Progress in the development and validation of a genetic model for use in the Testing of Spatial Structure Methods (TOSSM) project

David Tallmon[1], Karen Martien[2], and Ralph Tiedemann[3]

[1]Université Joseph Fourier, Grenoble, France.
[2]Southwest Fisheries Science Center, La Jolla, CA, USA.
[3]University of Potsdam, Potsdam, Germany.

Introduction

The International Whaling Commission has recognized the potential usefulness of genetic data to gain insight into whale population structure and management. However, the collection, processing and analysis of genetic data require a large investment of time and resources. In order to maximize the return on that investment, the Testing of Spatial Structure Methods (TOSSM; IWC, 2003) project was started with the purpose of using simulation performance testing to investigate how well genetic data can distinguish among different population structures and to guide decision-making. A particular focus of the project is to evaluate the performance of different genetic analytical methods for defining management units for use under specific management regimes. Thus, TOSSM requires the generation of simulated datasets where the true population structure is known. This report outlines the investigations of a simulation model, METASIM (Strand, 2002), identified for this purpose. METASIM creates a suite of realistic demographic scenarios using populations represented by matrices and tracks the genotypes of individuals cycling through these matrices. This simulation model provides the flexibility of well-known Lefkovich demographic matrix models to determine the transitions of individuals among life history stages, and tracks the genotypes of these individuals, as required to simulate population genetic processes. We used versions of METASIM, called RMETASIM 0.0.7 and 0.0.9, designed for use in the R statistical package (Team, 2003) rather than the stand-alone version of METASIM.

The underlying algorithm of RMETASIM is fairly straightforward (Figure 1). Individuals are the basic unit used in the simulation. Individuals within populations survive, move among life history stages, and reproduce, passing their genotypes on following Mendelian inheritance. It is also possible for individuals of any life history stage (or their gametes) to move among populations. Local populations can be subject to local extinction events. Part A of the Appendix contains more details about the underlying structure of RMETASIM.

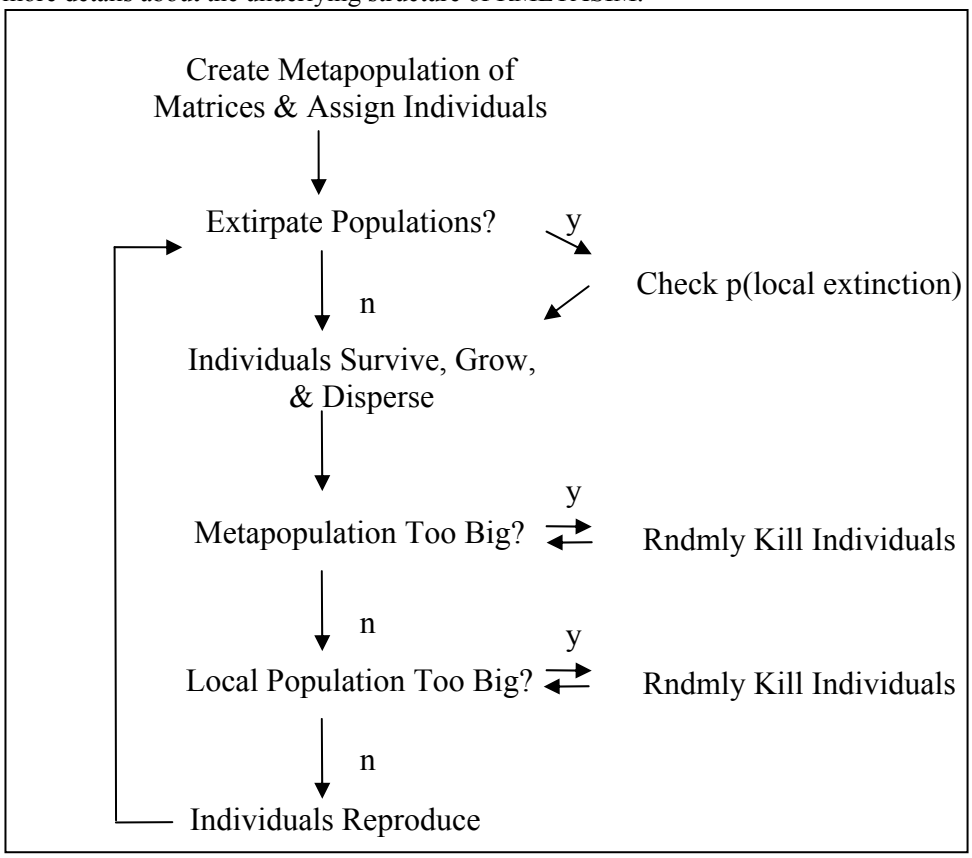

Figure 1: Basic algorithm used in RMETASIM (Strand, 2002) simulations.

The simulations described in this report were used to ensure that RMETASIM matches theoretical expectations from genetic and demographic theory, and to ensure that RMETASIM could be used efficiently to simulate whale demography and genetics. To test the performance of RMETASIM, various combinations of genetic and demographic parameters were used, populations were simulated, and the outputs compared to equations from population genetics and demographic theory. In many of the simulations described below, simple Leslie (age-based) matrices were used to construct isolated populations or populations linked by migration. Initially, we used discrete generation demography, without mutation, because it allowed us to use a vast body of theory developed by Wright and Fisher for comparison to genetic outputs from the model. In contrast, it is much more difficult to obtain expectations for the more complex genetic processes in large matrix models with age structure and overlapping generations. For the initial simulations, the size of each population was maintained using K set at a known value. The effective size (N_e) of a population determines the rate of decay of genetic variation, and setting N_e equal to K allowed us to compare observed model behaviors to theoretical expectations. In subsequent simulations we explored the effects of various mutation models and molecular markers of RMETASIM performance.

To examine the demographic algorithms of RMETASIM, more complex Leftkovich (stage-based) models with overlapping generations were simulated and compared to theoretical expectations. Specifically, we used models of gray whale (*Eschrichtius robustus*) demography to ensure the population growth rates from RMETASIM adequately match outputs predicted by matrix algebra (Caswell, 2001). The demographic and genetic inputs used in many of these simulations are described in some detail in the main body of the report. However, greater details and the exact inputs for a subset of the scenarios modeled are found in the Appendix. Additional justification and description of the vital rates used in the matrices can be found in Martien et al. (SC/56/SD5).

The tasks outlined by the TOSSM workgroup for this project and described in this report are the following:

1) check model outputs against theoretical expectations from population genetic theory in order to validate the biological and genetic algorithms used by RMETASIM.
2) examine the robustness of model predictions to genetic marker type.
3) augment existing model code to include flexible input of initial allele frequencies and genetic marker types and examine the impact of different initial allele frequencies on time to genetic equilibrium.
4) incorporate important aspects of whale population biology and examine their effects on times to demographic and population genetic equilibria and expected equilibrial values.

In most simulations, small population sizes were used to address these tasks because they run more efficiently, yet still provide an adequate test of RMETASIM performance for the tasks described. Fifty replicates were used in all scenarios presented unless otherwise indicated. Not all scenarios simulated were included in this report in order to keep the report and figures concise and easy to read. The mean and variance (where large enough to be visible) for all parameter estimates are shown in each figure. The input files used to assign parameter values for a few of the scenarios presented are in included the Appendix, and input files for all others can be obtained from the authors upon request.

For the most part, RMETASIM is a metapopulation simulation engine. A primary benefit of using RMETASIM as a program within the R statistical framework is that data produced by the RMETASIM simulations can be analyzed seamlessly within R using various contributed statistical packages. For this report, the R package "ape", contributed by an independent researcher for the analysis of molecular genetic data, was used in concert with R scripts written by Allan Strand and the authors to extract genetic summary statistics from the simulations.

Section 1. Comparison of model outputs to theoretical expectations

The validation of the genetic and demographic algorithms used in METASIM was achieved by comparing the rate of change and equilibrial values of common population genetics summary statistics from the model to expectations based upon single- and multiple- population genetic theory. First, the rate of decay of genetic variation within populations and divergence among populations were recorded for known demographic conditions and compared to parametric values. The initial simulations contained no mutation. The genetic properties of the model were examined by first checking the behavior of the model with a single population, and then by investigating the multiple population behavior with and without migration. For all of the scenarios in this section, below diploid and haploid markers were simulated. For most of these scenarios, both sequence and non-sequence molecular markers were simulated independently.

<u>No mutation</u>

The rate of decay of genetic variation in a population is an inverse function of the (effective) population size, N_e. Specifically, the expected amount of heterozygosity (H_t) at any point in time (t) can be calculated from the initial heterozygosity (H_i) and the effective population size (N_e) (Crow and Kimura, 1970), (p. 102).

$$H_t = H_i\left(1-\frac{1}{2N_e}\right)^t \qquad (1)$$

Across a range of population sizes, the observed decay of heterozygosity in the simulations closely matched expectations (Figure 2). Although there is variation among replicates, the expected value falls within the error bars calculated across

replicates and different effective sizes. (See Appendix, Part B, for example input for the decay of genetic diversity in a single population.)

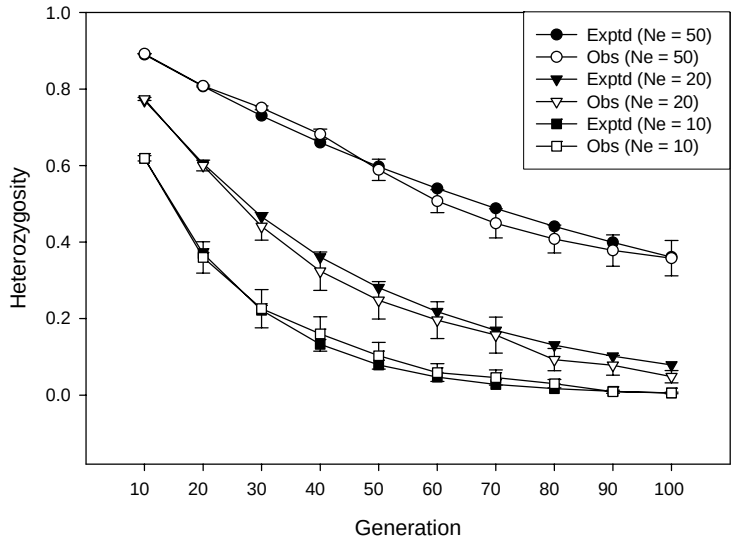

Figure 2: The expected and observed decay in heterozygosity for an isolated population at three effective population sizes (Ne = 50, 20, 10).

Because few species exist as single, unstructured populations, it is of interest to examine the distribution of genetic variation within and among multiple populations. Accordingly, we simulated 5 populations of known sizes and migration rates and observed the genetic divergence among these populations. (See Appendix, Part C, for example input.) Under these controlled conditions it is possible to obtain expected values for several metrics of population divergence. The most commonly used metric of population divergence is Weir and Cockerham's θ, which is a more robust version of Wright's well-known F_{st} in cases where the number of subpopulations is finite (Excoffier, 2001). The easiest multiple-population case is no migration, in which case θ approaches unity over time. The observed values from RMETASIM closely matched the predicted values as θ increased toward its equilibrium value (Figure 3). The combined observations from the rate of increase in θ for the no migrant case, and the decay of heterozygosity in single, finite populations, suggest the model is robust in terms of its Mendelian genetics and the interactions between basic demography and genetics.

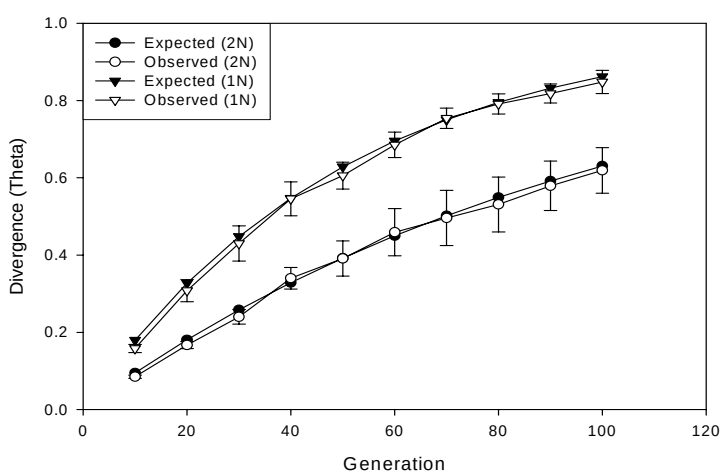

Figure 3: The expected and observed increase in divergence among populations at a diploid (2N) and haploid (1N)locus.

It is also important to assess model behavior with migration among populations, because this is of interest in many biological situations. Again, we used θ as the summary statistic of divergence among n populations of known effective size (N_e) and migration rate (m). For the case of no mutation, θ is given by the equation,

$$\theta=\frac{1-m*\frac{n}{n-1}}{2N_e(1-(1-m*\frac{n}{n-1}))+(1-m*\frac{n}{n-1})} \qquad (2)$$

provided by (Weir, 1996) (p. 183). Figure 4A shows that equilibrium values are reached quickly at small $N_e = 50$, and that RMETASIM approaches the expected values across a range of migration rates. Furthermore, it appears that the effect of population size on divergence follows expectation. Larger populations have slightly lower equilibrium divergence values and reach these values at a slower rate than smaller populations (Figure 4B). A doubling of the population size approximately doubles the time to reach 90% of the expected equilibrium divergence value, as found previously using F_{st} (Allendorf and Phelps, 1981).

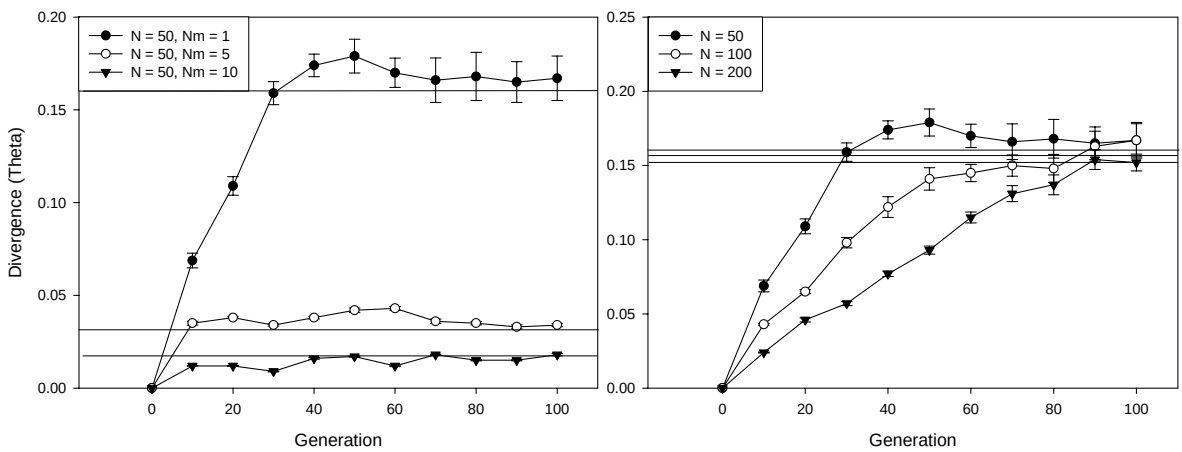

Figure 4: The observed increase in divergence among populations to equilibrium values with a) different numbers of migrants per generation and constant population size N = 50 and b) different sized local populations (N = 50, 100, and 200) exchanging one migrant per generation. Expected values are shown as horizontal lines.

Mutation

We used the single population demography described above to test the robustness of the model when a mutation parameter was added to the simulations. The expected level of genetic variation in a population is a function of N_e and the mutation rate (μ). The number of alleles, heterozygosity, or number of polymorphic DNA bases at a diploid locus can be predicted from the composite parameter Θ, which is the product of these two values and a constant (Nordberg, 2001).

$$\Theta=4N_e\,\mu \qquad (3)$$

We used this relationship to test whether the observed mean number of alleles, heterozygosity, and number of segregating sites (for sequence loci) produced by populations of known N_e and μ matched the expected value for Θ from this equation. Figure 5A shows theta as a function of time for a diploid locus with discrete (non-sequence) allelic states (such as microsatellites) determined by an infinite allele model for two values of μ. The curve that shows decrease in Θ depicts the decrease in genetic variation in a population with unique alleles assigned to all individuals of the 1st generation, as mutation-drift equilibrium is reached. The two curves depicting an increase in Θ depict populations initialised with 1 allele that increased to their equilibrium values as mutation added genetic variation over the course of the simulations. Similarly, Figure 5B shows the increase to the expected value of Θ for a haploid sequence of 150 bases in a population initialised with no genetic variation. For the diploid and haploid loci, the R package "ape" was used to calculate Θ from the number of alleles and segregating sites, respectively.

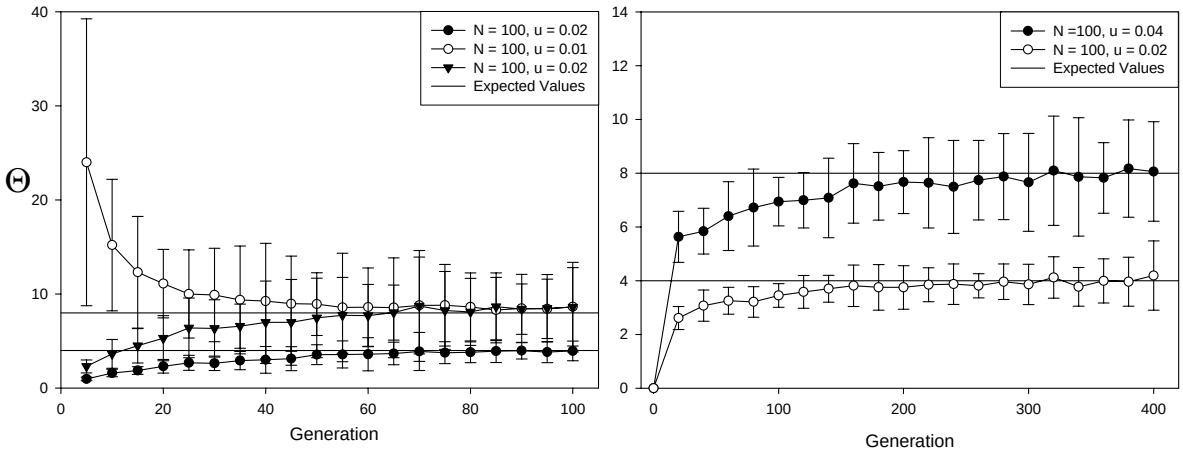

Figure 5: The observed change in theta to its equilibrium value at a) a diploid locus and b) a haploid sequence. Expected equilibrium values are shown as horizontal lines.

Section 2. Robustness of model predictions to marker type

There is a great deal of interest in the use of both bi-parentally and uniparentally inherited genetic markers for TOSSM because these markers often provide different and complementary insights into population structure. Consequently, we examined the performance of RMETASIM using both nuclear, bi-parentally inherited diploid markers and maternally inherited, haploid markers under many of the conditions described above. The results of simulations using haploid markers are shown in Figures 3 and 5. The observed asymptotic decay in haploid heterozygosity and increase in divergence meet the expectations from population genetic theory, calculated using equation 1 modified for haploid, maternal-inheritance; the same was true for the diploid markers.

The increasing availability of sequence data makes it important to examine how RMETASIM performs using this data type. To do this we re-ran many of the population simulation conditions described above using haploid and diploid sequences and compared them to the previous results from non-sequence data. Figure 6 provides one example in which there is no detectable departure from the expected rate of decay of mtDNA heterozygosity (allelic diversity). The result is consistent whether sequence or discrete locus data are considered. In all cases simulated, the sequence and discrete locus results were indistinguishable, suggesting that RMETASIM provides robust output whether sequence or non-sequence markers are used.

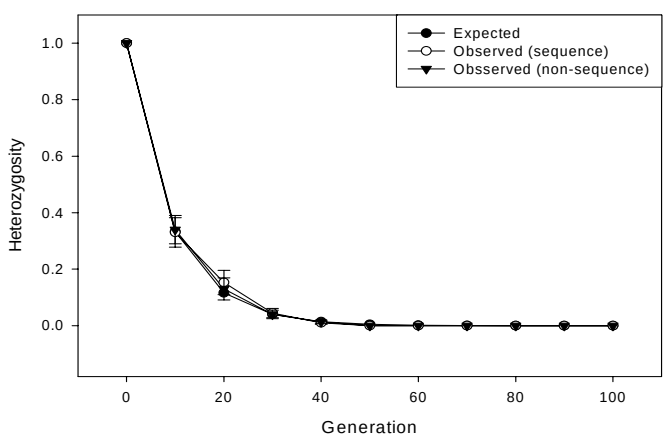

Figure 6: The observed and expected decay in heterozygosity for a haploid sequence or non-sequence locus ($N_e = 10$).

Similarly, the accumulation of genetic variation appears to follow the expectation for haploid, maternally inherited DNA sequences. Figure 5B depicts the increase to equilibrium for the composite parameter Θ (described in the previous

section), calculated from the observed number of segregating sites and compared to the parametric value. The cumulative results of all simulations suggest RMETASIM meets theoretical expectations for both haploid and diploid sequence and non-sequence markers, and is suitable for simulating the effects of population dynamics on commonly used molecular markers.

Section 3. Flexible input of initial allele frequencies and marker types

A recurrent issue in simulation modelling is determining the best means to initialize the genotypes and allele frequencies used in simulations. Simulation results should be robust to initial conditions in order to provide confidence that the results can be inferred to natural populations, where historical conditions are unknown. The current version of RMETASIM allows the user to stipulate the type of marker locus, including ploidy, transmission pattern, and mutation model. In addition, one can stipulate the initial number of alleles at each locus and their frequencies. We examined the convergence of RMETASIM to equilibrium values using a range of initial conditions.

Effects of Initial Allele Frequencies

The robustness of RMETASIM to the use of initial bi-allelic or infinite allele models can be seen in the figures showing decay of heterozygosity and divergence among populations. For the simulations of the decay of heterozygosity (Figures 2 and 6) each individual in the first generation was assigned one haplotype (haploid case) or two unique alleles (diploid case). In contrast, for the simulations of diverging populations, linked or unlinked by migration, only two alleles were assigned to all populations and these frequencies changed over time due to genetic drift (Figure 3, 4, and 5).

In simulations with 5 subpopulations with or without migration, individuals were assigned genotypes from a pool of 10, 100, or 1000 initial alleles. The number of alleles used to initialize individual genotypes did not affect the rate of approach to equilibrium values (results not shown). Rather, the demographic conditions determined the equilibrium values and were of predominant influence on the number of generations necessary to reach those values. This has been seen in previous modelling efforts (Allendorf and Phelps, 1981) (Varvio et al., 1986). However, there was a slight increase in the amount of real time required to run simulations with more initial alleles. This suggests that little is gained, and simulation efficiency is decreased, by using extremely high levels of genetic variation to start simulations. However, it is also clear that RMETASIM meets theoretical expectations regardless of initial allele conditions or molecular marker type.

RMETASIM is equipped with a flexible means of choosing the number of alleles to initialize the genotypes. An alternative means of initialising the genotypes is to use values drawn from the coalescent to determine allele frequencies. The coalescent provides a means of initialising initial allele frequencies that is robust to population history and computationally efficient, which could save vast amounts of TOSSM simulation time. We are currently developing scripts to employ the coalescent program SIMCOAL (v. 1.0, (Excoffier, 2000)) to create allele frequency distributions, which will then be used within RMETASIM to initialize individual genotypes.

Section 4. Incorporation of whale demography

It is apparent from the previous sections that RMETASIM has demographic and genetic algorithms that match theoretical expectations quite well. However, the use of RMETASIM for TOSSM is justifiable only if RMETASIM adequately incorporates the necessary subtleties of whale demography and can produce data sets of similar size and detail to empirical ones. Consequently, we tested the ability of RMETASIM to accurately incorporate whale demography, to produce data sets similar to empirical data sets for baleen whales, and the relative influence of some important parameters on the speed of the simulations.

Population Growth

Simulations were conducted using a more advanced Leftkovich matrix model designed by Martien et al. (SC/56/SD5) to closely approximate gray whale (*Eschrichtius robustus*) demography. Martien et al. present two matrices, one of which describes life history parameters near zero population density (Table 1A, henceforth referred to as the 'zero population density matrix'), the other of which describes the parameters near carrying capacity (Table 1B, the 'carrying capacity matrix'). We ran simulations using each of these matrices separately to ensure that the demographic behavior of RMETASIM conformed to expectations from matrix model theory. We started the population at a stable age distribution far below K and compared the observed growth rate obtained for each of the life history matrices to that expected based on the dominant eigenvalue of matrix (Caswell, 2001).

Table 1. Life history matrices from Martien et al. (SC/56/SD5)

A. Zero population density

	Juve1	juve2	fert	lact	male
juve1	0.797	0	0	1.0	0
juve2	0.123	0.718	0	0	0
fert	0	0.101	0.648	0.946	0
lact	0	0	0.300	0	0
male	0	0.101	0	0	0.954

B. Carrying capacity

	Juve1	juve2	fert	Lact	male
juve1	0.730	0	0	1.0	0
juve2	0.210	0	0	0	0
fert	0	0.47	0	0.946	0
lact	0	0	0.946	0	0
male	0	0.47	0	0	0.954

When we used the carrying capacity matrix (lambda = 1.00), the observed growth rate matched the deterministic values obtained from theory. However, when we used the zero population density matrix (lambda = 1.07), we uncovered model behavior that did not match expectations based on standard matrix model theory for an expanding population. Subsequent investigations by the authors of this report and RMETASIM's primary author revealed incorrect demographic algorithms in RMETASIM that caused the observed population growth rates to exceed the expected values. These problems stem from the way in which the reproduction and survival matrix elements are combined and multiplied by the stage vector in RMETASIM. Currently, RMETASIM is being revised by its author to address these problems, and a fully-functional version that matches standard matrix model expectations should be available in the next few weeks. The problems with the demographic algorithms in RMETASIM did not affect the other simulations described in this report.

Mutation Rates and Genetic Variation

We also examined genetic summary statistics produced by RMETASIM with different mutation rates, population sizes, and the aforementioned gray whale demographic matrix with positive growth rate ($\lambda = 1.07$) (Martien et al., SC/56/SD5). We then compared the model results to an eastern gray whale genetic dataset kindly provided by Richard LeDuc. The dataset included mtDNA control region sequences (an expanded version of the dataset described in (Leduc et al., 2002)) and data from 7 microsatellite loci. The motivation for this comparison was to ensure that RMETASIM would produce summary statistics similar to observed data sets using population sizes corresponding to reasonable values for the eastern gray whale population, and realistic mutation rates obtained from empirical data. In addition, if the results are reasonable, we can use the best mutation rate estimates to help narrow the parameter space used in subsequent simulations for TOSSM.

For these simulations, the population was initialized at N=K, so that the positive population growth rate would keep the population near K for the entire simulation. Mutation rate estimates for microsatellite and mtDNA sequence data were obtained from the human genetics literature, and used to obtain a high and low estimate for each marker type to bracket the likely values observed in real whale data sets . The high and low microsatellite mutation rates used were 2×10^{-3} and 2×10^{-4}/locus/generation, respectively (Kayser et al., 2000; Zhivotovsky et al., 2004). The mtDNA mutation rates we used were 5×10^{-3} and 5×10^{-5}/locus/generation, based on estimates from the control region (Heyer et al., 2001; Sigurdardottir et al., 2000). Population size estimates for the eastern gray whale stock range from ~16,000 to 27,000 (Rugh et al., 1999; Punt, pers. Comm.); so we used values of 15,000, 20,000, and 30,000.

These parameter estimates were used in simulations of whale population dynamics for 10,000 time steps. Each set of conditions was simulated 10 times, and at each time step a sample of 100 individuals was drawn from the population. The observed distributions of genetic summary statistics for diploid 8 microsatellite loci and a 500 base sequence (locus) of mtDNA were compared to empirical data sets for the eastern gray whale. An example input file for these simulations is contained in the Appendix, Part D.

Leduc et al. (2002) sequenced the mtDNA control region of a sample from the eastern gray whale population and found high haplotype diversity (hd = 0.95) and 33 haplotypes, each with a frequency less than 11% (Leduc et al., 2002). We calculated the mean expected heterozygosity of 0.78 from gray whale dataset of 7 microsatellite loci. These empirical data can be compared to the observed values from the simulations in Figures 7 and 8.

Figure 7 shows that the population sizes simulated have relatively little effect on the equilibrium amounts of genetic variation. Although the populations can be expected to reach different levels of genetic variation, their behaviour is similar over the time frame shown here. In contrast to the limited effect of population size, the mutation rates, which vary over 1-2 orders of magnitude, greatly impact the equilibrium levels of genetic variation. Only high mutation rates, in combination with the population sizes examined, bring the simulated levels of genetic variation close to that observed in the real sample.

Figure 8 shows the distributions of mtDNA haplotypes at the end of the simulations for a population of moderate size (N = 20,000), for both the high and low mtDNA mutation rates. The distribution for the high mutation rate is right skewed, and shows many haplotypes that occur in only one or a few copies in each sample. This distribution is similar to the empirical data set. In contrast, the distribution for the low mutation rate includes one very common allele and a rare allele, which demonstrates that the low mutation rate is unlikely to generate the observed distribution of haplotypes. Although the data are not shown, the general distribution of haplotypes was relatively insensitive to the population sizes simulated

(N=15,000, 20,000, 30,000). That is, all high mutation rate simulations produced highly diverse, skewed haplotype distributions and all low mutation rate simulations produced one very common haplotype with one or two uncommon ones.

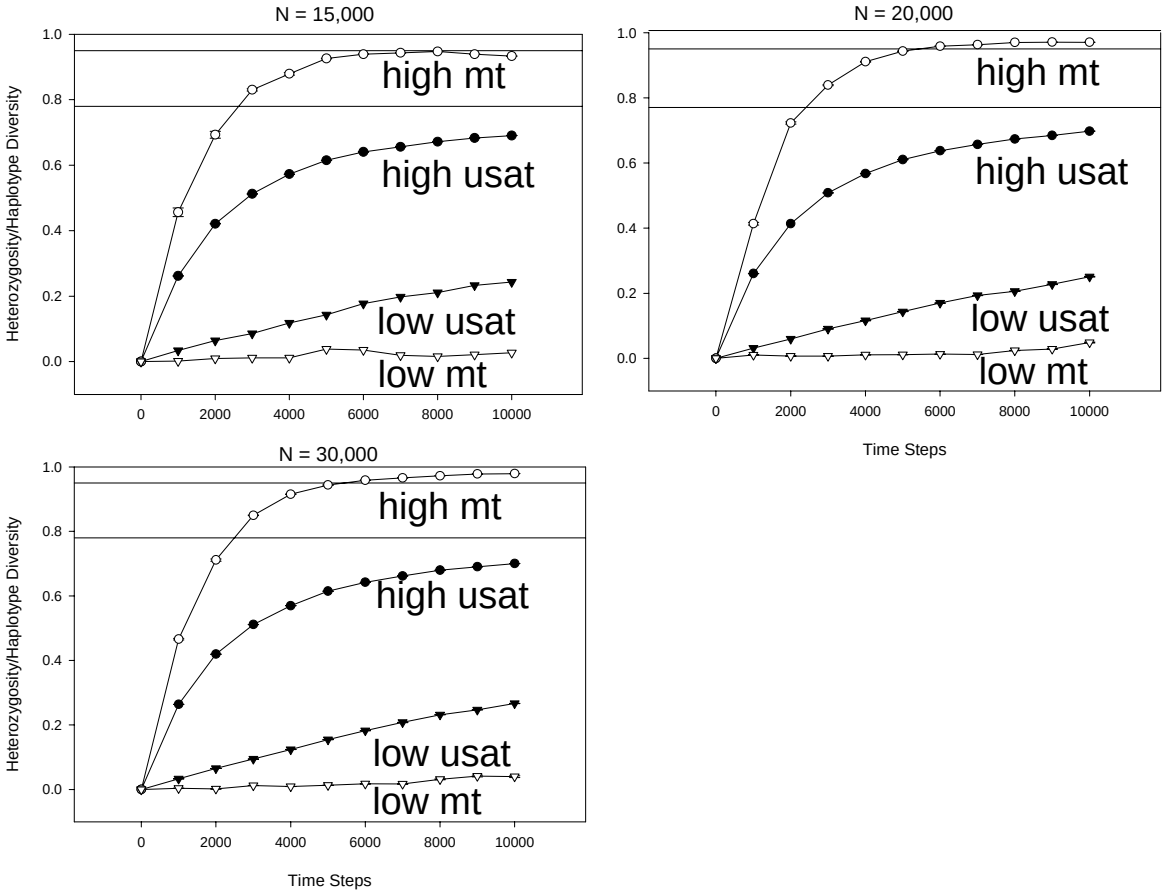

Figure 7: Expected microsatellite (usat) heterozygosities and mtDNA (mt) haplotype diversities for samples (n = 100) drawn from simulated whale populations with high and low mutation rates and N = 15,000 (A), 20,000 (B), or 30,000 (C) individuals. Horizontal lines provide empirical values from mtDNA (upper) and microsatellite (lower) data sets.

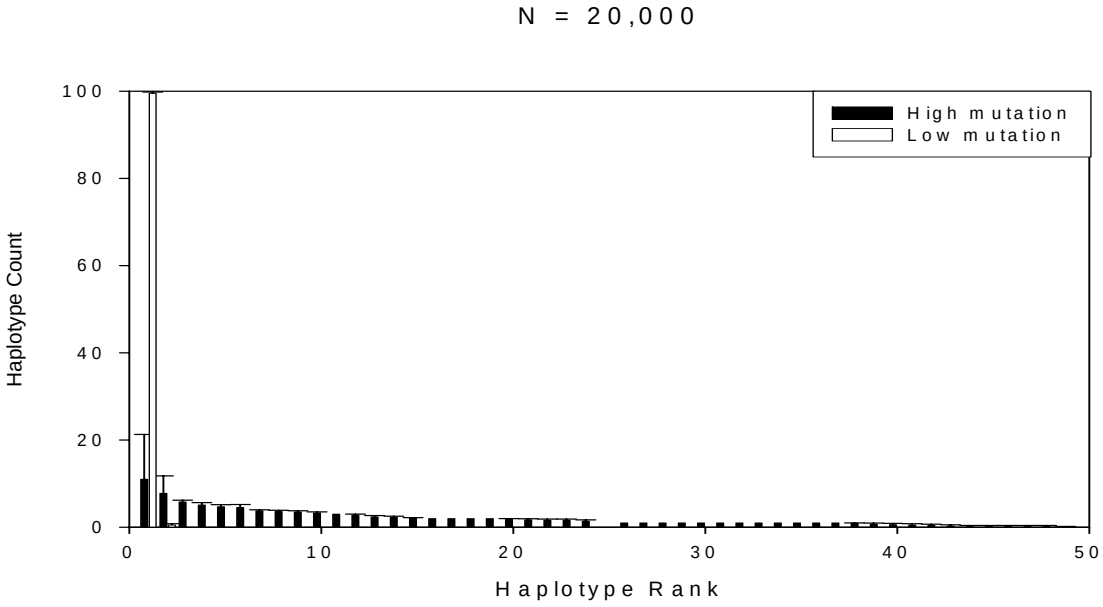

Figure 8: The frequency distribution of mtDNA haplotypes at the end of 10,000 time steps for a population. Haplotypes are ordered from most to least common in each sample (n = 100).

The simulated whale populations produced a wide range of heterozygosities, haplotype diversities, and haplotype distributions, independent of the population size. From these simulations and figures, it would appear that the higher end of the mutation rate estimates are better approximations to what is seen in whale data sets, assuming whale population effective sizes are not many orders of magnitude larger than those approximated by the population size used in this simulation. Such a substantial deviation is highly unlikely. Because these simulations were conducted using demographic parameters from eastern gray whales, the ratio of effective population size to census size (Ne/N) from the simulations should be similar to that for the real population. Deviations from the simulated ratio could be caused by a mating system that is more skewed than that simulated, or if the long-term harmonic mean abundance of the real population is significantly reduced due to the bottleneck imposed by historic whaling. However, both of these sources of bias would cause the ratio of Ne/N in the simulations to be positively biased, which would result in an underestimate of the mutation rate. Thus, it seems safe to conclude that the actual microsatellite and mtDNA mutation rates in eastern gray whales are near the upper end of the ranges we examined.

Demographic and Genetic Effects on Simulation Times

Finally, we examined the relative and absolute time required to run simulations of different levels of complexity (Table 2). This was done to provide insight into which parameters most affect the speed of the RMETASIM simulations, and what sorts of scenarios may be compared in a realistic time frame given the speed and number of computers available for TOSSM applications.

The type of molecular marker (discrete or sequence) has very little effect on the speed of simulations. And, for sequence markers, the length of the sequence has almost no effect. The number of diploid loci simulated had detectable effects on simulation rate. However, an increase in the number of loci from 1 to 10 less than doubled the simulation time, so the effect of number of loci is unlikely to be great unless many tens of markers are used.

Table 2: Absolute time (secs) required to run 10 replicates of whale demography simulations with varying numbers of populations, individuals/population (Popsize), and unlinked loci or sequence lengths. Each iteration was run for 100 time steps. All simulations were run on a 1.2 GHz laptop with 512 MB Ram and Windows XP OS.

1 Population					
Popsize	# 2N Loci			1N Sequence Length	
	1	10	20	100	500
100	0.40	0.76	1.11	0.39	0.38
1000	4.26	8.10	12.76	4.00	3.89
10000	61.73	117.37	158.35	57.57	59.10
5 Populations					
Popsize	# 2N Loci			1N Sequence Length	
N	1	10	20	100	500
100	2.81	4.29	5.95	2.59	2.63
1000	32.01	53.35	74.83	30.46	31.05
10000	386.02	662.82	1004.70	394.33	423.87

Diploid loci followed the infinite mutation model (rate = 0.001) and initialized with 2 alleles/locus.
Haploid loci were initialized with 2 alleles/locus (rate = 0.005).
Mitochondria DNA sequences were simulated as unlinked loci following a Jukes-Cantor mutation model (rate = 0.005) and initialized with 2 haplotypes/locus.

It is apparent that the greatest effect upon simulation speed is, by far, the number of individuals simulated. This makes sense because the model is individual-based, which requires that detailed demographic and genetic records be kept for each individual simulated. A 10-fold increase in the number of individuals slowed the rate of simulation more than 10-fold. This suggests that efforts to decrease the number of individuals simulated will provide the greatest return in efficiency. It took approximately 1 second to run a simulation of 5 populations of 10,000 individuals each with 20 loci for one time step on a reasonably fast laptop computer (see Table 2 for specs). Any increased efficiency by reducing the number of time steps could have a large cumulative impact across many scenarios.

The number of time steps required to reach equilibrium values of population divergence depends upon an array of parameter combinations, including population size, structure, number of populations, migration rates, and mutation rates. For large populations (10,000 or greater) it has been shown that thousands of generations are required if only 1 migrant is exchanged among populations per generation (Varvio et al., 1986); Tables 2 and 3). If migration rates are considerably higher, the number of generations to equilibrium decreases rapidly. Reaching within population mutation/migration/drift

equilibrium can take similarly large numbers of generations. Figure 7 shows the rate of increase to equilibrium levels of genetic variation for a single population with high or low migration rates can take thousands of generations, and, for a given population size, more time is required for a lower mutation rate. Given the time requirements shown in Table 2, simulating some of the individual scenarios of interest to TOSSM will require several hours to complete, even with low levels of replication (< 100 iterations).

Summary

The exercises described in this report have shown RMETASIM to be a well-functioning model that will be adequate to meet the needs of the TOSSM project. The population genetic functions of the model accurately matched theoretical expectations for haploid and diploid loci across an array of scenarios that included isolation, migration, no mutation, and mutation. The structure of the model proved flexible enough to allow the incorporation of a complex whale demography that included a realistic age at first reproduction and interbirth interval (see Martien et al., SC/56/SD5 for details). With the exception of the demography error noted in section 4, which should be corrected by the time of the Sorrento meeting, the output of the model met theoretical expectations from both demographic and genetic theory across a suite of conditions.

A few tasks remain before the model will be ready to generate simulated datasets. First, the coding error described in section 4 must be fixed. Allan Strand, the original author of RMETASIM, is correcting this mistake and expects to have a corrected version available by the end of June, 2004. Second, we need to implement density dependence in the model. This will be achieved by interpolating between two sets of demographic parameters, those expected at zero population density and those expected at carrying capacity, as described in Martien et al. (SC/56/SD5). This approach will allow density dependence to act on any life history stage. The algorithm has already been coded and will be implemented and debugged as soon as a corrected model is received from Allan Strand.

The third remaining task is the incorporation of a coalescent model for initialising haplotype and allele frequencies. We are currently in the process of writing the necessary scripts to integrate the coalescent program SIMCOAL (Excoffier, 2000) RMETASIM. We expect this task will be completed by mid-July.

Acknowledgements

Thanks to Allan Strand for generously providing programming advice, providing modified versions of RMETASIM, and for help with R scripts. We would also like to thank Rick LeDuc, Aimee Lang and David Weller for providing us with empirical datasets for use in investing appropriate mutation rates for the mode. This project was funded by the National Marine Fisheries Service and the International Whaling Commission.

Literature Cited

Allendorf FW, Phelps SR, 1981. Use of allelic frequencies to describe population structure. Canadian Journal of Fisheries and Aquatic Sciences 38:1507-1514.

Caswell H, 2001. Matrix population models, 2nd ed. Sunderland, MA: Sinauer Associates, Inc.

Crow JF, Kimura M, 1970. An introduction to population genetics theory. New York: Harper & Row.

Excoffier L, 2000. Simcoal: A general coalescent program for the simulation of molecular data in interconnected populations with arbitrary demography. In.

Excoffier L, 2001. Analysis of population subdivision. In: Handbook of Statistical Genetics (Balding DJ, Bishop M, Cannings C, eds). New York: Wiley & Sons, Ltd.; 271-307.

Heyer E, Zietkiewicz E, Rochowski A, Yotova V, Puymirat J, Lubuda D, 2001. Phylogenetic and familial estimates of mitochondrial substitution rates: study of control region mutations in deep-rooting pedigrees. American Journal of Human Genetics 69:1113-1126.

International Whaling Commission. 2003. Report of the Workshop to Design Simulation-based Performance Test for Evaluating Methods Used to Infer Population Structure from Genetic Data, 21-24 January 2003, La Jolla, USA. Paper SC/55/Rep 3 presented to the International Whaling Commission Scientific Committee, May 2003, Berlin. 23pp.

Kayser M, Roewer L, Hedman M, Henke L, Henke J, Braur S, Kruger C, Krawczak M, Nagy M, Dobosz T, Szibor R, Knijff Pd, Stoneking M, Sajantila A, 2000. Characteristics and frequency of germline mutations at microsatellite loci from human Y-chromosome, as revealed by direct observation in father/son pairs. American Journal of Human Genetics 66:1580-1588.

Leduc RG, Weller DW, Hyde J, Burdin AM, Rosel PE, R. L. Borwnell J, Wursig B, Dizon AE, 2002. Genetic differences between western and eastern gray whales (*Eschrichtius robustus*). Journal of Cetacean Research and Management 4:1-4.

Martien, KK, Tallmon DA, and Tiedemann R. Life history matrices for the TOSSM model. Paper SC/56/SD5 submitted to the Scientific Committee of the International Whaling Commission. Sorrento, Italy, June 2004.

Nordberg M, 2001. Coalescent Theory. In: Handbook of Statistical Genetics (Balding DJ, Bishop M, Cannings C, eds). Chichester: Wiley & Sons; 179-212.

Rugh, D.J., Muto, M.M., Moore, S.E., DeMaster, D.P. 1999. Status review of the eastern North Pacific stock of gray whales. NOAA Technical Memorandum NMFS-AFSC-103. 96pp. Available from NMML,7600 Sand Point Way NE, Seattle, WA 98115.

Sigurdardottir S, Helgason A, Gulcher JR, Stefansson K, Donnelly P, 2000. The mutation rate in the human mtDNA control region. American Journal of Human Genetics 66:1599-1609.

Strand A, 2002. METASIM 1.0: an individual-based environment for simulating population genetics of complex population dynamics. Molecular Ecology Notes 2:373-376.

Team RDC, 2003. R: A language and environment for statistical computing. In. Vienna, Austria: R Foundation for Statistical Computing.

Varvio S-L, Chakraborty R, Nei M, 1986. Genetic variation in subdivided populations and conservation genetics. Heredity 57:189-198.

Weir BS, 1996. Genetic Data Analysis II. Sunderland, MA: Sinauer Associates, Inc.

Zhivotovsky LA, Underhill PA, Cinnioglu C, Keyser M, Morar B, Kivisild T, others a, 2004. The effective mutation rate at Y chromosome short tandem repeats, with application to human population-divergence time. American Journal of Human Genetics 74:000-000.

Appendix:

Part A. Details of RMETASIM matrices.

RMETASIM can be thought of as a matrix of matrices that handle within and between population demography. Collectively these matrices form a metapopulation. Within-population demography consists of survival, male and female reproduction. Consequently, each population's demography is handled by three matrices, one for each component of demography. Among-population demographic events are handled by another set of matrices that determine the migration of individuals and gametes between populations. Many kinds of survival, reproduction strategies, and migration patterns can be accommodated in the metapopulation and these can vary among all populations in the metapopulation. It is important to note, however, that while the demographic parameters are entered and stored in the form of a matrix, METASIM does not rely on matrix projections. Rather, METASIM is an individual-based model (IBM) in which the fate of each individual is governed by the transition probabilities specified in the demographic matrices. Essentially, the demographic options are limited only by the constraints imposed by the matrix model structure. The basic algorithm used in RMETASIM is described in Figure 1.

For many of the simulations described in the main body of the report, the matrices and parameter values used to create the RMETASIM populations and metapopulation(s) are described in this Appendix. (This may be more detail than most readers require, but useful to those planning to do simulations of their own.) It should be possible to mimic, allowing for stochastic simulation among iterations, the results presented above by using RMETASIM with the following input files. Additional help files for running the programming and performing some genetic analyses can be obtained by downloading and installing the RMETASIM library.

One of important aspect of the current version of RMETASIM is that density dependence is imposed as a ceiling, K, on population size, N. For the initial simulations we sought to control the size of the adult stage, in order to control the effective size of each population. Therefore, it was necessary to split each population into a juvenile and adult population and then have all juveniles "migrate" into the adult population. As a consequence, each "population" in a demographic sense, consists of a juvenile matrix and adult matrix, each with a population size controlled by its K. This allowed us to tightly control the number of reproductive individuals, and thus ensure that the model meets the expectations from theory that hinge upon knowing the number of reproductive individuals. In other words, as a result of these manipulations, we were able to use common population genetic theory to test the model outputs. As shown below, each population contains two stages (two rows and columns) and each biological population (four rows and columns) is a coupled juvenile population and adult population. The example input shown below looks like the following in RMETASIM matrix form, with movement of individuals from columns to rows in each time step:

	juve1	juve2	adult1	adult2
juve1	0	0	0	**50**
juve2	0	0	0	0
adult1	0	0	0	0
adult2	**1**	0	0	**0**

The upper left quadrant of the matrix represents transitions within the juvenile population and the lower right represents transitions within the adult population. The other two quadrants represent movement between the two populations. In this example, all individuals in the juvenile population survive and move to the adult population with prob(survival+movement) = **1,** as shown in lower left corner of 4x4 matrix. In the adult population, each adult produces 50 offspring, all of which all enter the juvenile population (**50** in upper right corner), and then dies (as shown by prob(survival) = **0** in lower right corner). Under these conditions, there is no survival within the juvenile population or adult population; rather they simply pass individuals between them to form one biological population. By maintaining an overabundance of juveniles (K = 500) moving into a smaller adult population (K= 10), we could ensure that the effective size remained constant each generation because following each time step the adult population had to be culled from 500 back to its carrying capacity of 10. We examined the distribution of genetic variation in the adult population(s) only.

(Note: I use stage throughout this appendix because stage-based models were used to incorporate whale demography in more advanced models, but age is technically the correct term in the present context.)

Part B. The following scripts can be used to simulate a single population 50 times and observe the decay in heterozygosity. The "#" symbol is used to comment the code without interrupting it, should the reader wish to simply cut and paste the following text into R (assuming the RMETASIM library has been installed). Note the pathway to the output file for this and all subsequent examples is currently c:/ outputfilename.

```
library(rmetasim)
habitats <- 2
carryjuvpop <- 500
carryadpop <- 10
stages <- 2
rland <- NULL
rland <- new.landscape.empty()
rland <- new.intparam.land(rland, h = habitats, s = stages, totgen = 101)
rland <- new.switchparam.land(rland, mp = 1)
rland <- new.floatparam.land(rland)

Sj <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Rj <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Mj <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
rland <- new.local.demo(rland,S = Sj,R = Rj,M = Mj)

Sa <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Ra <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Ma <- matrix(c(0, 0, 0, 1), nrow = 2, byrow = T)
rland <- new.local.demo(rland,S = Sa,R = Ra,M = Ma)

S <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0),nrow = 4,byrow = T)
R <- matrix(c(0,0,0,50,0,0,0,0,0,0,0,0,0,0,0,0),nrow = 4,byrow = T)
M <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),nrow = 4,byrow = T)
rland <- new.epoch(rland,S=S,R=R,M=M,carry=c(carryjuvpop,carryadpop))

#one could simply change the ploidy of this locus to examine the decay in haploid "het"
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1500)


rland <- new.individuals(rland, c(500,0,0,10))

#the routine below simulates 50 replicates, provides heterozygosity every 10 generations, and #then writes this to an output
file on the C drive.
numreps <-  50
numsteps <- 10
stepsize <- 10
l.exp <- list(numreps)
```

```
het <- matrix(nrow=numsteps,ncol=numreps)
rland.start <- rland
for (j in 1:numreps) {
  rland <- rland.start
  l.exp[[j]] <- list(numsteps + 1)
  l.exp[[j]][[1]] <- rland
  for (i in 1:numsteps) {
    if (dim(rland$individuals)[1]>0)
      {
        rland <- simulate.landscape(rland, stepsize)
      }
    rland <- rland
    if (dim(rland$individuals)[1]==0){
      print (paste("Exctinction! rep, step",j,",",i));
      f <- 0
    } else {
      f <- exp.het.landscape(rland)[1]
                                #          if (is.na(f)){f <- 0}
    }
              het[i,j] <- f
    l.exp[[j]][[i + 1]] <- rland
  }
}

meanhet <- rowMeans(het)

for(i in 1:numreps){
   write(c(het[,i]),file=paste("C:/hetout.txt"),ncol=10,append=T)
}
```

Part C. The following text provides input for simulating the increase in Weir and Cockerham's θ at a haploid locus for 5 isolated populations of N = 50. One need only add the "#" symbol in front of the new.locus line indicating haploid and delete the same symbol from the new.locus line indicating the diploid locus to switch between the two ploidy levels. The demographic matrices are described in the previous section.

```
library(rmetasim)
habitats <- 10
carryjuvpop <- 500
carryadpop <- 50
stages <- 2
rland <- NULL
rland <- new.landscape.empty()
rland <- new.intparam.land(rland, h = habitats, s = stages, totgen = 101)
rland <- new.switchparam.land(rland, mp = 1)
rland <- new.floatparam.land(rland)

for(x in 1:5) {
Sj <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Rj <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Mj <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
rland <- new.local.demo(rland,S = Sj,R = Rj,M = Mj)

Sa <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Ra <- matrix(c(0, 0, 0, 0), nrow = 2, byrow = T)
Ma <- matrix(c(0, 0, 0, 1), nrow = 2, byrow = T)
rland <- new.local.demo(rland,S = Sa,R = Ra,M = Ma)
}
```

```
S <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0),nrow = 20,byrow = T)

R <- matrix(c(0,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,10,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,10,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0),nrow = 20,byrow = T)

M <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,
                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
                    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),nrow = 20,byrow = T)

rland <-
new.epoch(rland,S=S,R=R,M=M,carry=c(carryjuvpop,carryadpop,carryjuvpop,carryadpop,carryjuvpop,carryadpop,carryjuvp
op,carryadpop,carryjuvpop,carryadpop))

#the next two lines indicate the type of marker that will be used in the sims, currently only 1N
#rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 2)
rland <- new.locus(rland, type = 1, ploidy = 1, transmission = 1, numalleles = 2)


rland <- new.individuals(rland, c(500,0,0,50,500,0,0,50,500,0,0,50,500,0,0,50,500,0,0,50))

#need to use lines below to replicate 50 sims and extract wanted theta info to file

numreps <-  50
numsteps <- 10
stepsize <- 10
l.exp <- list(numreps)
the <- matrix(nrow=numsteps,ncol=numreps)
rland.start <- rland
for (j in 1:numreps) {
  rland <- rland.start
  l.exp[[j]] <- list(numsteps + 1)
  l.exp[[j]][[1]] <- rland
  for (i in 1:numsteps) {
    if (dim(rland$individuals)[1]>0)
    {
      rland <- simulate.landscape(rland, stepsize)
    }
    rland <- rland
    if (dim(rland$individuals)[1]==0){
      print (paste("Exctinction! rep, step",j,",",i));
      fst <- 0
    } else {
          rt = landscape.to.rtheta(rland)
                rt.sub = rt[rt$pop %in% c(2,4,6,8,10),]
          fst <- popstruct(rt.sub)[1]
    }
    the[i,j] <- fst
    l.exp[[j]][[i + 1]] <- rland
  }
}

meanthe <- rowMeans(the)

for(i in 1:numreps){
    write(c(the[,i]),file=paste("c:/thetaout.txt"),ncol=10,append=T)
}
```

Part D.

The text below can be cut and pasted to simulate a baleen whale population (N = 20000) based on a gray whale demographic matrix. The simulation includes 8 nuclear loci and 1 mtDNA sequence of 500 bases, sampled for 100 individuals every 1000 time steps. Ten replicates are simulated and an array of genetic summary statistics are obtained. Note that it is necessary to have the "ape" package, as well as the RMETASIM package, installed in order to use these scripts.

```
library(rmetasim)
library(ape)

habitats <- 1
carrycap <- 20000
stages <- 5
rland <- NULL

rland <- new.landscape.empty()
rland <- new.intparam.land(rland, h = habitats, s = stages, totgen = 10001)
rland <- new.switchparam.land(rland, mp = 1)
rland <- new.floatparam.land(rland)

S <- matrix (c(0.730, 0, 0, 0, 0,
               0.210, 0, 0, 0, 0,
               0, 0.470, 0, 0.946, 0,
               0, 0, 0.946, 0, 0,
               0, 0.470, 0, 0, 0.954), nrow=5, byrow = T)

R <- matrix (c(0, 0, 0, 1, 0,
               0, 0, 0, 0, 0,
               0, 0, 0, 0, 0,
               0, 0, 0, 0, 0,
               0, 0, 0, 0, 0), nrow = 5, byrow = T)


M <- matrix (c(0, 0, 0, 0, 0,
               0, 0, 0, 0, 0,
               0, 0, 0, 0, 0,
               0, 0, 0, 0, 1,
               0, 0, 0, 0, 0), nrow = 5, byrow = T)

rland <- new.local.demo(rland,S,R,M)
rland <- new.epoch.island(rland,0,c(0,0,0,0,0),c(0,0,0,0,0),0,c(0,0,0,0,0),c(0,0,0,0,0),
0,c(0,0,0,0,0),c(0,0,0,0,0), carry = rep(carrycap,habitats))

rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1, mutationrate = 0.002)
rland <- new.locus(rland, type = 2, ploidy = 1, transmission = 1, numalleles = 1, allelesize = 500, mutationrate = 0.005)


mrate <- rland$loci[[1]]$rate[[1]]

rland <- new.individuals(rland, c(7400,1400,2800,2600,5800))

ss <- 100
numreps <-  10
```

```
numsteps <- 10
stepsize <- 1000
l.exp <- list(numreps)
Nu4 <- matrix(nrow=numsteps,ncol=numreps)
het <- matrix(nrow=numsteps,ncol=numreps)
na <- matrix(nrow=numsteps,ncol=numreps)
knuc <- matrix(nrow=numsteps,ncol=numreps)
hnuc <- matrix(nrow=numsteps,ncol=numreps)
mtDNAt <- matrix(nrow=numsteps,ncol=numreps)
mthd <- matrix(nrow=numsteps,ncol=numreps)
mthaps <- matrix(nrow=numsteps,ncol=numreps)
mtbin <- matrix(data=0,nrow=numreps,ncol=ss)


rland.start <- rland
for (j in 1:numreps) {
  rland <- rland.start
  l.exp[[j]] <- list(numsteps + 1)
  l.exp[[j]][[1]] <- rland
  for (i in 1:numsteps) {
      rland <- simulate.landscape(rland, stepsize)
      rland <- rland
      rland2 <- rland
      rland2$individuals <- rland$individuals[1:ss,]
      alvec <- vector("list",length(rland2$loci)-1)
      acnt1 <- allelecount.landscape(rland2)
      cnt <- 0
      for (k in 1:length(alvec))
        {
         alvec[[k]] <- acnt1$Freq[(acnt1$pop==1)&(acnt1$loc==k)]
         cnt <- cnt + length(alvec[[k]])
        }
      het[i,j] <- mean(exp.het.landscape(rland2)[1:8])
      na[i,j] <- cnt/length(alvec)
      knuc[i,j] <- mean(theta.k.landscape(rland2)[1:8])
      hnuc[i,j] <- mean(theta.h.landscape(rland2)[1:8])
      mtDNAt[i,j] <- theta.s.landscape(rland2)[9]
      mthd[i,j] <- exp.het.landscape(rland2)[9]
      mthaps[i,j] <- length(acnt1$alleles[(acnt1$loc==9)])
      l.exp[[j]][[i + 1]] <- rland
  }
  x <- sort(acnt1$Freq[(acnt1$pop==1)&(acnt1$loc==9)],decreasing=T)
  for(h in 1:length(x)) {
            mtbin[j,h] <- x[h]
  }
}

meanNu4a <- rowMeans(knuc)
meanNu4b <- rowMeans(hnuc)
meanhet <- rowMeans(het)
meanna <- rowMeans(na)
meanhd <- rowMeans(mthd)
meanhaps <- rowMeans(mthaps)
mtbinned <- colMeans(mtbin)

vtha <- c(0,numsteps)
vthb <- c(0,numsteps)
vhet <- c(0,numsteps)
```

```
vna <- c(0,numsteps)
vhd <- c(0,numsteps)
vhaps <- c(0,numsteps)
for (i in 1:numsteps){
   vtha[i] <- var(knuc[i,])
   vthb[i] <- var(hnuc[i,])
   vhet[i] <- var(het[i,])
   vna[i] <- var(na[i,])
   vhd[i] <- var(mthd[i,])
   vhaps[i] <- var(mthaps[i,])
   }

vhbin <- c(0,ss)
for (i in 1:ss){
  vhbin[i] <- var(mtbin[,i])
  }

write(c("u=",mrate,"  K=",carrycap),file=paste("C:/output.txt"),ncol=4,append=T)
write(c("reps =",numreps,"  stepsize =",stepsize),file=paste("C:/output.txt"),ncol=4,append=T)
write("meantheta k-alleles (var)            meantheta heterozygosity (var)              meanhet(var)          mean#alleles(var)
meanmtDNAhd(var)           mean#mtDNAhaps(var)",file=paste("C:/ output.txt"),ncol=1,append=T)
   for(i in 1:numsteps) {
      write(c(meanNu4a[i],vtha[i],"      ",meanNu4b[i],vthb[i],"      ",meanhet[i],vhet[i],"     ",meanna[i],vna[i],"
",meanhd[i],vhd[i],"   ",meanhaps[i],vhaps[i]),file=paste("C:/ output.txt",collapse=" "),ncol=17,append=T)
    }
write(" ",file=paste("C:/output.txt"),ncol=1,append=T)
write("haplotype mean and variance",file=paste("C:/output.txt"),ncol=1,append=T)
for(i in 1:ss) {
      write(c(mtbinned[i],vhbin[i]),file=paste("C:/output.txt",collapse="    "),ncol=17,append=T)
    }
write(" ",file=paste("C:/output.txt"),ncol=1,append=T)
write(" ",file=paste("C:/output.txt"),ncol=1,append=T)
```