Progress on TOSSM Dataset Generation

Karen K. Martien, Southwest Fisheries Science Center, La Jolla, CA, USA

Abstract

This paper outlines progress made in the generation of simulated datasets for use in the Testing of Spatial Structure Methods (TOSSM) project, including changes made to the life history matrices being used in the simulations and changes made to Rmetasim (Strand, 2002), the program being used to run the simulations. During the TOSSM workshop held in La Jolla, CA, in January of 2003, a total of 90 different population structure scenarios were chosen for simulation during Phase I of the TOSSM project (IWC 2004). Descriptions of those 90 scenarios are given in Appendix 1. As of May 5, 2006, 12 scenarios have been completed, most of which have carrying capacities of K=7,500 individuals. With current computing capabilities, completion of all scenarios listed in Appendix 1 is expected to take an additional 6 months. I provide suggestions for changes in some parameter values and a more focused approach to sensitivity tests in order to speed the completion of the simulations.

Updated Life History Matrices

Martien et al. (SC/56/SD5) presented two life history matrices for use in the TOSSM model. These matrices describe vital rates when the population is near zero population density (ZPD) and near carrying capacity (K). These matrices were based on a post-birth pulse model, in which the population is censused immediately after new individuals are born. Subsequent to the development of these matrices, a new version of Rmetasim was released in which reproduction occurs before growth and survival in each year of the simulation. Thus, Rmetasim now represents a pre-birth pulse model, such that at the end of each simulation year, the youngest animals in the population are one year old. This changed required re-calculation of the life history matrices in order to produce the expected growth rates ($\lambda = 1.072$ at ZPD, $\lambda = 1$ at K).

The new matrices were re-calculated using the same fixed-stage-duration model (Table 1) as described in Martien et al. (SC/56/SD5). The duration of the first juvenile stage was reduced by one year and the fertility rates were multiplied by the juvenile survival rate to account for the fact that new individuals must be born and survive for one year before being counted.

Table 1. Estimates of survival rate (σ_i) and duration (T_i) for each stage and the resulting parameter estimates for the fixed stage duration model. P_i is the probability of an individual surviving and remaining in stage i, while G_i is the probability of an individual surviving and moving into the next stage. F_i is the fertility rate for stage i. Because the probability that a lactating female gave birth is 1.0, F_i is simply equal to the juvenile survival rate.

Stage	σ_i	T_i	P_i	G_i	F_i
Near zero population density:					
Juvenile1	0.94	3	0.73	0.210	0
Juvenile2	0.94	1	0	0.94	0
Fertile female	0.946	1	0	0.946	0
Lactating female	0.946	1	0	0.946	0.94
Adult male	0.954	-	0.954	-	0
At carrying capacity:					
Juvenile1	0.925	5	0.768	0.157	0
Juvenile2	0.925	4	0.720	0.205	0
Fertile female	0.946	3	0.648	0.300	0
Lactating female	0.946	1	0	0.946	0.925
Adult male	0.954	-	0.954	-	0

The juvenile survival rate in the carrying capacity matrix was also increased slightly, from 0.92 to 0.925. This parameter was not based on empirical estimates, but rather was calculated to produce the desired value of λ. The value calculated for juvenile survival rate by Martien et al. (SC/56/SD5) resulted in a λ of 0.998. The new value of 0.925 results in a λ of 1.0003, closer to the desired value of 1.0. The updated matrices are given in Table 2.

Table. 2. Updated stage-based matrices for use at a) zero population density and b) carrying capacity. Stage class abbreviations are juve1 = juvenile1, juve2 = juvenile2, fert = fertile female, lact = lactating female, and male = adult male.

a)	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

b)	juve1	juve2	fert	lact	male
juve1	0.768	0	0	1.0	0
juve2	0.157	0.720	0	0	0
fert	0	0.102	0.648	0.946	0
lact	0	0	0.300	0	0
male	0	0.102	0	0	0.954

Updates to Rmetasim

In the last few months, three changes have been made to Rmetasim (Strand 2002) that have expanded its functionality and utility for the TOSSM project. They are the incorporation of density dependent growth, the tracking of individual identity and parentage, and the ability to initialize simulations from a coalescent model.

Density Dependence – I modified the source code of Rmetasim 1.0.2 to allow the incorporation of density dependent growth. I sent my modifications to the developer of Rmetasim (Allan Strand), who has incorporated them into the latest release of Rmetasim, version 1.0.8. This version includes a new 'switch parameter' named 'dd' which, when set to 1, creates a landscape with density dependent growth. The user must then enter two sets of local demography matrices, one representing the vital rates near zero population density (ZPD), the second representing vital rates at carrying capacity (K). The model then interpolates between those matrices as a linear function of population size in order to determine the vital rates at any given point in the simulation. This allows the user complete control over the life history stage at which density dependence occurs. For instance, in the life history matrices being used for TOSSM and described in both the previous section and in Martien et al. (SC/56/SD5), the ZPD and K matrices differ in terms of juvenile survival rate, age at first reproduction, and inter-birth interval.

The ZPD and K life history matrices are both entered using the Rmetasim command 'new.local.demo()'. That command now accepts a new optional argument called 'k'. If 'k = 0' (default), the life history matrix being enter is assumed to represent vital rates at ZPD. If 'k = 1', the matrix represents vital rates at K. If multiple local demographies are entered, the K matrices must be entered in the same order as their corresponding ZPD matrices. Furthermore, all ZPD matrices must be entered before the K matrices are entered.

To test the density dependence features of Rmetasim 1.0.8, the expected growth trajectory for a population was calculated by projecting the above life history matrices forward through time, at each time step interpolating between the K and Z matrices to determine the appropriate projection matrix for the current time step. The resulting growth trajectory was compared to that obtained by running Rmetasim ten times and averaging the abundance at each time step across replicate simulations. The trajectory from Rmetasim closely matched the projected growth curve, suggesting the density dependence is behaving as expected (Figure 1).

When density dependence is turned off, Rmetasim employs a 'hard ceiling' at carrying capacity, meaning that if the abundance of a population exceeds carrying capacity at the end of a simulation year, individuals are killed at random until the abundance is reduced to carrying capacity. However, when the density dependent option is turned on, a 'hard ceiling' at carrying capacity would result in an average abundance slightly below carrying capacity because it would be possible for abundance to drift below carrying capacity but not above it. To avoid this artifact, the 'hard ceiling' is set at 110% of carrying capacity when density dependence is employed. Even with density dependence, a 'hard ceiling' is still necessary to prevent small populations from growing beyond their carrying capacity if they are receiving large numbers of dispersers from a large neighboring population.

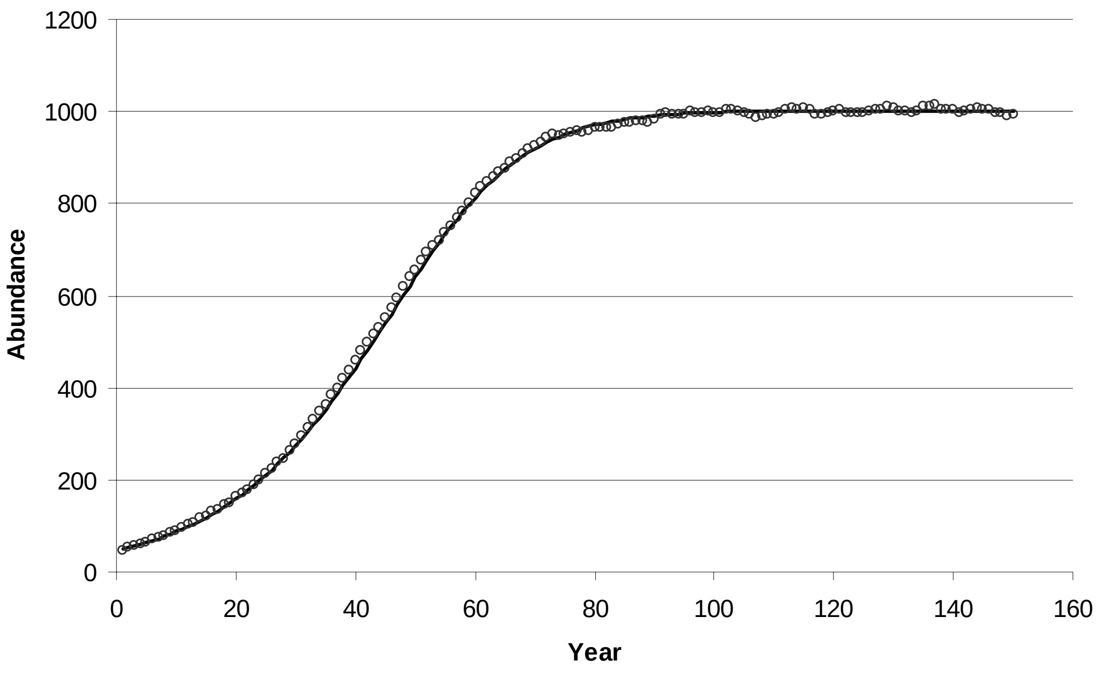

Figure 1. Average abundance at each simulation year for ten replicate simulations of population growth using Rmetasim 1.0.8. Populations were initialized with an abundance of 50 and had carrying capacities of 1000. The solid line shows the expected growth trajectory based on matrix projection of the life history matrices described in the previous section.

Individual identification and paternity/maternity – In response to input received at the TOSSM workshop in Potsdam, Germany, this spring, Allan Strand modified Rmetasim so that it outputs the information necessary to track individuals through time. Each individual is assigned a unique identifier which makes it possible to determine whether or not a given individual appears in samples drawn at different time points in the model. This feature is required if the datasets generated by Rmetasim are to be used to test analytical methods that are based on genetic mark-recapture data. Furthermore, the unique identifiers of each individuals' parents are also recorded, which will be useful for testing methods that involve the estimation of relatedness or paternity.

To implement these changes, Strand inserted into the 'individuals' matrix in each landscape three new columns, which contain the individual's unique identifier, its mother's unique identifier, and its father's unique identifier. Note that this change to the individuals matrix means that landscapes generated by older versions of Rmetasim are not compatible with Rmetasim 1.0.8, and vice versa.

Initializing Rmetasim – At the first TOSSM workshop in 2003, it was agreed that we would initialize Rmetasim using the output from a coalescent model (IWC 2004). The workshop participants agreed that the coalescent was not sufficiently flexible to allow incorporation of all of the complexity that we ultimately hope to encompass in TOSSM. However, initializing with a coalescent substantially decreases the time necessary for Rmetasim simulations to achieve mutation-drift-dispersal equilibrium as compared to initializing with either random allele frequencies or with only a single allele at each locus.

During the Second TOSSM workshop held in Potsdam, Germany, in March of this year, Allan Strand, with the help of other workshop participants, wrote a script to process the output files generated by the coalescent program SimCoal 2.1.2 (Laval and Excoffier 2004) and use them to initialize an Rmetasim landscape. This script will be incorporated into a future release of Rmetasim as a built-in method. In the meantime, I have modified it to work with the latest release of Rmetasim and have used it to initialize all TOSSM simulations.

Generating Datasets

Because of the need to initialize Rmetasim with genetic data generated by a coalescent model, simulating a given TOSSM scenario requires several steps, as outlined below. All scripts and input files used to simulate Archetype 2, scenario 17, are given in Appendix 2 as an example. Scripts and input files used for all other scenarios are available upon request.

Step 1: Estimate Ne. The carrying capacity specified for each TOSSM scenario and used to parameterize the Rmetasim simulations constrains the census population size, not effective popualtion size, N_e. The abundance estimate required to parameterize the coalescent models, however, is N_e. It is therefore necessary to estimate N_e for each scenario by running a simulation that contains a single microsatellite locus and a 500 bp mitochondrial DNA sequence, each with no mutation and initialized with 1000 alleles. The simulation is run for 2000 years (100 generations). By calculating heterozygosity at the beginning and end of the simulation, N_e can be estimated using the equation

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

where H_t is heterozygosity at time t and H_o is heterozygosity at time zero. I replicate the simulation 10 times and average across replicates to obtain a mean N_e for both nuclear and mitochondrial markers. It is necessary to estimate N_e separately for the two marker types because the haploid and uni-parentally inherited nature of mitochondrial DNA results in a markedly smaller effective population size for the mitochondrial genome.

Step 2: Generate Coalescent Datasets. The haplotype and allele frequencies used to initialize Rmetasim are generated using the coalescent model SimCoal 2.1.2 (Laval and Excoffier 2004). A 500 bp mtDNA sequence and 18 unlinked microsatellite loci are simulated (see 'Genetic Marker Characteristics' for a more complete description of the loci). 1,000 SimCoal datasets are generated for each scenario.

Step 3: Initialize and Run Rmetasim Simulations. Each SimCoal dataset is used to initialize one Rmetasim landscape using the function 'coalinput.landscape().' This function is currently available as an independent script (see *Initializing Rmetasim* above), but will be incorporated into a future release of Rmetasim. Each landscape is simulated in Rmetasim for 1,000 years (50 generations). The final state of the landscape is saved as an R object (*.rda file). In addition, the microsatellite data are saved to a text file formatted for use with the program Convert (Glaubitz 2004) and the mtDNA data are saved in a text file formatted for analysis in the program Arlequin (Excoffier et al. 2005). These formats were chosen because they are used by many different analytical methods and because both Convert and Arlequin include utilities for converting data into multiple other formats (in fact, that is the primary function of Convert). Separate formats are required for mtDNA and microsatellite data because Convert does not support mtDNA data, while Arlequin is not able to convert data into as many formats as Convert.

Genetic Marker Characteristics

mtDNA – Mitochondrial DNA haplotypes being simulated in the TOSSM datasets are 500 bp in length and have a mutation rate of 5x10-3 per generation for the full sequence. This mutation rate is based on mutation rate estimates for the mitochondrial control region (Heyer et al. 2001) and produced haplotype distributions consistent with what is seen in many large whale species (Figure 2). For Archetype 1, scenario 1 (one population, N=7,500), the number of haplotypes in the final datasets averaged across replicates was 118 (s.d. = 8.62) (Figure 3).

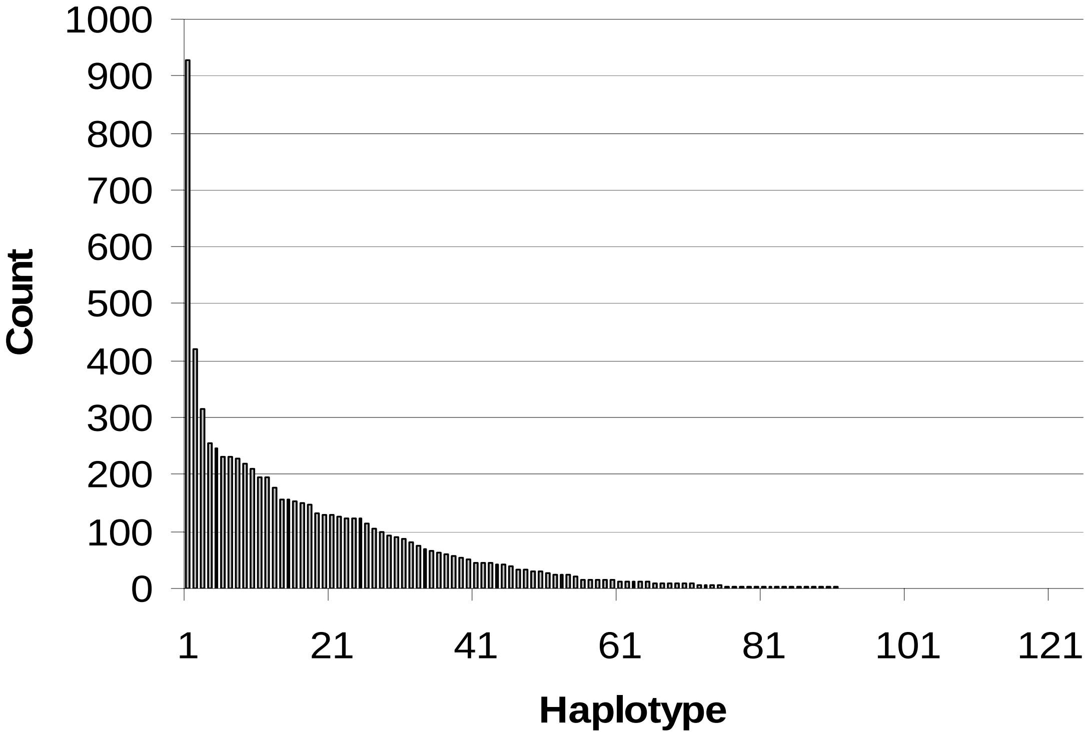

Figure 2. Example haplotype frequency distribution for one replicate of Archetype 1, scenario 1. Haplotypes are order along the x-axis from most to least frequent. The number of individuals with each haplotype (out of 7465) is given on the y-axis. The distribution includes a few very common haploltypes and a long tail of very low frequency haplotypes, as is typically seen in empirical datasets. Note that there were 125 haplotypes in this replicate; haplotypes 92 through 125 were each represented by a single individual and are not visible on the graph.

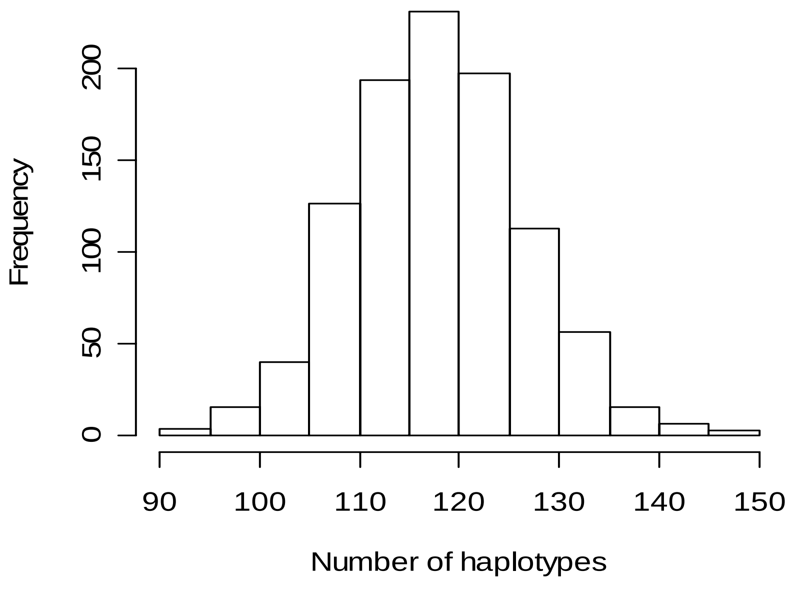

Figure 3. Distribution of number of haplotypes across replicates for Archetype 1, scenario 1.

Microsatellites – The TOSSM datasets use 18 microsatellite loci, six each with mutation rates of 1x10-3, 2x10-3 and 3x10-3. These mutation rates were based on estimates from the literature (see Brohede 2003 for a review) and resulted in allele frequency distributions consistent with those typically seen in datasets used in studies of population structure (Table 3). The number of alleles per locus ranged from 6 to 26 (Figure 4), which is consistent with the number of alleles typically observed in empirical datasets. However, the average number of alleles for replicates of Archetype 1, scenario 1 was nearly the same for the three mutation rates (Table 4). Thus, the mutation rates currently being simulated produce essentially identical allelic distributions, precluding the possibility of using the TOSSM datasets to examine the impact of mutation rate on power for detecting population structure. However, because the number of alleles varies between loci and between replicates, it will be possible to examine the impact of number of alleles on power.

Table 3. Example allele frequencies at all 18 loci for one replicate of Archetype 1, scenario 1.

Allele	Loc1	Loc2	Loc3	Loc4	Loc5	Loc6	Loc7	Loc8	Loc9
1	3276	6349	3798	3920	2813	3873	4237	3785	2487
2	2723	2618	2649	3672	2773	2808	3355	2894	1742
3	2513	2132	2341	1989	2472	1558	3229	2803	1671
4	2008	1881	1982	1919	2397	1344	2144	1680	1584
5	1624	825	1367	1522	2094	1290	1025	1415	1475
6	1400	318	1245	1138	833	1267	685	1169	1308
7	1146	285	725	425	660	1085	244	510	1254
8	237	166	393	277	567	847	11	435	1129
9	3	156	313	68	191	280	0	93	748
10	0	122	106	0	130	259	0	58	714
11	0	61	11	0	0	165	0	49	452
12	0	17	0	0	0	114	0	20	330
13	0	0	0	0	0	40	0	19	28
14	0	0	0	0	0	0	0	0	8
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0

Allele	Loc10	Loc11	Loc12	Loc13	Loc14	Loc15	Loc16	Loc17	Loc18
1	5151	3376	3105	2842	3161	4461	6205	3368	4394
2	3715	2652	2961	2096	2054	3503	3048	2468	2749
3	1984	2594	2871	1701	2038	3399	1799	2392	2527
4	1466	1561	1534	1650	1578	1461	1438	1848	1599
5	957	1210	1300	1640	1237	1094	666	1062	1224
6	726	981	1225	1226	1038	383	580	1044	1088
7	517	827	1111	1127	887	346	468	770	614
8	408	469	680	1101	839	268	211	670	579
9	6	459	131	727	734	15	186	561	112
10	0	445	12	464	729	0	168	354	44
11	0	354	0	340	425	0	158	153	0
12	0	2	0	16	72	0	3	99	0
13	0	0	0	0	60	0	0	92	0
14	0	0	0	0	59	0	0	45	0
15	0	0	0	0	10	0	0	3	0
16	0	0	0	0	7	0	0	1	0
17	0	0	0	0	2	0	0	0	0

Table 4. Average number of alleles per locus for 1000 replicates of Archetype 1, scenario 1. Averages are over all six loci of a given mutation rate.

Mutation Rates	# alleles	s.d.	Range
1x10-3	12.04	2.77	5-24
2x10-3	12.65	2.95	6-26
3x10-3	12.93	2.96	6-26

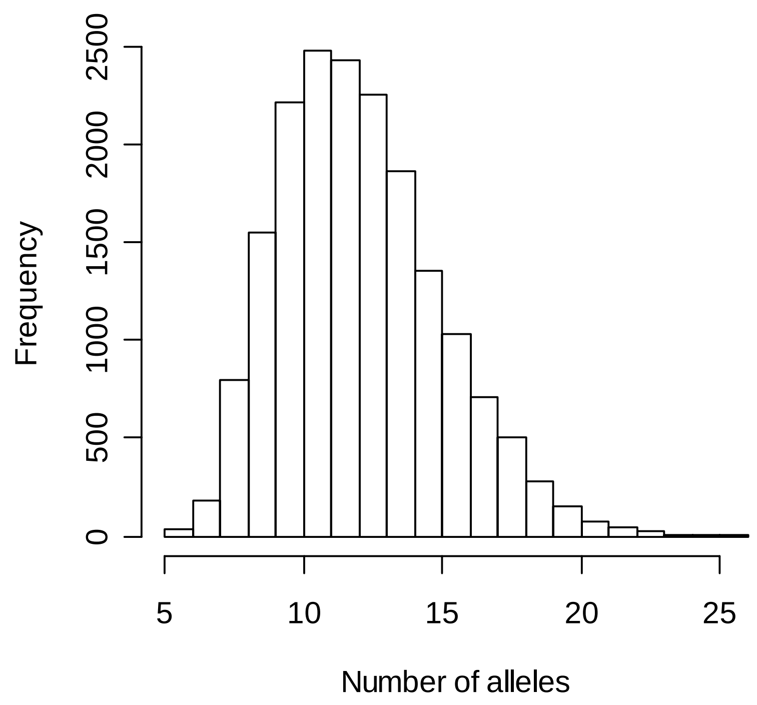

Figure 4. Distribution of number of alleles per locus. Data are combined for all 18 microsatellite loci from all 1000 replicates of Archetype 1, scenario 1.

Runtime

As of May 5, 2006, datasets have been generated for 12 of the 90 scenarios outlined at the first TOSSM workshop. All simulations run so far have been run on either a 2.00GHz Pentium M laptop with 1 GB of RAM or a Sun workstation running Solaris 7. The workstation has 3 processors, making it possible to run three simulations simultaneously with no loss in performance. Furthermore, the workstation is available 24 hours per day, 7 days per week, whereas the PC is only available during non-working hours.

1000 replicates of a scenario with a carrying capacity of 7,500 can be completed in approximately 17 hours on the PC, but take approximately 5 days on the Solaris machine (using only one of three processors). Scenarios with carrying capacities of 15,000 take about twice as long to run – approximately 34 hours on the PC and 10 days on the Solaris machine (again using only a single processor). No scenarios with carrying capacities of 30,000 have been run, but it is expected they will take twice again as long as those with K=15,000. Table 5 shows the number of scenarios remaining to be simulated for each carrying capacity and their approximate runtimes on both the PC and Solaris. Assuming simulations are run 80 hours per week on the PC (40 hours on weekends and 13.3 hours per night, 3 nights per week) and around the clock on all three processors of the workstation (allowing for 20% downtime due to computer maintenance, delays in starting new a simulation when one finishes, etc.), the remaining simulations can be complete in 24 weeks.

Allan Strand, creator of Rmetasim, has offered the use of 4 linux machines for running TOSSM simulations. No tests have been performed to determine the runtime of simulations on these machines. Nonetheless, use of Strand's cluster reduce the time to completion by an unknown, but hopefully substantial, amount.

Table 5. Approximate run time for scenarios of a given carrying capacity and number of scenarios remaining to be simulated for Archetype x Carrying Capacity combination.

	Runtime (days)		Scenarios remaining			
K	PC	Solaris	Archetype 1	Archetype 2	Archetype 3	Archetype 5
7,500	0.695	5	0	10	4	5
15,000	1.4	10	1	18	4	5
30,000	2.8	20	1	20	4	5

In light of the amount of time required to complete all simulations specified in Appendix 1, especially those with high carrying capacities, the TOSSM steering committee should revisit the question of choosing parameter combinations to simulate. One of the goals of TOSSM is to generate datasets and performance analyses that will be useful to geneticists working on a wide range of species. To that end, simulating carrying capacities of 2,500 individuals, 7,500 individuals and 15,000 individuals may be more appropriate. Though very large populations are common for marine species, terrestrial geneticists typically work in systems with much lower abundance and are therefore likely to be interested in simulations of only a few thousand individuals. Furthermore, the impact of increased abundance on the performance of various analytical methods is likely to be adequately demonstrated by the simulations with carrying capacities of 15,000. Substituting scenarios in which K=2,500 for all of the K=30,000 scenarios listed in Appendix 1 would reduce the expected time to completion of the datasets by nearly 14 weeks. If there are specific regions of the parameter space in which the steering committee feels more could be learned by running simulations with abundances greater than 15,000, additional select simulations could be run in the future rather than running all thirty K=30,000 scenarios currently planned (Appendix 1).

Similarly, given the long runtime of the scenarios with carrying capacities of 15,000, a selective approach to running scenarios with this carrying capacity may be more appropriate than a full-cross of this value of K with the other parameter values. For instance, the impact of unequal abundance can likely be investigated using only those scenarios with carrying capacities of 7,500 (Archetype 2, scenarios 16-20 and 46-50). Eliminating scenarios with carrying capacities of 15,000 and unequal population sizes would reduce the total remaining computation time by approximately 2.5 weeks.

Simulating Archetypes III – V

Archetypes III-V have not yet been simulated. The steps necessary to begin these simulations are as follows:

Archetype III – Unlike the other four archetypes, Archetype III assumes continuous genetic variation over space. At the 2003 TOSSM workshop in La Jolla, CA, it was agreed that this archetype could be simulated by creating an Rmetasim landscape that contained many small populations with relatively high dispersal rates between adjacent populations (IWC 2004). Ralph Tiedeman has offered to design the necessary landscape matrices to implement this Archetype.

Archetype IV – As noted above, Archetype IV can be simulated by assigning overlapping spatial coordinates to samples drawn from Archetype II. Thus, no additional Rmetasim simulations are necessary in order to generate Archetype IV datasets.

Archetype V – Archetype V postulates population structure at mitochondrial loci but panmixia at nuclear loci. Rmetasim includes a male reproduction matrix that specifies the probability of a male siring an offspring in each population. Thus, simulating Archetype V will simply requiring modifying the scripts used in Archetype II so that each male has an equal probability of siring an offspring in each population.

Storage, archiving, and availability of datasets

The TOSSM datasets are quite large. For scenarios in which K=7,500, the Rmetasim landscape file is approximately 1450 KB while the Convert files are 650 KB. When K=15,000, the landscape and Convert files are 2750 KB and 1400 KB, respectively. Zipping files with WinZip reduces the file size by approximately 80%. If all replicates of all scenarios listed in Appendix 1 are run, the resulting data will occupy approximately 100 GB of computer storage after being zipped.

The intention is to make the TOSSM datasets available via the web. However, the large size of the data files presents some challenges. It is unclear whether the expertise necessary to serve such a large quantity of data efficiently are present at Southwest Fisheries Science Center. This matter is currently under investigation.

Summary

All programming necessary to use the programs SimCoal 2.1.2 (Laval and Excoffier 2004) and Rmetasim 1.0.8 (Strand 2002) to generate the TOSSM datasets is complete and dataset generation has begun. Completion of all simulations specified in Appendix 1 is expected to take nearly six months. From the simulations that have been completed, the following conclusions can be drawn:

- The mutation rates being used for the mitochondrial and microsatellite loci are appropriate and result in distributions of the number of haplotypes/alleles comparable to what is seen in empirical datasets.
- The three mutation rates being used for the microsatellite loci are not sufficiently different to allow meaningful comparisons between mutation rates. However, the variation in the number of alleles within and between replicates is sufficient to allow stratification on that basis.
- The runtime of the model is long enough that simulation of all combinations of parameters specified in Appendix 1 is not practical. The steering committee should consider eliminating all scenarios with carrying capacities of 30,000 and replacing them with scenarios with carrying capacities of 2,500, thereby shifting the range of abundances to be considered. Simulations with carrying capacities of 15,000 should be run selectively rather than with all possible combinations of other parameters. Simulations with carrying capacities greater than 15,000 can be run on an as-needed basis if regions of the parameter space have not been adequately explored with the lower abundance scenarios.
- The quantity of data being generated is large enough that storage and distribution could be problematic. This issue warrants further investigation.

References

Brohede, J. 2003. Rates and patterns of mutation in microsatellite DNA. Acta Universitatis Upsaliensis. *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 803. 51pp. Uppsala. ISBN 91-554-5524-7.

Excoffier, L., G. Laval, and S. Schneider. 2005. Arlequin ver. 3.0: An integrated software package for population genetics data analysis. Evolutionary Bioinformatics Online. **1**:47-50.

Glaubitz, J.C. 2004. CONVERT: A user-friendly program to reformat diploid genotypic data for commonly used population genetic software packages. Molecular Ecology Notes. **4**:309-310.

Heyer E., E. Zietkiewicz, A. Rochowski, V. Yotova, J. Puymirat, D. Lubuda. 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.

IWC. 2004. Report of the Workshop to design simulation-based performance tests for evaluation methods used to infer population structure from genetic data. Journal of Cetacean Research and Management. **6**(Suppl.):469-485.

Laval, G. and L. Excoffier. 2004. SIMCOAL 2.0: a program to simulate genomic diversity over large recombining regions in a subdivided population with a complex history. Bioinformatics. **20**:2485-2487.

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.

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

Appendix 1 – Review of the TOSSM dataset specifications

The participants at the TOSSM workshop held in La Jolla, CA, in January of 2003 outlined the parameters to be used in generating datasets for the TOSSM project (IWC 2004). The datasets are to represent five population structure archetypes:

Archetype I: a single panmictic population
Archetype II: a linear stepping-stone, with dispersal occurring only between adjacent populations. There are four variants of this archetype: a) two populations with equal abundance, b) two populations with unequal abundance (90:10), c) three populations with equal abundance, and d) three populations with unequal abundance (45:45:10).
Archetype III: diffusion-like isolation by distance
Archetype IV: two populations with separate breeding grounds but overlapping feeding ground. This archetype can be simulated by simply drawing samples from Archetype IIa datasets and assigning overlapping spatial coordinates to samples from the two populations.
Archetype V: A single breeding population with separate feeding grounds. Feeding ground philopatry is learned from the mother.

For each Archetype, three carrying capacities will be simulated – 7,500, 15,000 and 30,000. These represent total abundances summed across all populations in an archetype. Because all simulations are initialized at carrying capacity and are simulated under density dependent population growth, the simulated populations will remain at or very near carrying capacity for the entire simulate.

Annual dispersal rates of 0, 5x10-6, 5x10-5, 5x10-4 and 5x10-3 were chosen to span the range of rates that might be of interest conservation biologists. The lowest of these rates corresponds to approximately one disperser per generation and will therefore result in populations that are following independent evolutionary trajectories. The highest rate will result in populations that are demographically independent, but genetically very similar.

The total number of simulations that need to be run cannot be determined by simply multiplying the number of archetypes, dispersal rates and carrying capacities, as some parameters are not relevant to certain archetypes (e.g., there is no dispersal for archetype I) and Archetype IV can be simulated by simply sampling from Archetype II. Table A1 summarizes the total number of scenarios that must be simulated.

Table A1. Number of scenarios that must be simulated for each Archetype

Archetype	Variants	Carrying Capacities	Dispersal Rates	No. of scenarios
I	1	3	--	3
II	4	3	5	60
III	1	3	4	12
IV	1	3	5	0*
V	1	3	5	15
Total				90

*Since Archetype IV can be generated by assigning overlapping spatial coordinates to samples drawn from Archetype II datasets, no new simulations will be required for it.

Table A2 is a complete list of all scenarios laid out at the TOSSM workshop in La Jolla, CA (IWC 2004). For each scenario, I indicate whether or not it has been completed and, if not, a priority level to indicate the order in which I intend to run the remaining simulations.

Table A2. Specifications for scenarios using all possible combinations of parameters specified at the TOSSM workshop (IWC 2004). Archetype 4 is not included in the list, as it can be generated by simply sampling from Archetype 2 simulations. A key to interpreting the last column follows the table.

Scenario name	Archetype	#populations	abundance	dispersal	Priority/Status*
Arch1_1	1	1	7500	0	X
Arch1_2	1	1	15000	0	3
Arch1_3	1	1	30000	0	e
Scenario name	Archetype	#populations	abundance	dispersal	
Arch2_1	2	2	7500even	0	1
Arch2_2	2	2	7500even	5x10-6	X
Arch2_3	2	2	7500even	5x10-5	X
Arch2_4	2	2	7500even	5x10-4	X
Arch2_5	2	2	7500even	5x10-3	X
Arch2_6	2	2	15000even	0	3
Arch2_7	2	2	15000even	5x10-6	X
Arch2_8	2	2	15000even	5x10-5	3
Arch2_9	2	2	15000even	5x10-4	X
Arch2_10	2	2	15000even	5x10-3	3
Arch2_11	2	2	30000even	0	e
Arch2_12	2	2	30000even	5x10-6	e
Arch2_13	2	2	30000even	5x10-5	e
Arch2_14	2	2	30000even	5x10-4	e
Arch2_15	2	2	30000even	5x10-3	e
Arch2_16	2	2	7500uneven	0	2
Arch2_17	2	2	7500uneven	5x10-6	X
Arch2_18	2	2	7500uneven	5x10-5	2
Arch2_19	2	2	7500uneven	5x10-4	2
Arch2_20	2	2	7500uneven	5x10-3	2
Arch2_21	2	2	15000uneven	0	e
Arch2_22	2	2	15000uneven	5x10-6	e
Arch2_23	2	2	15000uneven	5x10-5	e
Arch2_24	2	2	15000uneven	5x10-4	e
Arch2_25	2	2	15000uneven	5x10-3	e
Arch2_26	2	2	30000uneven	0	e
Arch2_27	2	2	30000uneven	5x10-6	e
Arch2_28	2	2	30000uneven	5x10-5	e
Arch2_29	2	2	30000uneven	5x10-4	e
Arch2_30	2	2	30000uneven	5x10-3	e

* key to ‘Priority/Status’ column:
X = complete
e = proposed for elimination
r = proposed for replacement with an identical scenario except with abundance = 2,500
? = simulating scenario depends on receipt of appropriate dispersal matrices; no priority assigned
1-4 = relative priority, 1 being highest and 4 being lowest

Table 2A continued.

Scenario name	Archetype	#populations	abundance	dispersal	Priority/Status*
Arch2_31	2	3	7500even	0	1
Arch2_32	2	3	7500even	5x10-6	X
Arch2_33	2	3	7500even	5x10-5	X
Arch2_34	2	3	7500even	5x10-4	X
Arch2_35	2	3	7500even	5x10-3	X
Arch2_36	2	3	15000even	0	3
Arch2_37	2	3	15000even	5x10-6	3
Arch2_38	2	3	15000even	5x10-5	3
Arch2_39	2	3	15000even	5x10-4	3
Arch2_40	2	3	15000even	5x10-3	3
Arch2_41	2	3	30000even	0	e
Arch2_42	2	3	30000even	5x10-6	e
Arch2_43	2	3	30000even	5x10-5	e
Arch2_44	2	3	30000even	5x10-4	e
Arch2_45	2	3	30000even	5x10-3	e
Arch2_46	2	3	7500uneven	0	2
Arch2_47	2	3	7500uneven	5x10-6	2
Arch2_48	2	3	7500uneven	5x10-5	2
Arch2_49	2	3	7500uneven	5x10-4	2
Arch2_50	2	3	7500uneven	5x10-3	2
Arch2_51	2	3	15000uneven	0	e
Arch2_52	2	3	15000uneven	5x10-6	e
Arch2_53	2	3	15000uneven	5x10-5	e
Arch2_54	2	3	15000uneven	5x10-4	e
Arch2_55	2	3	15000uneven	5x10-3	e
Arch2_56	2	3	30000uneven	0	e
Arch2_57	2	3	30000uneven	5x10-6	e
Arch2_58	2	3	30000uneven	5x10-5	e
Arch2_59	2	3	30000uneven	5x10-4	e
Arch2_60	2	3	30000uneven	5x10-3	e
Scenario name	Archetype	#populations	abundance	dispersal	
Arch3_1	3	2	7500	5x10-6	?
Arch3_2	3	2	7500	5x10-5	?
Arch3_3	3	2	7500	5x10-4	?
Arch3_4	3	2	7500	5x10-3	?
Arch3_5	3	2	15000	5x10-6	?
Arch3_6	3	2	15000	5x10-5	?
Arch3_7	3	2	15000	5x10-4	?
Arch3_8	3	2	15000	5x10-3	?
Arch3_9	3	2	30000	5x10-6	e
Arch3_10	3	2	30000	5x10-5	e
Arch3_11	3	2	30000	5x10-4	e
Arch3_12	3	2	30000	5x10-3	e

* key to ‘Priority/Status’ column:
X = complete
e = proposed for elimination
r = proposed for replacement with an identical scenario except with abundance = 2,500
? = simulating scenario depends on receipt of appropriate dispersal matrices; no priority assigned
1-4 = relative priority, 1 being highest and 4 being lowest

Table 2A continued.

Scenario name	Archetype	#populations	abundance	dispersal	Priority/Status*
Arch5_1	5	2	7500	0	4
Arch5_2	5	2	7500	5x10-6	4
Arch5_3	5	2	7500	5x10-5	4
Arch5_4	5	2	7500	5x10-4	4
Arch5_5	5	2	7500	5x10-3	4
Arch5_6	5	2	15000	0	4
Arch5_7	5	2	15000	5x10-6	4
Arch5_8	5	2	15000	5x10-5	4
Arch5_9	5	2	15000	5x10-4	4
Arch5_10	5	2	15000	5x10-3	4
Arch5_11	5	2	30000	0	e
Arch5_12	5	2	30000	5x10-6	e
Arch5_13	5	2	30000	5x10-5	e
Arch5_14	5	2	30000	5x10-4	e
Arch5_15	5	2	30000	5x10-3	e

* key to 'Priority/Status' column:
X = complete
e = proposed for elimination
r = proposed for replacement with an identical scenario except with abundance = 2,500
? = simulating scenario depends on receipt of appropriate dispersal matrices; no priority assigned
1-4 = relative priority, 1 being highest and 4 being lowest

Appendix 2 – Scripts and input files for Archetype 2, scenario 17

Step 1: Estimating Ne – The first script is an R script used for estimating Ne. It requires that the packages Rmetasim 1.0.8 and ape be installed.

```
# Script for estimating Ne

library(rmetasim)
library(ape)

scenario <- 'Arch2_sc17'
carrycap <- 7500
dispersal.rate <- 0.000005

habitats <- 2
stages <- 5
rland <- NULL

rland <- new.landscape.empty()
rland <- new.intparam.land(rland, h = habitats, s = stages, totgen = 10000)
rland <- new.switchparam.land(rland, mp = 1, dd = 1)
rland <- new.floatparam.land(rland)

#life history matrices at zero population density
SZ <- 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)

RZ <- matrix (c(0, 0, 0, .94, 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,SZ,RZ,M)

#life history matrices at carrying capacity
SK <- matrix (c(0.768, 0, 0, 0, 0,
              0.157, 0.720, 0, 0, 0,
              0, 0.102, 0.648, 0.946, 0,
              0, 0, 0.300, 0, 0,
              0, 0.102, 0, 0, 0.954), nrow=5, byrow = T)

RK <- matrix (c(0, 0, 0, .925, 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)

rland <- new.local.demo(rland,SK,RK,M,k=1)
bigS <- bigR <- bigM <- matrix(0,(stages*habitats),(stages*habitats))
for (s in 1:stages){
       bigS[s,(s+stages)] <- dispersal.rate/(habitats-1)
       bigS[(s+stages),s] <- bigS[s,(s+stages)]
}
rland <- new.epoch(rland,bigS,bigR,bigM,carry = c((carrycap*.9),(carrycap*.1)))

rland <- new.locus(rland, type = 1, ploidy = 2, transmission = 0, numalleles = 1000, mutationrate = 0.00)
rland <- new.locus(rland, type = 2, ploidy = 1, transmission = 1, numalleles = 1000, allelesize = 500,
mutationrate = 0.00)

ev <- eigen((RK+diag(dim(RK)[1]))%*%SK)$vectors[,1]
ev <- ev/sum(ev)
rland <- new.individuals(rland, c((as.real(round(ev*(carrycap*.9)))),(as.real(round(ev*(carrycap*.1))))))
```

```
numreps <-  10
numsteps <- 100
stepsize <- 20

het <- array(dim=c(habitats,2,(numsteps+1),numreps))

rland.start <- rland
for (j in 1:numreps) {
  rland <- rland.start
  het[,,1,j] <- exp.het.landscape(rland)
  for (i in 1:numsteps) {
      rland <- sim.landscape(rland, stepsize)
      het[,,(i+1),j] <- exp.het.landscape(rland)
       save (rland,het,file="Ne.test.rda")
  }
}
Ne <- array(-99, dim=c(habitats,numreps,2))
aveNe <- matrix(-99,habitats,2)
for (i in 1:habitats){
       Ne[i,,] <- apply(het[i,,,],1,function(x) 0.5/(1-(x[(numsteps+1),]/x[1,])^(1/numsteps)))
       aveNe[i,] <- colMeans(Ne[i,,])
}
```

Step 2: Generate Coalescent Datasets – The following are input files to the program SimCoal 2.1.2 (Laval and Excoffier 2004), which is used to generate coalescent datasets with which to initialize Rmetasim. The first generates the mitochondrial sequence data:

```
//Parameters for the mtDNA for Archetype 2, scenario 17
2 samples to simulate
//Population effective sizes (number of genes)
927
124
//Samples sizes
50
10
//Growth rates
0
0
//Number of migration matrices : 0 implies no migration among demes
1
//Migration rates matrix  0
0       0.000005
0.000005       0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix number
0 historical events
//Number of independent loci [chromosome] (Number of sequences of 300 bp per gamete)
1 0
// Chromosome structure 1 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
DNA    500        0.000      0.005  0.66
```

The next file is for using SimCoal to generate microsatellite data:

```
//Parameters for the microsatellite data for Archtype 2, scenario 17
2 samples to simulate
//Population effective sizes (number of genes)
3953
463
//Samples sizes
400
40
//Growth rates
0
0
//Number of migration matrices : 0 implies no migration among demes
1
//Migration rates matrix  0
0       0.000005
0.000005       0
//historical event: time, source, sink, migrants, new deme size, new growth rate, migration matrix number
0 historical events
//Number of independent loci [chromosome] (Number of sequences of 300 bp per gamete)
```

```
18 1
// Chromosome structure 1 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.001
// Chromosome structure 2 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.001
// Chromosome structure 3 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.001
// Chromosome structure 4 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.001
// Chromosome structure 5 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.001
// Chromosome structure 6 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.001
// Chromosome structure 7 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.002
// Chromosome structure 8 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.002
// Chromosome structure 9 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.002
// Chromosome structure 10 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.002
// Chromosome structure 11 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.002
// Chromosome structure 12 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.002
// Chromosome structure 13 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.003
// Chromosome structure 14 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.003
// Chromosome structure 15 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.003
```

```
// Chromosome structure 16 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.003
// Chromosome structure 17 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.003
// Chromosome structure 18 begins with number of linkage blocks
1
//per block: data type, number of loci, per generation recombination and mutation rates and optional
parameters
Microsatellite    1        0.000      0.003
```

Step 3: Initialize and Run Rmetasim Simulations – The final script is an R script used to generate 1000 replicate datasets for Archetype 2, scenario 17, each initialized with a different SimCoal dataset. This script requires the R packages Rmetasim 1.0.8 and ape. It also requires other R scripts, which are available from the author upon request.

```
# Archetype 2, scenario 17 script.  Landscapes are initialized from SimCoal data.

library(rmetasim)
library(ape)
source("coal2rmet.1.0.7.R")
source("new_locus.R")
source("rland2convert.R")

scenario <- 'Arch2_sc17'
carrycap <- 7500
dispersal.rate <- 0.000005
numreps <-  1000
stepsize <- 1000

habitats <- 2
stages <- 5
rland <- NULL

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

#life history matrices at zero population density
SZ <- 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)

RZ <- matrix (c(0, 0, 0, .94, 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,SZ,RZ,M)

#life history matrices at carrying capacity
SK <- matrix (c(0.768, 0, 0, 0, 0,
              0.157, 0.720, 0, 0, 0,
              0, 0.102, 0.648, 0.946, 0,
              0, 0, 0.300, 0, 0,
              0, 0.102, 0, 0, 0.954), nrow=5, byrow = T)

RK <- matrix (c(0, 0, 0, .925, 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)

rland <- new.local.demo(rland,SK,RK,M,k=1)
bigS <- bigR <- bigM <- matrix(0,(stages*habitats),(stages*habitats))
for (s in 1:stages){
       bigS[s,(s+stages)] <- dispersal.rate/(habitats-1)
       bigS[(s+stages),s] <- bigS[s,(s+stages)]
}
rland <- new.epoch(rland,bigS,bigR,bigM,carry = c((carrycap*.9),(carrycap*.1)))

rland.start <- rland
mt.data <- paste(scenario,"_mt/",scenario,"_mt_",sep="")
nuc.data <- paste(scenario,"_nuc/",scenario,"_nuc_",sep="")
mut.rates <- c(5e-3,rep(1e-3,6),rep(2e-3,6),rep(3e-3,6))

for (i in 1:numreps){

  rland.init <- rland.start
  rland.init <- coalinput.landscape(rland.init,npp=c((carrycap*.9),(carrycap*.1)),
       arlseq=paste(mt.data,(i-1),'.arp',sep=""),
       arlms=paste(nuc.data,(i-1),'.arp',sep=""),mut.rates)
  rland.end <- sim.landscape(rland.init, stepsize)
  save(rland.end, file=paste('Landscapes/',scenario,"_",i,".rda",sep=""))
  rland2convert(rland.end,title=paste(scenario,"_",i," final landscape",sep=""),
       filename=paste("Landscapes/",scenario,"_",i,".txt",sep=""))
}
```