = mean(N_female), col = "red");
abline(v = median(N_female), col = "red", lty = "dashed")
```




```
hist(r_est);
abline(v = mean(r_est), col = "red");
abline(v = median(r_est), col = "red", lty = "dashed")
```


Histogram of r_est
Frequency
0
5
10
15
20
25
30
0.8
0.9
1.0
1.1
1.2
r_est