# Tutorial – *MRmix*

Authors: Saang-Yoon Hyun, Joel Reynolds, and Peter Galbreath

School for the Marine Science and Tech., University of Massachusetts Dartmouth, MA

US Fish and Wildlife Service, Anchorage, AK

Columbia River Inter-Tribal Fish Commission, Portland, OR

## What is *MRmix*?

In a mark-recapture experiment, it is rare to fully incorporate tag loss from a mixture of single- and double-tags into estimation of animal abundance in a closed population. Our model, MRmix estimates abundance, tag loss rate, capture probabilities, and the uncertainty of those estimates, using the maximum likelihood method. It is written in ADMB (Fournier 2007).

## Download

Executable file: MRmix.exe

Example data file: sample.dat

## Step-by-step tutorial

1. Once you download **MRmix.exe**, and **sample.dat**, then you change input values in

**sample.dat** using your own data.

1. Open MS DOS window (aka MS-DOS Shell, or Command Prompt), and make sure that you are in the directory (folder) where the executable and data files are downloaded. Type

  **MRmix –ind sample.dat**

then click ENTER. The model will run, generating a series of output files.

1. Out of the output files, mrmix.std has key results (i.e., estimates of parameters with standard error estimates).

## Document that describes detailed theories

Hyun, S., J.H. Reynolds, and P.F. Galbreath. 2011. Accounting for tag loss and its uncertainty in a mark-recapture study with a mixture of single- and double tags. Trans. Am. Fish. Soc. In press.

## References

Fournier, D.A. 2007. An introduction to AD Model Builder Version 8.0.2 for use in nonlinear modeling and statistics. Otter Research Ltd., Sidney, B.C., Canada.

## Codes in ADMB

DATA\_SECTION

 //Binary question for tag loss(=1), OR no loss of tag (=0);

 init\_int LossTag\_NolossTag;

 //Binary question: (i) both tags are identical non-permanent tags(=1), OR

 //(ii) one tag is non-permanent and the other tag is permanent (=0);

 init\_int bothNP\_oneP;

 //1st sample for marking

 init\_number M\_1\_A; //number of animals single-tagged in color 1

 init\_number M\_2\_B; //number of animals double-tagged in color 2

 //2nd sample for recapture or resight

 init\_number n2; //number of fish captured later (they include marked and

 //unmarked fish); n2 in a common notation

 init\_number mA; //recoveries with only one tag in color 1

 init\_number mB; //recoveries with only one tag in color 2

 init\_number mBB; //recoveries with two tags in color 2

 init\_number rho\_guess; //guess initial value of rho;

 number n1; //total number of initially tagged animals

 number R; //total recoveries = mA + mB + mBB;

 number mU; //number of animals in the 2nd sample, which do not have tags

 //mU = n2 - (mA + mB + mBB), if there is loss of a tag

 //mU = n2 - (mA + mBB), else if no loss of a tag;

 !!n1=(M\_1\_A+M\_2\_B);

 int dim; //dimension in data and parameters between loss and no loss of a tag;

 int rho\_phz; //consider or ignore by the assumption

 LOCAL\_CALCS

 if(LossTag\_NolossTag==1) {

 dim=4;

 rho\_phz=4;

 }

 else {

 dim=3;

 rho\_phz=-4;

 };

 END\_CALCS

 vector celldat1(1,dim);

 LOCAL\_CALCS

 if(LossTag\_NolossTag==1) {

 celldat1(1)=mA;

 celldat1(2)=mB;

 celldat1(3)=mBB;

 celldat1(4)=n2-(celldat1(1)+celldat1(2)+celldat1(3)); //animals with no tags

 R=(mA+mB+mBB);

 }

 else { // No loss of a tag

 celldat1(1)=mA;

 celldat1(2)=mBB;

 celldat1(3)=n2-(celldat1(1)+celldat1(2)); //animals with no tags

 R=(mA+mBB);

 };

 END\_CALCS

 !!mU=n2-R;

 number N\_low;

 number N\_up;

 number N\_guess;

 vector N\_boundary(1,2);

 !!N\_boundary(1)=n2;

 !!N\_boundary(2)=n1;

 !!N\_guess=(n1+1)\*(n2+1)/(R+1)-1;

 !!N\_low=max(N\_boundary);

 !!N\_up=N\_guess+N\_guess\*100/100;

 number p\_guess; //capture probability

 !!p\_guess=(n1+n2)/(2\*N\_guess);

 number stp; //single tag proportion, stp = M\_1/n1;

 !!stp=M\_1\_A/n1;

 !!cout<<"R "<<"n1 "<<"N\_low "<<"N\_up "<<"stp"<<endl;

 !!cout<<R<<" "<<n1<<" "<<N\_low<<" "<<N\_up<<" "<<stp<<endl;

PARAMETER\_SECTION

 init\_bounded\_number N(N\_low,N\_up,1); //N: abundance;

 init\_bounded\_number p1(0.0,1.0,2); //capture prob at the 1st sample;

 init\_bounded\_number p2(0.0,1.0,3); //capture prob at the 2nd sample;

 init\_bounded\_number rho(0.0,1.0,rho\_phz); //rho: tag loss rate;

 vector theta(1,dim+1); //cell prob

 vector celldat2(1,dim+1); //cell data 2;

 number Esc; //unknown escapment = N-n2

 objective\_function\_value f; //negative total logarithm

INITIALIZATION\_SECTION

 N N\_guess;

 p1 p\_guess;

 p2 p\_guess;

 rho rho\_guess;

PROCEDURE\_SECTION

 f=0.0;

 Esc=N-n2; //animals not captured in the second sample; //Escapement;

 for(int i=1;i<=dim;i++)

 celldat2(i)=celldat1(i);

 celldat2(dim+1)=Esc;

 if(LossTag\_NolossTag==1) {

 if(bothNP\_oneP==1) {

 theta(1)=M\_1\_A\*(1-rho)\*p2/N; //for data on mA

 theta(2)=M\_2\_B\*2\*(1-rho)\*rho\*p2/N; //for data on mB or mF

 theta(3)=M\_2\_B\*(square((1-rho)))\*p2/N; //for data on mBB or mAF

 theta(4)=M\_1\_A\*rho\*p2/N+M\_2\_B\*(square(rho))\*p2/N+(N-n1)\*p2/N; //for data on mU

 theta(5)=1-(theta(1)+theta(2)+theta(3)+theta(4)); //for 5th cell of unknown escapement (=N-n2)

 }

 else {

 theta(1)=M\_1\_A\*(1-rho)\*p2/N; //for data on mA

 theta(2)=M\_2\_B\*rho\*p2/N; //for data on mB or mF

 theta(3)=M\_2\_B\*(1-rho)\*p2/N; //for data on mBB or mAF

 theta(4)=M\_1\_A\*rho\*p2/N+(N-n1)\*p2/N; //for data on mU

 theta(5)=1-(theta(1)+theta(2)+theta(3)+theta(4));

 };

 }

 else { //No loss of a tag

 theta(1)=M\_1\_A\*p2/N; //for data on mA

 theta(2)=M\_2\_B\*p2/N; //for data on mBB or mAF

 theta(3)=(N-n1)\*p2/N; //for data on mU

 theta(4)=1-(theta(1)+theta(2)+theta(3)); //for 4th cell of unknown escapement (=N-n2)

 };

 //binomial

 f+=-1.0\*gammln(N+1)+gammln(N-n1+1)-n1\*log(p1)-(N-n1)\*log(1-p1);

 //multinomial

 theta/=sum(theta);

 f+=-1.0\*gammln(N+1)+gammln(N-n2+1)-sum(elem\_prod(celldat2,log(theta)));

 //f+=-1.0\*gammln(N+1)+gammln(N-C+1)-sum(elem\_prod(celldat2,log(theta)))-N\*(1-sum(theta)); //N plays a role of Lagrange Multiplier

 // cout<<f<<" ";

REPORT\_SECTION

 report<<"N guess"<<" "<<"p guess"<<" "<<"single tag proportion"<<endl;

 report<<N\_guess<<" "<<p\_guess<<" "<<stp<<endl;

 report<<"mU"<<" "<<"rho"<<" "<<"N"<<" "<<"p1"<<" "<<"p2"<<endl;

 report<<mU<<" "<<rho<<" "<<N<<" "<<p1<<" "<<p2<<endl;

 report<<"M\_A"<<" "<<"M\_BB"<<" "<<"total M"<<endl;

 report<<M\_1\_A<<" "<<M\_2\_B<<" "<<n1<<endl;

 report<<"celldat1 :"<<celldat1<<endl;

 report<<"celldat2 :"<<celldat2<<endl;

 report<<"N\_low and N\_up: "<<N\_low<<" "<<N\_up<<endl;

RUNTIME\_SECTION

 maximum\_function\_evaluations 100,150,300,10000

 convergence\_criteria .01,.0001,1e-7

TOP\_OF\_MAIN\_SECTION

 gradient\_structure::set\_MAX\_NVAR\_OFFSET(1000); //maximum number of depdendent variables of 400 exceeded

 gradient\_structure::set\_NUM\_DEPENDENT\_VARIABLES(1000);

 gradient\_structure::set\_GRADSTACK\_BUFFER\_SIZE(100000);

 gradient\_structure::set\_CMPDIF\_BUFFER\_SIZE(1000000);

 arrmblsize=900000;

GLOBALS\_SECTION

 #include <admodel.h>

 #include <math.h>

 #include <stdio.h>

 #include <stddef.h>

 #include <stdlib.h>