The broad steps I’m taking to simulate a population of lemon sharks (or cownose rays) and fit a CKMR model are as follows:

# **Simulation Parameters**

These are the parameters that are setting up the individual-based simulation.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value for Lemon Sharks** | **Value for Cownose Ray** |
| **Initial population size** | 3000 | 3000 |
| **Operating sex ratio** | 0.5 | 0.5 |
| **Age of reproductive maturity** | 12 | 7 |
| **Maximum age** | 30 | 19 |
| **Mating periodicity** | 2 (years) | 1 (year) |
| **Potential number of mates** | 1:3 (per year) | 1 (per year) |
| **Average number of offspring per mate** | 3 | 1 |
| **Birth sex ratio** | 0.5, 0.5 | 0.5, 0.5 |
| **Adult survival** | 0.85 | 0.9 |
| **Juvenile survival** | 0.85 | 0.86 |
| **Young-of-year survival** | 0.65 | 0.75 |
| **Years of simulation** | 60 | 90 |
| **Sample years** | 58:60 | 58:60 |
| **Sample size per year** | varies | Focusing on 150 for now |

# **CKMR parameters**

These are the parameters that are being used or estimated in the CKMR model.

year\_est: This is the year on which the abundance estimate is focused.

Lambda: I’ve been fixing lambda to the mean population growth rate from the birth year of the oldest sampled individual (min\_cohort) to the last year of the simulation. In the model, I make abundance in a given year a function of abundance in year\_est x lambda raised to the number of years between year\_est and the younger sibling birth year (ys\_birth).

Adult survival: I’ve been fixing this value to the mean observed survival rate of adults through the simulation.

Female abundance: Trying to estimate.

Male abundance: Trying to estimate.

# **Kinship probability**

Where

k = individual adult female

F = all adult females in population

ys\_ birth = the birth year of the younger individual in the comparison

os\_birth = the birth year of the older individual in the comparison

yr\_est = the birth year of the oldest sampled individual i.e. the year for which we’re estimating abundance

= survival of mother k from the year of the older sibling birth to the year of the younger sibling birth.

* Note that the younger individual birth year will be a larger number than the older individual birth year (e.g. 2021 vs 2010)
* I have been fixing this value to the observed survival of adults through the simulation

= the number of total females alive in the year of estimation

* This is the value I’m estimating

= The mean population growth rate raised to the number of years between the birth year of the younger individual and the year of estimation (i.e. birth year of oldest sampled individual).

* I have been fixing this value to the mean population growth between yr\_est and the last year of the simulation.
* This seems to have a major effect on the estimate of NF
* I’m presently trying to include this as an estimated parameter, but constraining it to +/-1%. I wonder if giving this parameter a little wiggle room will help.

The same equation above is constructed for males. The male and female equation are saved together as an R function called get\_P\_lemon. The negative log likelihood of this function is minimized with respect to the parameters of interest (male and female abundance).

# **Process**

1. I simulate a population setting the simulation parameters above to emulate either lemon shark or cownose ray life history.
2. After the simulation, I randomly sample the population over multiple years.
   1. I filter the sample set for:
      1. Recaptures of self (keep just one instance of capture for each individual)
      2. Full siblings (if found, keep just one)
3. Then, I set up a pairwise comparison matrix, where each individual is compared against every other.
   1. I remove within-cohort comparisons from the pairwise comparison matrix before proceeding. This is supposed to help ensure that comparisons are independent, since the variance in reproductive output should not affect the probability of recapture (i.e. sampling a half-sib on the second occasion)
4. Half siblings are identified
5. The pairwise comparison matrix is split into four separate matrices:
   1. Maternal half-sibs
   2. Paternal half-sibs
   3. Maternal negative comparisons
      1. Includes all comparisons that were not maternal half-sibs
   4. Paternal negative comparisons
      1. Includes all comparisons that were not paternal half-sibs
6. I fit the CKMR model to the data and maximize the likelihood with respect to abundance using a Hessian matrix and BFGS method
   1. Initial parameter values for male and female abundance are set to the values at the beginning of the simulation.
7. I calculate standard error with the delta method (though I don’t really understand this yet)
8. I determine the true value of adult abundance by extracting the number of adult males and females from the year of estimation
9. I examine the relative bias via

# **Kinship probability code**

Note:

Pars1 = vector of parameter values (in this case female abundance and male abundance)

maxAge = maximum age (obviously)

min\_est\_cohort = the birth year of the oldest sampled individual

f\_age\_at\_mat = the year at maturity for females (m\_age\_at\_mat is for males)

The code below is for cownose rays. The functions for lemon sharks are identical.

#Set up empty array that will be filled with function below

P\_Mother = P\_Father = array(0,dim=c(n\_yrs,n\_yrs)) #Dimensions are older sib birth year and younger sib birth year (all of which are specified by n\_yrs)

#CKMR model: populate array with kinship probabilities based on birth years

get\_P\_cownose <- function(Pars1,P\_Mother,P\_Father,t\_start,t\_end){

N\_F=exp(Pars1[1]) #number of mature females

for(os\_birth in min\_est\_cohort:(n\_yrs-1)){

for(ys\_birth in (os\_birth+1):n\_yrs){

if((ys\_birth - os\_birth) <= ((maxAge+1) - f\_age\_at\_mat)){

#Fill in array with kinship probability from CKMR equation

P\_Mother[os\_birth, ys\_birth] <- (surv^(ys\_birth - os\_birth))/(N\_F\*lam^(ys\_birth-min\_est\_cohort))

} else P\_Mother[os\_birth, ys\_birth] <- 0

}

}

##Repeat the above with males

N\_M=exp(Pars1[2]) #number of mature males

for(os\_birth in min\_est\_cohort:(n\_yrs-1)){

for(ys\_birth in (os\_birth+1):n\_yrs){

if((ys\_birth - os\_birth) <= ((maxAge+1) - m\_age\_at\_mat)){

#Fill array with kinship probability from half-sib CKMR equation

P\_Father[os\_birth,ys\_birth] <- (surv^(ys\_birth - os\_birth))/(N\_M\*lam^(ys\_birth-min\_est\_cohort))

} else P\_Father[os\_birth,ys\_birth] <- 0

}

}

return(list(P\_Mother=P\_Mother, P\_Father=P\_Father)) #return makes sure this is moved out of the loop into the environment

}

# **Likelihood function**

Note: The Negatives and Pairs dataframes refer to the separate pairwise comparison matrices for positive and negative kinship comparisons.

cownose\_neg\_log\_lik <- function(Pars1, Negatives\_Mother, Negatives\_Father, Pairs\_Mother, Pairs\_Father, P\_Mother, P\_Father, t\_start, t\_end) {

P=get\_P\_cownose(Pars1 = Pars1, P\_Mother = P\_Mother, P\_Father = P\_Father, t\_start = t\_start, t\_end = t\_end)

loglik=0

#likelihood contributions for all negative comparisons

for(irow in 1:nrow(Negatives\_Mother)){

loglik = loglik + Negatives\_Mother[irow, 3] \* log(1 - P$P\_Mother[Negatives\_Mother[irow, 1], Negatives\_Mother[irow, 2]])

}

for(irow in 1:nrow(Negatives\_Father)){

loglik = loglik + Negatives\_Father[irow, 3] \* log(1 - P$P\_Father[Negatives\_Father[irow, 1], Negatives\_Father[irow, 2]])

}

#likelihood contributions for positive comparisons

for(irow in 1:nrow(Pairs\_Mother)){

loglik = loglik + Pairs\_Mother[irow, 3] \* log(P$P\_Mother[Pairs\_Mother[irow, 1], Pairs\_Mother[irow, 2]])

}

for(irow in 1:nrow(Pairs\_Father)){ loglik = loglik + Pairs\_Father[irow, 3] \* log(P$P\_Father[Pairs\_Father[irow, 1], Pairs\_Father[irow, 2]])

}

-loglik

}

# **Results**

The models that I’ve been using have returned unbiased results when fit to data from a population simulated from a Leslie Matrix, where kinship was assigned based on the probabilities specified in the CKMR model (Figure 1).

Chart, box and whisker chart

Description automatically generated

**Figure 1:** Relative bias by sample size under three scenarios of population growth. The populations were simulated from a Leslie Matrix, and kinship was assigned based on the probabilities specified in the model. As sample size increases, so does precision. These results suggest that at the population level, half-sibling CKMR performs as expected.

However, when I use individual-based simulation and fit a model to those data, the model consistently returns estimates that are positively biased. I have tried this with both Lemon Shark (6 pups per female) and Cownose Ray (1 pup per female) life histories and they all give positively biased estimates (Figure 2A).

Timeline

Description automatically generated

**Figure 2:** Relative bias by sample size. This simulation was run to test whether there's a difference in bias when ages are misassigned. The key reason to include this here is that the estimates in A are positively biased, even when ages are correct.

# **Troubleshooting**

I’ve run various tests to examine where the bias in Figure 2 might originate. None have proven successful so far.

**Test 1: Include only one individual from each litter**

**Why?** CKMR assumes that the odds of sampling an individual are independent of that individual’s reproductive output. This forms the basis of the Expected Relative Reproductive Output (ERRO) framework outlined in Bravington et. al. (2016). If multiple litter mates are included in the comparison, then the kinship probabilities are dependent not just on the expected reproductive output, but also the variance.

**Test 2: Use cownose ray life history (1 pup per female per year)**

**Why?** In case the bias is caused by persistent variation in litter size among individuals, using a cownose ray life history – where individuals have just one pup per year – should clear up that issue.

**Test 3: Allow lambda to vary in the model**

**Why?** Using a CKMR model that relies on population growth (lambda) to set a single year for abundance estimation results in a model that is sensitive to the value of lambda. But with a non-deterministic population, lambda varies among years. So, I copied code from Paul Conn to include lambda as an estimable parameter, constraining it to +/-1%.

# **Presently …**

04/22/20121

**Current test 1:** I am looping over different values of lambda and survival and seeing where the discrepancy lies.

* values of lambda from obs\_lambda-0.01 : obs\_lambda+0.01, by = .002
* values of survival from obs\_survival-.02 : obs\_survival+0.02, by = 0.01

**Current test 2:** I am removing lambda and setting the true value to the average abundance from all the cohort years.

Taking post-breeding census i.e. age 0 individuals are included.

Does mortality occur before birth/sampling?

Sampling occurs from each dataframe after the entire simulation is run.

Are adults reproductively mature AT age 7?

YES

**Order:**

Birthdays

Mate

Mortality

Sampling occurs after birthday and mate, but doesn’t include mortality.