Development of a Boundary Setting Algorithm based on migration rates estimated using BayesAss and its preliminary application to TOSSM datasets

Charles T. T. Edwards*[1] and Douglas S. Butterworth[1]
**contact email*: edwcha001@uct.ac.za

Abstract

A Boundary Setting Algorithm (BSA) is developed that utilises migration rates estimated by the program BayesAss. It is subjected to preliminary testing through application of the BSA to a subset of TOSSM datasets from Archetypes I and II. Two Fully Internally Mixed Areas (FIMAs) are considered, connected by varying rates of migration. The objective of the BSA, implemented through the *run.tossm* control program, is to decide whether to manage FIMAs as one or two units. Performance is assessed using the depletion and catch statistics produced by *run.tossm*. Although BayesAss has the potential to provide information relevant to management, performance of the BSA was found to be poor, apparently due to inconsistent and unreliable estimation of the migration rate.

Keywords: TOSSM, BSA, BayesAss

Introduction

The effective management of exploited species requires identification of demographically isolated populations that can be considered as independent management units. The International Whaling Commission's Revised Management Procedure (RMP) [2] includes options to distribute catches within a given *Region* between a number of *Small Areas*, with each *Small Area* treated as if it contained an isolated population. However there is currently no formal basis for delineating *Small Areas* or the populations they encompass. The potential utility of genetic information to achieve this objective prompted the Testing of Spatial Simulation Models (TOSSM) project [3]. The aim of TOSSM is to investigate a number of existing genetic methods capable of defining certain attributes of population structure, and to test whether they may provide useful information for management.

[1] Department of Mathematics and Applied Mathematics, University of Cape Town, Rondebosch, 7701, South Africa.

In this investigation a Boundary Setting Algorithm (BSA), used to partition management units (MUs), is developed. The BSA is based on migration rates estimated by the BayesAss program. Genetic data were provided by the simulation program Rmetasim [13]. It was assumed that individuals have their own breeding populations, but are sampled from a number of Fully Internally Mixed Areas (FIMAs), which represent areas where harvesting takes place. The BSA groups different FIMAs into MUs, and the suitability of this grouping is assessed through future projections of population dynamics, under catches set by the RMP.

Methods

Genetic methods for distinguishing demographically isolated populations can be grouped into those that segregate a sample of individuals into distinct populations, thereby estimating population structure, and those that assume a specific structure and then estimate migration rates between the component populations. Because neither approach is likely to be sufficient for management purposes [4], it is probable that a combination will be required to define management units when using genetic samples from undefined populations.

Structural methods [6, 11] assign individuals to distinct populations on the assumption that allele frequencies within each population are in Hardy-Weinburg equilibrium and sites are close to linkage equilibrium. This requires a prior assumption about the number of populations that exist. If the number of populations is known, and samples of known origin are available from every one, then it is possible to assign individuals to populations other than those from which they were sampled, and therefore to classify them as migrants or of migrant ancestry [5, 12]. This is achieved using the transient linkage disequilibrium observed in multilocus immigrant genotypes. A further extension of this approach is to utilise migrant ancestries to explicitly estimate migration rates per generation [14]. This is possible because migration rates provide information on the probability of observing particular ancestries, with the precision of this probability improving with the number of generations considered. Here we implement this approach using the program BayesAss [14].

Theoretical background

BayesAss [14] estimates recent migration rates into a population using multilocus genotype data allowing for multiple alleles at each locus. Migration rate

m is measured as the proportion of the population that has immigrated from a given source, per generation. Notation for the model is specified as follows:

X is the multilocus diploid genotype of an individual;

S is an index that refers to the population from which the individual was sampled;

M is a population index that refers to the source of an individuals ancestry ($M \neq S$ implies the individual is an immigrant or of immigrant ancestry);

t is the time at which immigration occured, indexed in generations prior to the current generation (with $t = 0$ referring to the current generation);

$\mathbf{F}$ is the vector of inbreeding coefficients across populations; and,

$\mathbf{p}$ is a three dimensional array of allele frequencies at each locus in each population.

We assume that S is known and that all potential source populations have also been sampled. X is considered to be from a non-migrant, migrant (referred to as a first generation immigrant) or from an individual with one migrant parent (a second generation immigrant). These correspond to times of immigration $t = 0, 1$ and 2 respectively. The ancestry of each individual genotype can therefore be described by the values of M and t. Given $\mathbf{p}$, and values for M and t, the probability of an observed genotype $P(X \mid S)$ can be calculated assuming generalised Hardy-Weinburg equilibrium within each population. Deviations from Hardy-Weinburg frequencies are allowed through estimation of $\mathbf{F}$. Because allele frequencies are assumed to be at equilibrium, it is also necessary to assume that $\mathbf{p}$ is unchanged by migration (i.e. that m is small relative to effective population sizes).

The likelihood of a particular ancestry conditional on m, $L(M, t \mid m)$ is calculated making use of the assumption that m is small. This allows the probability (here equivalent to the likelihood) that the genotype of an individual is of a particular ancestry to be calculated using first order terms in m only. Specifically, if ϕ is the proportion of alleles in an individual that are of migrant ancestry, then the probability that an individual is a first generation migrant ($M \neq S, t = 1, \phi = 1$) is equal to m, and the probability that an individual has one migrant parent ($M \neq S, t = 2, \phi = \frac{1}{2}$) is $2m$. These allow calculation of $L(M, t \mid m)$ (see [14]). Earlier generations are not currently considered by the method because of the presumed statistical difficulties in detecting values of $\phi > \frac{1}{2}$ with a small number of loci. According to these prior assumptions we therefore expect a fraction m of the population to be migrants, $2m$ to have one migrant parent and the remainder to be non-migrants. This implies that $0 \leq m \leq \frac{1}{3}$.

Using Bayes theorem it is then possible to calculate the joint posterior probability density for any given X and S:

$$f(m, M, t, \mathbf{F}, \mathbf{p} \mid X, S) = \frac{L(X, S \mid M, t, \mathbf{p}, \mathbf{F}) L(M, t \mid m) f_m(m) f_p(\mathbf{p}) f_F(\mathbf{F})}{P(X \mid S)}$$

The parameters m, $\mathbf{p}$ and $\mathbf{F}$ are accorded uninformative (uniform) prior distributions $f_m(m)$, $f_p(\mathbf{p})$ and $f_F(\mathbf{F})$ respectively. The posterior probability is estimated using the Markov Chain Monte Carlo (MCMC) method.

The Boundary Setting Algorithm

The BSA was written as a R function that applies BayesAss to estimate migration rates between sampled populations using microsatellite loci. The R code is given in the Appendix. The BSA first formats the genetic data appropriately, calls BayesAss as an external program and then collects information on migration rates from the MCMC output. BayesAss has been modified so that the raw MCMC output for all parameters is available for analysis. Migration rates are estimated from the mean of the Bayesian posterior distribution and input into a square matrix representing migration rates into each population from every other population. The BSA then decides whether to consider each pair of populations (FIMAs) as either isolated or demographically linked. Two populations are grouped in the same MU if the estimated migration rate between them (in either direction) is greater than a specified critical value of m, termed *critical.m*. Specifically for two populations l and q, if $\{m_{l \to q} \cup m_{q \to l}\} \geq critical.m$ then l and q are grouped into the same MU. After pairwise groupings, the BSA will further group MUs together if populations in different MUs are shared. For example if $\{m_{1 \to 2} \cup m_{2 \to 1}\} \geq critical.m$ and $\{m_{2 \to 3} \cup m_{3 \to 2}\} \geq critical.m$ but $\{m_{1 \to 3} \cup m_{3 \to 1}\} \ngeq critical.m$, then pairwise groupings will give the pairs $\{1, 2\}$ and $\{2, 3\}$. Because population 2 is shared, the BSA will then combine these to give a single MU $\{1, 2, 3\}$.

Data and simulation procedure

The IWC Scientific Committee has identified five stock structure archetypes for use within the TOSSM framework [3]. Due to the preliminary nature of this work, the focus here is on a subset of scenarios from Archetypes I and II. Archetype I represents a panmitic population, and Archetype II two or three populations linked by migration. In this case, an evenly distributed population of 7500 individuals was assumed for both Archetypes. For Archetype II only the two population case was considered, with migration rates per year of 5×10^{-6}, 5×10^{-5}, 5×10^{-4} and 5×10^{-3}. A generation time of 20 years is assumed throughout, so that these rates correspond to migrant proportions of 1×10^{-4}, 1×10^{-3}, 1×10^{-2} and 1×10^{-1} per generation, respectively. Because populations sizes are equal and migration is symmetric, these can be considered as values of m. In this context Archetype I is considered to be the same as Archetype II with a migration rate of one, so that an effective total of five migration rates is considered. Unfortunately the data were not available for Archetype II with a migration rate of zero. The data sets utilised were therefore (in order of increasing migration rates): Arch2_sc2_1; Arch2_sc3_1; Arch2_sc4_1; Arch2_sc5_1; and, Arch1_sc1_1.

In accordance with previous work [9], implementations of the *run.tossm* control program specified a total annual catch of 300 for five years (split equally between FIMAs), followed by application of the BSA and a then 95 year projection period. During this period the RMP was applied with catches set according to the *Small Areas* (MUs) defined by the BSA, and population abundance estimates becoming available every five years. Note that within this implementation the BSA is not revisited during the 95 year projection period.

Experimental Design

The total population in each projection was divided into two FIMAs. For Archetype I, individuals were assigned randomly to each FIMA; for Archetype II, each breeding population had its own FIMA with no overlap. The objective of the BSA is to decide whether or not to manage each FIMA separately.

To calibrate the performance of the BSA, management was first simulated assuming fixed, predetermined boundaries: manage as either one contiguous or two separate populations. The BSA was then applied to each data set assuming *critical.m* values of 0.0, 0.1, 0.2 and $\frac{1}{3}$. Because $0 \leq m \leq \frac{1}{3}$ (see Theoretical background) it was expected that the extreme values of m would reproduce results obtained assuming fixed boundaries, with $critical.m = 0.0$ equivalent to a prespecified contiguous population and $critical.m = \frac{1}{3}$ equivalent to two separate populations. Intermediate values of *critical.m* were included to investigate

Table 1: Number of management units output by the BSA for 20 combinations of *critical.m* and migration rate

Migration Rate	*critical.m*			
	0.0	0.1	0.2	1/3
5×10^{-6}	1	1	2	2
5×10^{-5}	1	2	1	2
5×10^{-4}	1	1	1	2
5×10^{-3}	1	1	1	2
1.0	1	1	1	2

the performance of the BSA across a range of migration rates.

Because of computer time limitations, analysis was restricted to only one replicate for each combination of five migration rates and and four *critical.m* values. For each of these 20 replicates, 50 genetic samples were drawn from each FIMA with 10 microsatellite loci per sample [4]. When invoked by the BSA, BayesAss was run for 21×10^6 iterations with the first 2×10^6 discarded as burnin [7]. Samples were collected from the MCMC chain every 2000 iterations. Delta values for each parameter (which define the 'temparature' of the MCMC chain) were set to 0.10, giving acceptance rates of 20-60% [7]. The performance of each run in a management context was assessed using the following statistics [9]: i) Lowest depletion; ii) Final depletion; and, iii) Sum of catches during implementation of the RMP. Depletion was calculated as a percentage of the starting number of individuals for each breeding population, using the actual population numbers tracked by *run.tossm$abund.b*.

Results

The decision reached by each implementation of the BSA is given in Table 1. The outcomes for extreme values of *critical.m* are as expected. When $critical.m = 0.0$, FIMAs are returned by the BSA as one MU, and when $critical.m = \frac{1}{3}$ FIMAs are kept separate.

Depletion statistics following projections under each of the decisions listed in Table 1 are given in Tables 2 and 3. For reasons of clarity, where there are two underlying breeding populations, we only report depletion statistics for one. Specifically, if the two breeding populations are being managed as one

Table 2: Lowest recorded depletion

Migration Rate	Fixed	*critical.m*				Fixed
	1 MU	0.0	0.1	0.2	1/3	2 MU
5×10^{-6}	17.61	11.14	12.94	52.72	61.95	57.39
5×10^{-5}	5.36	12.79	60.25	17.80	62.34	56.38
5×10^{-4}	18.94	4.22	0.03	8.15	58.51	56.85
5×10^{-3}	8.36	25.03	4.10	16.02	54.98	56.86
1.0	53.25	54.41	56.13	57.78	58.18	54.07

Table 3: Final depletion

Migration Rate	Fixed	*critical.m*				Fixed
	1 MU	0.0	0.1	0.2	1/3	2 MU
5×10^{-6}	36.65	11.22	28.80	61.04	64.92	62.87
5×10^{-5}	5.36	12.90	65.27	36.52	68.76	56.62
5×10^{-4}	23.05	34.34	7.57	11.26	64.52	59.80
5×10^{-3}	28.10	40.76	45.98	34.52	61.15	56.86
1.0	63.10	61.51	58.81	60.20	60.27	61.13

MU (i.e. the two FIMAs are combined), we report depletion statistics only for the population being harvested (recall that if FIMAs are considered as a single MU, then catches are only taken from one FIMA) . If the breeding populations are managed separately, depletion for both is similar. Total catches from all breeding populations are given in Table 4.

Assuming fixed boundaries, it is clear that when two breeding populations exist, management outcomes are improved if FIMAs are defined as distinct MUs (so that each breeding population is managed separately): depletion is less severe and catches are higher. To emphasise this point: if breeding populations are considered as distinct MUs, not only does this reduce the chances of depletion (increases abundance), but it simultaneously increases the total harvested catch. When the population is panmitic (migration rate is one), depletion is similar irrespective of whether the population is considered as one or two management units, however catches are increased if the population is managed as a single MU.

Table 4: Sum of catches across breeding populations

Migration Rate	Fixed	*critical.m*				Fixed
	1 MU	0.0	0.1	0.2	1/3	2 MU
5×10^{-6}	7315	6235	6315	15940	16640	15190
5×10^{-5}	6755	6550	16095	6890	15545	15580
5×10^{-4}	6870	6030	4900	6760	16060	16370
5×10^{-3}	8050	10155	9460	10040	18535	18690
1.0	15980	16635	16860	16145	7120	8865

The primary concern here is with performance of the BSA at intermediate values of *critical.m*. However examination of the MCMC chains produced by BayesAss suggest that it may not have produced any meaningful or consistent information regarding the actual simulated migration rates. The MCMC chain appears to spend a large amount of time close to the boundary values of m, so that $m \approx 0.0, 0.3$. This seems to be a particular problem at low migration rates (Figure 1). At higher migration rates there is evidence for improved mixing of the MCMC chain (Figure 2) so that estimates of m may be more accurate (although clearly still not reliable). For migration rates of 5×10^{-3} this is true for each of the four BayesAss runs, making them the most informative indicators of performance of the BSA. However in each case migration rate was estimated to be high with $m > 0.2$ for at least one of the FIMAs. The BSA therefore joined FIMAs into single MUs, which in this case may be the least appropriate management option.

Discussion

Here a BSA based on migration rates estimated using BayesAss has been developed and applied to simulated TOSSM datasets in a preliminary manner through implementation of the *run.tossm* control program. Although the number of replicates was small, performance of the BSA appeared to be poor, largely due to inadequate mixing of the MCMC chain produced by BayesAss during estimation of m. Notably, mixing was not improved when simulations were re-run using 30 loci from each sampled individual (data not shown). In instances in which BayesAss has been applied to real datasets in the literature [1, 14], estimated migration rates frequently appear to be close to the bounds. This suggests that modifications to the program may have to be made

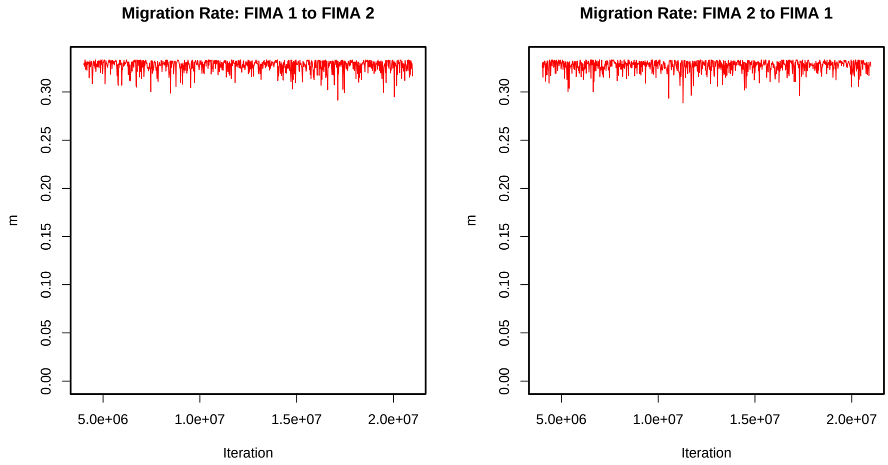

Figure 1: Representative MCMC output of migration rates estimated by BayesAss between two FIMAs connected by a migration rate of 5×10^{-6}, with the samples shown taken every 20,000 iterations and the first 2×10^{6} iterations discarded as burnin.

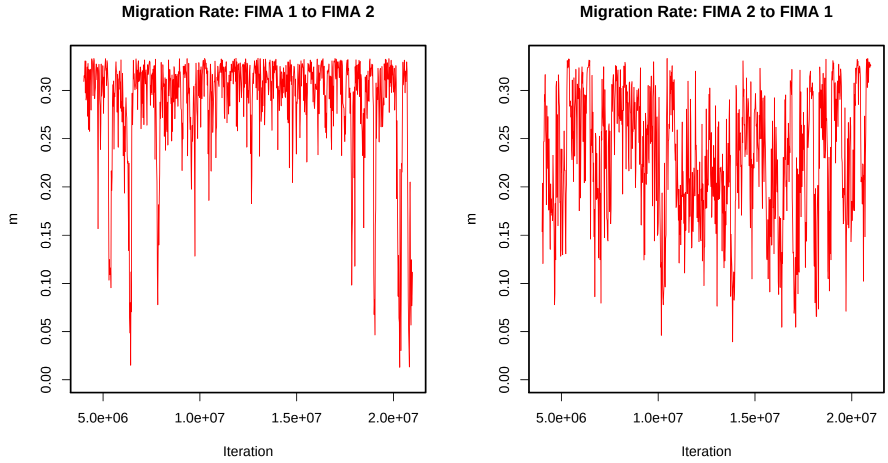

Figure 2: Representative MCMC output of migration rates estimated by BayesAss between two FIMAs connected by a migration rate of 5×10^{-3}, with the samples shown taken every 20,000 iterations and the first 2×10^{6} iterations discarded as burnin.

for it to perform in a more informative manner. One possibility is to implement Metropolis-Coupled MCMC [8], which runs multiple chains of different temperature, thereby allowing more complete exploration of the parameter space.

There is an obvious computational burden to this approach. A single run of 21×10^6 iterations took approximately two hours on a standard desktop computer (Pentium 4 processor). However, since these long chains did not appear to improve mixing, it is probable that much shorter chains of 3×10^6 are adequate [10, 1], particularly in the decision rule framework required here.

It has also been shown that estimation of m may be hindered if the assumption of constant allele frequencies in each population (unchanged by immigration) is violated [7], although whether this is true here was not investigated. However, despite these problems BayesAss has a number of potential advantages. Specifically it is proposed by its authors as a means to estimate asymmetric migration rates on an ecologically relevant timescale. Because such information is invaluable to management it is clear that further investigations are warranted.

Acknowledgements

This work was completed with the invaluable assistance of Karen Martien, Dave Gregovich and Mark Bravington.

References

[1] C. Ciofi, G. Wilson, L. Beheregaray, C. Marquez, J. Gibbs, W. Tapia, H. Snell, A. Caccone, and J. Powell, *Phylogeographic history and gene flow among giant Galápagos tortoises on Southern Isabela island*, Genetics, 172 (2006), pp. 1727–1744.

[2] International Whaling Commission, *The revised management procedure (RMP) for Baleen whales*, Journal of Cetacean Research and Management, Supplement 1 (1999), pp. 251–258.

[3] ——, *Report of the workshop to design simulation based performance tests for evaluating methods used to infer population structure from genetic data*, Journal of Cetacean Research and Mangement, Supplement 6 (2004), pp. 469–485.

[4] ——, *Report of the 2nd TOSSM (Testing Of Spatial Structure Models) workshop.* 2006.

[5] J.-M. Cornuet, S. Piry, G. Luikart, A. Estoup, and M. Solignac, *New methods employing multilocus genotypes to select or exclude populations as origins of individuals*, Genetics, 153 (1999), pp. 1989–2000.

[6] K. Dawson and K. Belkhir, *A Bayesian approach to the identification of panmitic populations and the assignment of individuals*, Genetic Research, 78 (2001), pp. 59–77.

[7] P. Faubert, R. Waples, and O. Gaggiotti, *Evaluating the performance of a multilocus Bayesian method for the estimation of migraion rates*, Molecular Ecology, 16 (2007), pp. 1149–1166.

[8] C. J. Geyer, *Markov chain monte darlo maximum likelihood*, in Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, E. M. Keramidas, ed., Fairfax Station, Virginia, 1991, Interface foundation, pp. 156–163.

[9] D. Grez, J. Cope, A. E. Punt, and J. Brandon, *Application of a sequential hypothesis testing method to example tossm datasets*, Scientific Report SC/58/SD1, International Whaling Commission, 2006.

[10] B. Gum, R. Gross, and R. Kuehn, *Discriminating the impact of recent human mediated stock transfer from historical gene flow on genetic structure of european grayling* Thymallus thymallus *L.*, Journal of Fish Biology, 69 (Supplement C) (2996), pp. 115–135.

[11] J. Pritchard, M. Stephens, and P. Donnelly, *Inference of population structure using multilocus genotype data*, Genetics, 155 (2000), pp. 945–959.

[12] B. Rannala and J. Mountain, *Detecting immigration by using multilocus genotypes*, Proceedings of the National Academy of Science USA, 94 (1997), pp. 9197–9201.

[13] A. Strand, *Metasim 1.0: an individual based environment for simulating population genetics of complex population dynamics*, Molecular Ecology Notes, 2 (2002), pp. 373–376.

[14] G. Wilson and B. Rannala, *Bayesian inference of recent migration rates using multilocus genotypes*, Genetics, 163 (2003), pp. 1177–1191.

Appendix

Boundary Setting Algorithm

```
bayesAss.BSA <- function(gs, ea, vea, C, critical.m=0.1,
n.samples.per.fima=50,n.fimas=2,n.microsat.loci=30,sample.year=5)
{
  #Extract data from rmetasim output into correct format for BayesAss
  baInput=c()
  for (FIMA in 1:n.fimas)
  {
    for (ind in 1:n.samples.per.fima)
    {
      for (loci in 2:(n.microsat.loci+1)) #1st column is mtDNA seq
      {
     baInput=c(baInput,ind+(FIMA-1)*n.samples.per.fima,FIMA,loci,
     gs[[sample.year]][[FIMA]][,,1][ind,loci],
     gs[[sample.year]][[FIMA]][,,2][ind,loci])
      }
    }
  }
  baInput=data.frame(matrix(baInput,
  nrow=n.fimas*n.samples.per.fima*n.microsat.loci,ncol=5,byrow=TRUE))
  names(baInput)=c("ind","FIMA","loci","a1","a2")
  #write data file
  write.table(baInput,file="baInput.txt",sep="\t",
  row.names=FALSE,col.names=FALSE)
  #call BayesAss
  baRun(n.f=n.fimas)
  #read in migration matrix
  baOut <- read.table("baOut.txt")
  #create starting list with each FIMA assigned to
  #its own management unit
  numpops=length(baOut[,1])
  m.units=list()
  for (i in 1:numpops)
    m.units[[i]]=i
  #to obtain management units first group each FIMA with
  #other FIMAs for which m >= crtical.m
  for (i in 1:numpops)
    for (j in 1:numpops)
      if(i!=j && baOut[i,j]>=critical.m)
        m.units[[i]]=c(m.units[[i]],j)
  #then combine vectors within m.units if FIMAs are shared
   for (i in 1:(length(m.units)-1))
      for (j in 1:length(m.units[[i]]))
          for (p in (i+1):length(m.units))
            for (q in 1:length(m.units[[p]]))
            {
              try(if(m.units[[i]][j]==m.units[[p]][q])
                  {
                    m.units[[i]]=c(m.units[[i]],m.units[[p]])
                    m.units[[p]]=NA
                  },silent=TRUE)
            }
   #finally remove duplicates from each vector within m.units
    for (i in 1:length(m.units))
      m.units[[i]]=unique(m.units[[i]])
   #and trim any NA's from the list
    if(any(is.na(m.units)))
     m.units=m.units[-c(which(is.na(m.units)==TRUE))]
  return(m.units)
}
```

Function to call BayesAss

```
baRun <- function(inputFile="baInput.txt",n.f)
{
  file.rename(inputFile,"infile.txt")
  start.time=proc.time()
  shell("bayesass+.exe")
  end.time=proc.time()
  bayesAss.Lk.output <- read.table("output.txt",header=T)
  column=3
  baOut=matrix(rep(0,n.f*n.f),nrow=n.f,ncol=n.f)
  for (i in 1:n.f)
    for (j in 1:n.f)
    {
```

```
    column=column+1
    print(paste("Migration rate from",i,"to",j,":",
    mean(bayesAss.Lk.output[,column][-c(1:1001)])))
    baOut[i,j]=mean(bayesAss.Lk.output[,column][-c(1:1001)])
    }
  print(paste("BayesAss run time:",(end.time[3]-start.time[3])/60,"mins"))
  write.table(baOut,file="baOut.txt",sep="\t",row.names=FALSE,
  col.names=FALSE)
}
```

Function to obtain depletion and catch statistics from *run.tossm* output

```
tossm.stats<-function(tossm.trial,start.RMP=6,
end.RMP=length(tossm.trial$abund.b[1,]))
{
 n.bps=length(tossm.trial$abund.b[,1])
 n.years=length(tossm.trial$abund.b[1,])
 depletion=matrix(rep(0,n.bps*n.years),nrow=n.bps,ncol=n.years)
 lowest.depletion=c(rep(0,n.bps))
 final.depletion=c(rep(0,n.bps))
 total.catch=c(rep(0,n.bps))
 sum.catches=0
 for (bp in 1:n.bps)
   for (year in 1:n.years)
     depletion[bp,year]=
     (tossm.trial$abund.b[bp,year]/tossm.trial$abund.b[bp,1])*100
 for (bp in 1:n.bps)
 {
   lowest.depletion[bp]=min(depletion[bp,])
   final.depletion[bp]=depletion[bp,n.years]
   for (year in start.RMP:end.RMP)
     total.catch[bp]=total.catch[bp]+tossm.trial$catches[bp,year]
   sum.catches=sum.catches+total.catch[bp]
 }
 for (bp in 1:n.bps)
 {
  print(sprintf("%s %i","Breeding population",bp))
  print(sprintf("%s %.2f","lowest.depletion:",lowest.depletion[bp]))
```

```
   print(sprintf("%s %.2f","final.depletion:",final.depletion[bp]))
   print(sprintf("%s %i","total.catch:",total.catch[bp]))
   print("")
  }
  print(sprintf("%s %i","Sum of catches:",sum.catches))
}
```