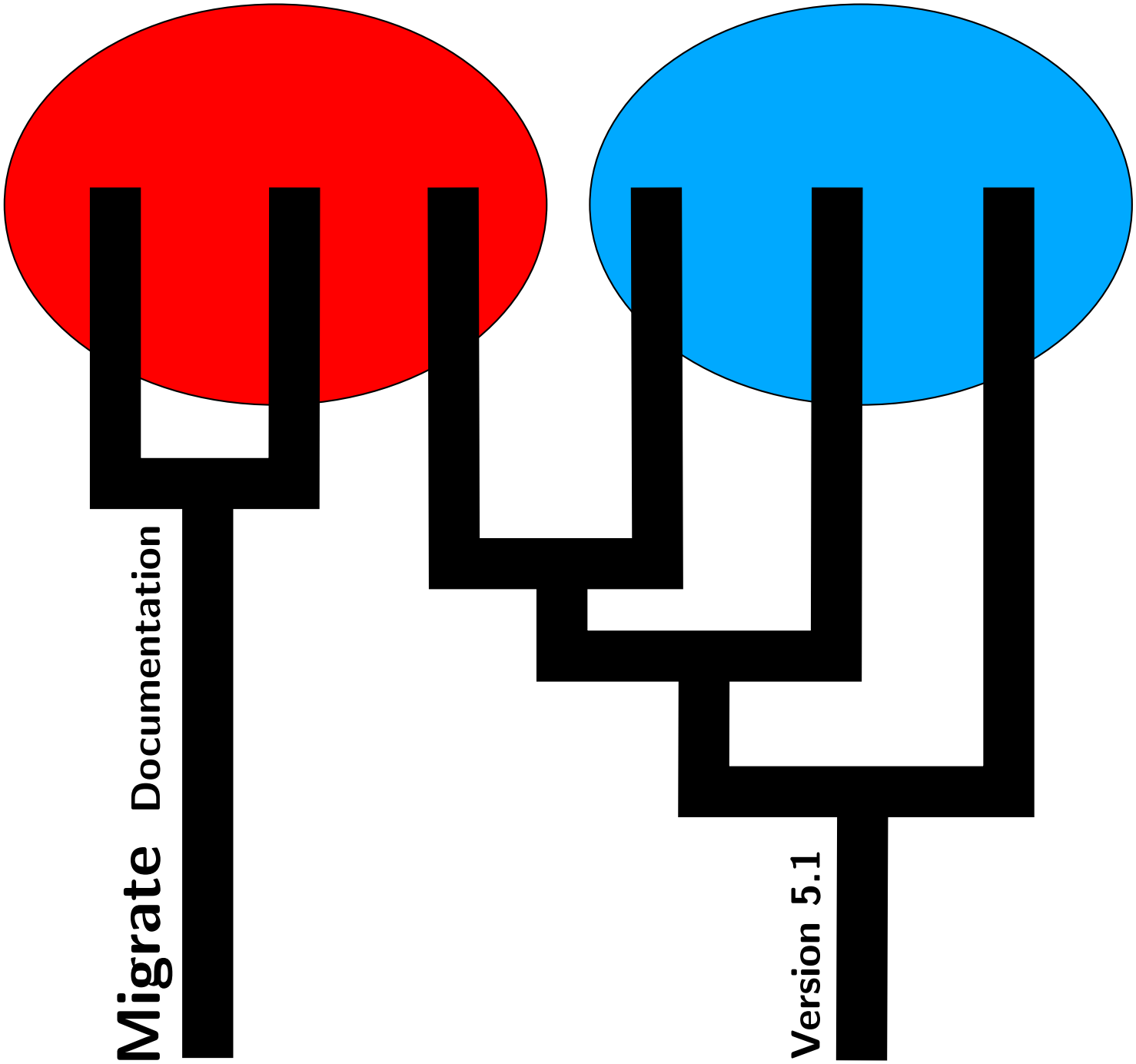


Peter Beerli
Department of Scientific Computing
Florida State University
Tallahassee, FL 32306-4120
email:beerli@fsu.edu

Last update: March 18, 2026
Started: January 1, 1997

For the impatient

Reading manuals is not a favored task of many, me included. But to achieve some results with MIGRATE you should read at least the sections about

- **Data file specifications**
- **Quick guide for achieving "good" results with migrate**.
- You may want to consider to work through this tutorial:
  https://currentprotocols.onlinelibrary.wiley.com/doi/pdfdirect/10.1002/cpbi.87

Good luck,
Peter Beerli Tallahassee, Fall 2021

Contents

Introduction

The program MIGRATE *estimates population size, migration, population splitting parameters using genetic/genomic data.*

For many purposes in biology, we need to know the effective population size of a population and also how well populations interact with other populations. There are essentially two very different approaches to getting such information: a behavioral or ecological approach that monitors individuals in a focus population and recognizes residents and newcomers. Often individuals are marked with tags or other means (banding in birds, toe clipping in amphibians, and more recently inserting magnetic tags under the skin of animals). Such approaches are complex with large populations, or a small number of immigrants, or species with a hidden lifestyle.

Since 1960 an alternative approach has been used. This approach uses an individual's genetic makeup as a tag and measures similarities (or differentials) among groups of individuals. This work led to estimators such as F_{ST}, which indicate how isolated populations are from each other and several other measures that are based on allele frequencies within populations or individuals. These methods are most often based on simple population models that Sewall Wright and Ronald Fisher invented. The most common applications used the Wright-Fisher population model that assumes that the population does not grow or shrink, that every individual has the same chance to reproduce, and that adults are replaced by their offspring every generation. Interestingly, this simple model was (and is) amazingly stable. Even applications to species where such a model seems outlandish (Elephants, humans, etc.) allowed considerable insight into the history of populations. Unfortunately, practitioners are still using these methods despite significant advances in population genetic theory. Problematic issues with these allele-frequency approaches primarily stem from the fact that the assumptions of symmetric immigration rates and equal population sizes need to be fulfilled (*Beerli*, 2004).

Recent approaches based on the coalescent (*Kingman*, 2000b) allow better formulations of an explicit probabilistic model that can handle different immigration rates and different population sizes and additional complications, such as recombination, population splitting. These programs come in two classes: site frequency-based method and full probabilistic coalescent-based approaches. The site frequency methods are fast with few individuals and usually force the infinite sites model. The computer program MIGRATE belongs to the class of full coalescent-based probabilistic model.

MIGRATE in its most simple form can only handle population sizes, immigration rates, and some forms of population splittings; therefore may not be suitable for all datasets. But often, it may help to decide what to do next, despite potential problems with assumption violation (*Beerli*, 2009).

This manual describes the program MIGRATE, its benefits, but also its shortcomings. You will learn in detail about how to use the program and what options are available. This manual is only a start, I suggest that you subscribe to the `migrate-support@googlegroups.com` and participate in the community that uses MIGRATE. Tutorials are also available on the MIGRATE website.

1 A quick overview of the calculations and specification of the parameters

A short overview of the math that is used by the program MIGRATE. *If you want to treat* MIGRATE *as a black box, then skip to the section on* ***parameter definitions****.*

The program MIGRATE infers population genetic parameters from genetic data. Essentially we want to find the Bayesian posterior probability density of parameters $\mathcal{P}$ of a particular model given the Data $\mathcal{D}$:

$$\mathrm{p}(\mathcal{P}|\mathcal{D}).$$

This posterior probability density of the population genetics parameters $\mathcal{P}$, such as population sizes or migration rates, can be calculated in principle by integrating over all possible relationships $\mathcal{G}$ of the sample data $\mathcal{D}$ using an expansion of the coalescent theory (*Kingman*, 1982b,a, 2000a) which includes migration (*Hudson*, 1991; *Nath* and *Griffiths*, 1993; *Notohara*, 1990) and/or population splitting (for example, *Nielsen*, 1998).

$$\mathrm{p}(\mathcal{P}|\mathcal{D}) = \frac{\mathrm{p}(\mathcal{P},\mathcal{D})}{\mathrm{p}(\mathcal{D})} \tag{1.1}$$

which is equivalent to Bayes formula:

$$\mathrm{p}(\mathcal{P}|\mathcal{D}) = \frac{\mathrm{p}(\mathcal{P})\mathrm{p}(\mathcal{D}|\mathcal{P})}{\mathrm{p}(\mathcal{D})}. \tag{1.2}$$

The famous Bayes formula is very general and discusses little what needs to be done in detail, but we want to estimate the probability of particular model parameter values for a given dataset. This posterior probability density depends on the likelihood of the data given the parameters and its priors scaled by the integral over all parameters. the denominator is a simple scaler to make the left hand side a proper probability. the likelihood is problematic because for our problem we need to relate the data (genetic material) with a model about populations. We can achieve that by using random genealogies and weight them how well they fit the data. We simple could consider all genealogies of our sampled data and integrate over them and use

$$\mathrm{p}(\mathcal{P}|\mathcal{D}) = \frac{\mathrm{p}(\mathcal{P})\int_G \mathrm{p}(G|\mathcal{P})\mathrm{p}(\mathcal{D}|G)dG}{\mathrm{p}(\mathcal{D})}. \tag{1.3}$$

The likelihood calculations is the most problematic because of the integration over genealogies

$$\mathrm{L}(\mathcal{P}) = \mathrm{p}(\mathcal{D}|\mathcal{P}) = \int_G \mathrm{p}(G|\mathcal{P})\mathrm{p}(\mathcal{D}|G)dG. \tag{1.4}$$

The key issue is that there are huge number if different topologies, each with continuous branch lengths; it is a sum over all possible labeled histories and integrals over all possible branch lengths b_i

$$\mathrm{L}(\mathcal{P}) = \sum_T \int_{b_1} ... \int_{b_k} \mathrm{p}(T, \underline{b}|\Theta)\mathrm{p}(\mathcal{D}|T, \underline{b})db_1...db_k. \tag{1.5}$$

Older versions of MIGRATE than version 4.0 could use both approaches to estimate the parameters. It became a major burden updating the program to maintain both likelihood and Bayesian inference, so that I decided to strip out the likelihood material and focus on the Bayesian approach, which is often easier to code and maintain. One of the major headaches with the likelihood approach was the maximization of the likelihood function that became more and more complicated with more parameters, this maximization is not needed in the Bayesian approach, although we still need to calculate likelihoods.

1.1 How is it actually calculated?

The above formulae do not really reflect the actual calculations, because there are many complicating factors, for example how many parameters are there really? How to specify the prior distribution? How to calculate tree changes or parameter changes? What to do with multiple loci? MIGRATE uses Markov chain Monte Carlo to approximate the posterior density distribution of the parameters, we only collect the genealogies with a specific option, but usually think of genealogies as nuisance variables. We integrate over all genealogies to infer the parameter values of the model. The basic steps in the code are described in algorithm 1 The most interesting parts are the options definitions and the menu and how the MCMC is executed. The options and menu are discussed in their own chapters. The MCMC step is explained in the section *1.3 Bayesian Inference*.

1.2 Parameter specification and definition

MIGRATE needs a population model and this model is framed in parameters that describe population sizes and geneflow between populations. Specifically, MIGRATE estimates migration rates, population growth, a fractional coalescent parameter α, population splitting times, and effective population sizes of 1 to many populations using genetic data (Fig 1). The parameters to estimate are

$$\mathcal{P} = \left(\underline{\Theta} \quad \underline{g} \quad \underline{\alpha} \quad \underline{\mathcal{M}} \quad \underline{\Delta} \quad \underline{S_\Delta}\right), \tag{1.6}$$

Data: sequence/msat/SNP datafile, options file
Result: output (text and PDF format) describing the posterior densities of the parameters
Read the options file if present;
Display the menu, let user choose run parameters;
Read the data;
while *there are still loci to work on* **do**
    Create a random genealogy ;
    Run the MCMC;
    Store the intermediate results;
end
Collect all intermediate results;
Print the output files;

Algorithm 1: MIGRATE executes this sequence in a single-CPU computer. On a computer cluster the while look can be run in parallel using the MPI interface.

where $\underline{\Theta}$ are the mutation-scaled population sizes of each subpopulation, using $\Theta = xN_e\mu$, where x is the inheritance scalar, which is for diploids 4 and for haploids 2; N_e is the effective population size ; and μ is the mutation rate per site and per generation with DNA/RNA data, for microsatellite data μ is for the whole locus per generation. The growth rates $\underline{g}$ are the mutation-scaled exponential growth rates of the populations assuming that the reported Θ is of today. The $\underline{\alpha}$ is a measure of the variability of the potential of parents to have offspring, replacing the Kingman coalescent with the fractional coalescent. The $\underline{\mathcal{M}}$ are the mutation-scaled immigration rates which are $m_{i\to j}/\mu$, $m_{i\to j}$ is the rate of immigration into population j from i per generation. The direction is interpreted in a standard population genetics way: an individual is in population i at time z and then either its gametes or itself arrives in population j at time $z+1$. The theory does not really consider the emigrants (similarly we do not worry about the many gametes that we do not choose between generations, only the ones that make into the next generation or, here, into the other population or stay are relevant. The divergence parameters $\Delta_{i\to j}$ and $S_{\Delta,i\to j}$ are the parameters of a distribution (commonly a normal) that express the divergence time and its standard deviation. The divergence time is the time where a descendent j splits off of the ancestor i, the ancestor i can but need not to be a sampled population. The divergence time is similarly scaled as the other parameters and MIGRATE estimates a mutation-scaled divergence time. The time is measured in generation times the mutation rate per site and generation. For comparison to other software you may want to divide the divergence time by the sum of all scaled population sizes to scale the time in coalescent units. This works well when compared to the simulation software MS (*Hudson*, 2002).

1.2.1 Units of the parameters used in migrate

A short section about the parameter units used in MIGRATE for details look at the individual parameter descriptions.

Parameter	Equivalent	Units	Other	Explanation
Θ	$xN_e\mu$	$\# \times \mu$	$N_e = \frac{\Theta}{x\mu}$	N_e: effective population size; x: inheritance scalar, 4 is for diploids, 2 for haploids, etc.; μ: mutation rate per generation, for DNA/RNA data it is per site and for microsatellite of allozyme data it is per locus
g	r/μ	$\# \times \mu$	$r = g\mu$	r:population growth rate r per generation; μ: mutation rate per generation, for DNA/RNA data it is per site and for microsatellite of allozyme data it is per locus
α		-		α has a range of 0 to 1; a value of 1 is equivalent to the exponential waiting times for coalescences as Kingman's coalescent.
$\mathcal{M}$	$\frac{m}{\mu}$	rate/μ	$xNm = \Theta M$	m: immigration rate per generation; μ: mutation rate per generation, for DNA/RNA data it is per site and for microsatellite of allozyme data it is per locus
Δ	gen.$\times\mu$	$\# \times \mu$	$t = \Delta/\mu$	gen.: generations before today; μ: mutation rate per generation, for DNA/RNA data it is per site and for microsatellite of allozyme data it is per locus; t: is time in generations
σ_Δ	gen.$\times\mu$	$\# \times \mu$	$t = \sigma_\Delta/\mu$	Standard deviation of the divergence time; gen.: generations before today; μ: mutation rate per generation, for DNA/RNA data it is per site and for microsatellite of allozyme data it is per locus; σ_t: is the standard deviation of the time in generations

1.2.2 Mutation-scaled population sizes

$$\underline{\Theta} = \begin{pmatrix} \Theta_1 & \Theta_2 & \dots & \Theta_n \end{pmatrix} \tag{1.7}$$

where each $\Theta_i = xN_e\mu$ with μ that is the mutation rate per generation and with x that is a multiplier that depends on the ploidy and inheritance of the data, for nuclear data it $x = 4$, for haploid data it is $x = 2$, and for mtDNA in vertebrates with female-only transmission, no sex-change during its life, and a sex ratio of 1:1, it is $x = 1$. Life history is important, for example some fish species, such as Grouper, change sex in their lifetime and therefore all individuals can transmit mtDNA resulting in having $x \simeq 2$ and not $x = 1$.

1.2.3 Relative scale α for the offspring variance

Using an $0 < \alpha < 1$ introduces the *fractional coalescent Mashayekhi* and *Beerli* (2019), this allows to discuss environmental effects on the number of offsprings and may be used to look at adaptation. We are currently funded by a NSF DBI grant to research this topic (2021-2024). [to come]

1.2.4 Mutation-scaled exponential growth rates

[to come]

1.2.5 Mutation-scaled immigration rates

$$\underline{\mathcal{M}} = \begin{pmatrix} - & \mathcal{M}_{2\to1} & \mathcal{M}_{3\to1} & \dots & \mathcal{M}_{n\to1} \\ \mathcal{M}_{1\to2} & - & \mathcal{M}_{3\to2} & \dots & \mathcal{M}_{n\to2} \\ \dots & \dots & \dots & \dots & \dots \\ \mathcal{M}_{1\to n} & \dots & \dots & \mathcal{M}_{(n-1)\to n} & - \end{pmatrix} \tag{1.8}$$

which is the immigration rate per generation m divided by the mutation rate per generation μ, it is a measure of how much more important immigration is over mutation to bring new variants into the population. The traditional number of immigrants per generation $xN_im_{j\to i}$ is $\Theta_i\mathcal{M}_{j\to i} = xN_i\mu \times m_{j\to i}/\mu = xN_im_{j\to i}$. The mutation rate μ is per site and generation for DNA data and per locus and generation for microsatellite or allelic data. If you compare results of different programs make sure that you understand what are the units of μ, oftentimes it μ per locus even with DNA data!

There seems to be considerable confusion about migration directions in the recent literature. When Sewall Wright discussed migration, he considered immigration rate into a population, and since we do not observe immigrants directly, he also assumed that our data represents the offspring of locals and immigrants. In his framework he did not consider emigration and that immigration will happen instantaneous (e.g no delayed influx of gene through seed banks etc). If we see $\mathcal{M}_{2\to1}$ then we always use a forward time perspective: in the past the individual was in population 2 and now is in population 1.

The immigration parameter in MIGRATE is a longtime average over the genealogy of the individuals in the sample. This works well for populations that fluctuate around a value, but even with mildly growing populations this works fine, only with strongly growing populations one may get underestimates using MIGRATE.

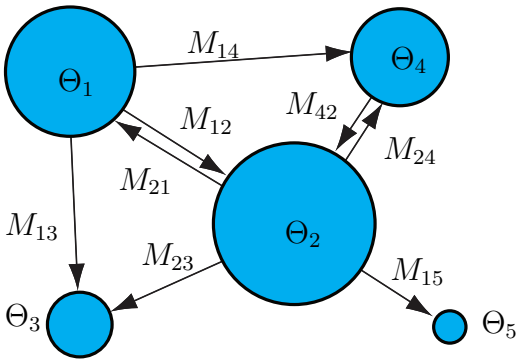


Figure 1.1: Populations exchanging migrants with rate $m_{j\to i}$ per generations and with size N_e. The parameters are scaled by mutation rate μ which is with sequence data per site per generation. The estimated parameters are therefore: Θ_i which is $xN_e^{(i)}\mu$ and $\mathcal{M}_i$ which is m_i/μ, the migration estimate is often also expressed as xNm which is just $\Theta\mathcal{M}$, x is the inheritance parameter and depends on the data, commonly 4 for nuclear data, and 1 for mtDNA data. The example model is not a complete (full) model because some migration routes are not estimated and set to zero.

1.2.6 Mutation-scaled population splitting times Δ and standard deviations S_Δ

The population splitting times are model differently to other programs that all use the model of IM (*Hey*, 2010). Our new model was described by *Beerli* et al. (2022). We treat population splitting events as individual events on a lineage. Looking backward in time, a lineage ℓ currently in population κ is at risk of being in a different population either by migration with rate $\mathcal{M}$ or into another population by a splitting event. The Splitting event is proposed during the MCMC run using a hazard function of the splitting time distribution, for simplicity we use a truncated Normal distribution with mean Δ and standard deviation S_Δ. Drawing these splitting events using a hazard function is equivalent to the coalescent or migration event which are also hazard functions. During the MCMC runs many different splitting events will be proposed from the Normal distribution with Δ_j and S_{Δ_j} for j population splits. The current version needs guidance which populations are merging (looking backwards in time), for an example see the Bayes Factor section. In the parmfile and the menu the setting of the splitting parameters is combined with setting the migration model, for an example see definition of the custom migration matrix.

1.3 Bayesian inference

MIGRATE estimates the parameters using a Bayesian paradigm (see formula 1.3), simulation studies of simple models show that there are few differences with the ML runs, although some combinations of parameters might be be easier to estimate with the Bayesian approach (*Beerli*,

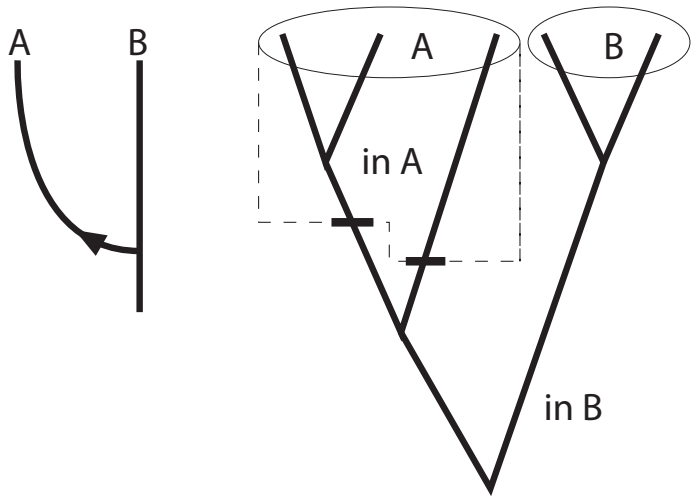


Figure 1.2: Populations splitting, left is the model specification, two populations A and B, A splits off B. Right, detail view for the gene tree. Each individual can split off by itself.

2006). One of the bigger problems with the Likelihood approach is the effort that has to be made to calculate the confidence intervals of the best estimates, default analyses often led to narrow support intervals thus making researchers overconfident about the explanatory power of their data; the use of the prior distribution seems to make a Bayesian approach less vulnerable to this problem. Bayesian inference is commonly based on Markov chain Monte Carlo (MCMC) because we usually cannot integrate the function of interest analytically or by simple numerical approach. MCMC was described first by *Metropolis* et al. (1953) and refined by *Hastings* (1970). For an introduction see *Hammersley* and *Handscomb* (1964) or *Chib* and *Greenberg* (1995), and see *Kuhner* et al. (1995a) for a first application using MCMC in the context of coalescence theory.

Commonly we think of the marginal posterior distribution of each parameter (summing/integrating over all 'nuisance' parameters). We use particular values to drive the MCMC, and in a Bayesian analysis we get them from the Prior distribution, this may help to get good results in situations where the data suggests a very rough landscape of parameter- and tree-space (see Fig. 1.3 for an example with a smooth surface). MIGRATE allows using several different prior distributions, some are more appropriate for you data than others. I often suggest to use the uniform prior distribution because it is simple and shows obvious deficiencies in an analysis very quickly, but it also tends to increase the credibility intervals because it supports very large or very small parameter values equally. This may sound as an odd choice because, for example, we know that the population size of humans never was zero and never was 10^{10}, so a uniform prior distribution with range 0 to 10^{10} does not sound right although such a prior would do fine in an analysis because the data is strong enough to suggest that the posterior probability near a size of zero is close to zero and the probability of a size of 10^{10} is also small.

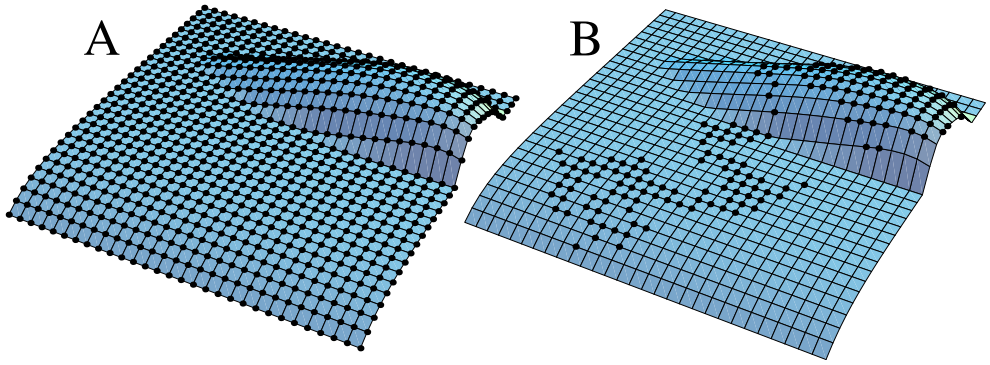


Figure 1.3: (A) On an imaginary, infinite surface that contains every possible genealogy with any possible branchlength we would need to sample every possible genealogy and sum over all these values which is not possible, but genealogies with low probability will not contribute much to the final posterior density, (B) by biasing towards better genealogies we can sample effectively from those genealogies with high contribution to the final posterior and can approximate the likelihood and posterior density (1.4).

1.3.1 Prior distributions

Uniform prior distribution

The parameters have a uniform distribution between a minimal and a maximal value of the parameters. MIGRATE calculates the uniform by using

$$\mathrm{p}(\mathcal{P}_i) = \frac{1}{\mathcal{P}_{max} - \mathcal{P}_{min}} \tag{1.9}$$

it is implemented using a windowing method with window size Δ, that is preferrably around $1/10$ of the whole range.

$$\mathcal{P}_{\mathrm{new}} = \mathcal{P}_{old} + (2\Delta r - 1) \begin{cases} \mathcal{P}_{\mathrm{new}} < \mathcal{P}_{\mathrm{min}} & \mathcal{P}_{\mathrm{min}} + |\mathcal{P}_{\mathrm{min}} - \mathcal{P}_{\mathrm{new}}| \\ \mathcal{P}_{\mathrm{new}} > \mathcal{P}_{\mathrm{max}} & \mathcal{P}_{\mathrm{max}} - |\mathcal{P}_{\mathrm{max}} - \mathcal{P}_{\mathrm{new}}| \end{cases} \tag{1.10}$$

Gamma distribution prior

The truncated gamma distribution has four parameters α, β, minimum a, and maximum b. The gamma prior in MIGRATE is defined by the mean μ, α, a, and b

$$\alpha = \mu/\beta \tag{1.11}$$

$$\beta = \text{minimization of the mean of the truncated gamma and the parameter } \mu \tag{1.12}$$

$$\mathrm{p}(\mathcal{P}_i) = \text{probability of the truncated gamma} \tag{1.13}$$

Exponential prior distribution

The parameters have a exponential distribution, MIGRATE calculates three versions
Simple exponential prior distribution

$$\mathrm{p}(\mathcal{P}_i) = \int_0^\infty exp(-P_i/P_{\mathrm{mean}})/P_{\mathrm{mean}} d\mathcal{P}_i = exp(-P_i/P_{\mathrm{mean}}) \tag{1.14}$$

$$\mathcal{P}_{new} = -\mathcal{P}_{\mathrm{mean}}\ ln(r) \tag{1.15}$$

Exponential prior distribution with fixed window

$$\mathrm{p}(\mathcal{P}_i, \mathcal{P}_{min}, \mathcal{P}_{max}) = \frac{\int_{\mathcal{P}_{min}}^{\mathcal{P}_{max}} exp(-P_i/P_{\mathrm{mean}})/P_{\mathrm{mean}} d\mathcal{P}_i}{exp(-P_{\mathrm{min}}/P_{\mathrm{mean}}) - exp(-P_{\mathrm{max}}/P_{\mathrm{mean}})} \tag{1.16}$$

$$= \frac{exp(-P_{\mathrm{min}}/P_{\mathrm{mean}}) - exp(-P_{\mathrm{x}}/P_{\mathrm{mean}})}{exp(-P_{\mathrm{min}}/P_{\mathrm{mean}}) - exp(-P_{\mathrm{max}}/P_{\mathrm{mean}})} \tag{1.17}$$

$$\mathcal{P}_{\mathrm{new}} = -\mathcal{P}_{\mathrm{mean}}\ ln(\frac{r}{exp(\mathcal{P}_{\mathrm{max}}/\mathcal{P}_{\mathrm{mean}})} - \frac{r-1}{exp(\mathcal{P}_{\mathrm{min}}/\mathcal{P}_{\mathrm{mean}})}); \tag{1.18}$$

2 Files in migrate

MIGRATE *can use many different input methods and output methods, but most of them have a very special purpose, as a minimum you need to supply an input datafile, here called infile.*

There are multiple ways to set up things. MIGRATE can use very different ways to manipulate the data and as a result many different files are needed or produced. Minimally, you need the data file, its default name is *infile*, and MIGRATE produces two files that contains results: the *outfile* (ASCII text file) and a PDF output file that contains the same information (well almost, as you see later). The program produces both formats because for quick checking of results the ASCII file can be opened on the command line or with any text viewer, whereas the PDF file requires a PDF reader, for example for macs Preview.app and for windows NitroPDF; unfortunately modern versions of the standard Adobe Acrobat Reader fail to read the PDF files generated by MIGRATE correctly – I will work on porting my PDF writer to a newer system, but this has low priority and will take a while (Older versions of Adobe Reader work fine!)

2.1 Input files

Filename	Type	Short description	Necessary?	Name changeable
infile	Input	holds you data	YES	Yes
parmfile	Input	holds options	-	Yes*
geofile	Input	holds a (geographic) distance matrix between the populations	-	Yes
datefile	Input	holds the date (default is years) of the sample. When used then you need also to supply a generation time and and a mutation rate per year in the `parmfile` or the Menu.	-**	Yes

* Under Unix the parmfile name ca be given as an argument to the program

** When different sample dates are used then this file is needed

2.1.1 Main input files

infile if this file is not present in the current directory than the program will ask for a data file, and you can give the path to it, you need to type the path, which is for Macintosh and Windows users probably rather uncomfortable. In the **menu** or **parmfile** you can specify an other default name for your datafile.

datefile When the samples came from different years/generations and you believe that this makes a difference (for example you work with HIV or other fast mutating species, or you have ancient DNA), specify the date as the time backward from today (for example years before 2007). With this analysis type, you need to specify a mutation rate in the same units as the dates of the samples.

bayesallfile The `bayesallfile` or `bayesallfile.gz` allows to reuse a previous Bayesian inference run, the parmfile needs to have the option
`recover=YES`
This option cannot be created with the menu. All the other options in the parmfile should be the same as when the `bayesallfile` was created.

2.1.2 Optional input files

parmfile can hold specific menu options, this file and the possible options for the menu are explained in detail in section **menu and parmfile**.

geofile holds the geographic or arbitrary distances between the populations. When this is used then the migration rates are not only scaled by the mutation rate but also by this distance. This allows to detect environmental barriers when we assume that the genetic potential to migrate is the same in all populations; without a barrier the rates should be all the same per distance unit. The format is like a distance file in the PHYLIP package (*Felsenstein*, 2005), but you can use the # as a commentary character.

```
# Example geofile for 3 populations,
# the order of the population must be the same as in the data file
#
  3
Tallahasse0.0 10.0 150.0
St.Marks  10.0 0.0 160.0
Pensacola 150.0 160.0 0.0
```

The example scales the mutation-scaled migration rates by the 'geographic' distance. If the migration rate is linear with the inverse of the distance then the migration rates between all locations will be the same, here we scale the migration rate per distance unit, therefore if we have a range of 0 to 160 miles, the rates are are scaled per mile. As a result migration rates will be all relatively high because a mile is usually not a large distance for vagile species.

2.2 Output files

Filename	Short description	Name changeable
parmfile	holds options, menu can rewrite this file	see menu
outfile	will be created and replace any file with the same name in the same directory	Yes
outfile.pdf	contains the same output as outfile and histograms, you need a PDF viewer to read this file	Yes
bayesfile	contains the histogram data of a Bayesian run (the outfile.pdf used these to generate the posterior distributions.	Yes
bayesallfile	contains the raw data of a Bayesian run, can be run through TRACER when only a single replicate and a single locus is used.	Yes
mighistfile	contain the distribution of migration events over time.	Yes
skylinefile	contains the distribution of the parameter values over time as calculated by using the expected parameter values for a short time intervals.	Yes
treefile	holds genealogies, this file will be created and will replace any file with the same name in the same directory	Yes
logfile	logs the progress information that is displayed onto the screen into a file	Yes

2.2.1 Main output files

Some conbination of the output files are not possible, for example a standard Bayesian run will not fill values into the treefile, etc.

outfile and **outfile.pdf** Somewhere you want to read the results, that is it! The name **outfile** is the default, but can be changed either in the menu or the parmfile. The PDF file contains graphical representation of some of the table and values. Currently, most of the output is represented in the PDF file.

treefile holds all trees, or the best tree(s) for each locus. The likelihood of each tree is given ($\mathrm{p}(\mathcal{D} \mid \mathcal{G})$) in a comment. The programs writes trees with migrations using the Newick format with extensions from the Nexus format. Writing trees to a treefile adds some burden to the program, it will run slower, especially with the option BEST. Parallel runs increase the communication with the master node and therefore may slow down your run.

bayesfile holds the posterior histogram data shown in the PDF files. You can use other program packages like the matplotlib package in python (http://www.python.org), GNUPLOT (http://www.gnuplot.org), or the GMT package (http:www.soest.hawaii.org/gmt3) to recreate the histograms.

bayesallfile holds the raw posterior values for all parameters. This option reduces the memory footprint by writing all intermediate results to disk and then rereads them for summarizing and printing the final results. This file can be also used to independently test whether MIGRATE converged or not using the program TRACER (*Rambaut*, 2007; *Drummond* and *Rambaut*, 2007), MIGRATE uses a simple 1-step Effective sample size (ESS) calculator that may not always be very accurate, although comparison showed that seeing high autocorrelation in MIGRATE means to see high autocorrelation (small ESS) in TRACER.

mighistfile holds the histogram over time of the frequency of migration and coalescence events, with simulated data these plots show typically an exponential decay. When there were changes of parameters over time then the data will enforce different patterns, that can be used to discuss the results.

skylinefile holds the averages of the expected parameter values at specific times. These plots are similar to the skyline plot reported in BEAST (*Drummond* et al., 2005), although their derivation is an extension of the original skyline plots of (*Strimmer* and *Pybus*, 2001). MIGRATE reports changes of population sizes and migration rates over time and summarizes over multiple loci.

3 Data models

A short overview of the different datatypes and how multiple loci are summarized.

MIGRATE allows for several different input data types, such as electrophoretic marker data, microsatellite data, sequence data as stretches of contiguous sites and as single nucleotide polymorphisms.

3.1 Infinite allele model

This assumes that every mutation will result in a new allele, there is no back mutation (Fig. 3.1). This model is used in all current implementations of electrophoretic data analyses packages (Biosys-1, GDA among others) and perhaps is appropriate for this kind of data. MIGRATE is calculating the parameters for each locus independently and summarizes at the end taking the likelihood surfaces or Posterior distributions of each locus into account.

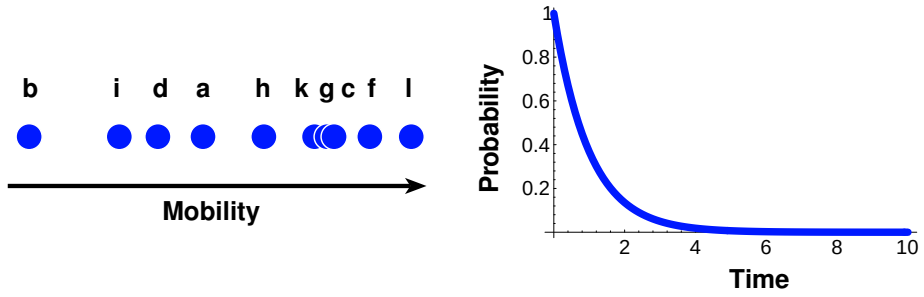


Figure 3.1: Left: Mobility of electrophoretic marker in an electric field. the alleles a,b,c,.. describe a possible sequence of mutation, their mobility is not correlated with the mutational history. Right: The probability that a given allele is not mutating during some time, this is a simple exponential relationship.

3.2 Microsatellite model

3.2.1 Ladder model

The ladder model was invented by citeohta:1973:amm and *Kimura* and *Ohta* (1978) for electrophoretic markers, but was not as good as expected in describing real electrophoretic alleles. For microsatellites this model seems much more appropriate cite[e.g.][]valdes:1993:afm, but see *Di, Rienzo A* et al. (1994), here obviously change happens mostly by slippage of the two DNA strands creating with higher probability a new allele which is only 1 step apart from the old than one which 2 steps apart (Fig. 3.2).

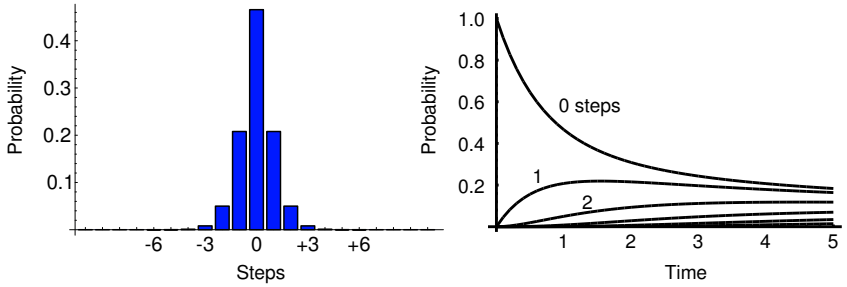


Figure 3.2: Left: Number of repeat changes of a microsatellite marker. The probability to have a slippage of only one repeat is higher than the slippage of more than one repeat, in a given time, here time=0.1. Right: The probability that a change of 0,1,2,.. steps is occurring during some time.

3.2.2 Brownian motion approximation to the ladder model

This replaces the discrete stepwise mutation model with a continuous Brownian motion model The results are very similar to the exact stepwise mutation model, but the parameter estimation is several times faster. This is a crude approximation that has some difficulties when the dataset is not very variable because it uses a cutoff for the the probability that there is no change between two points on a branch, during a time of x the Brownian motion approximation replaces discrete jumps between repeats with a continuous approximation.

3.3 DNA/RNA model

3.3.1 Sequence model

Migrate implements the sequence model of Felsenstein (1984) available in `dnaml` (PHYLIP 4.0, Felsenstein 1997)(Fig. 3.4). The transition probabilities were published by Kishino and

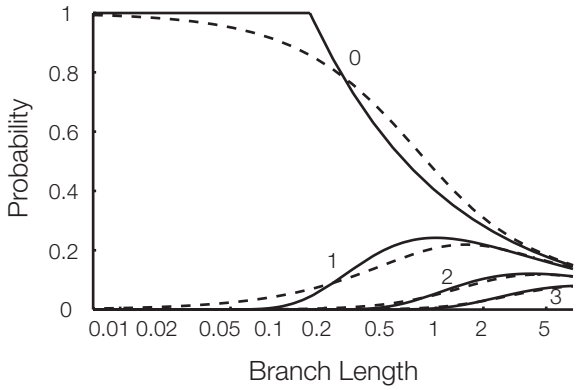


Figure 3.3: Comparison of stepwise mutation model with Brownian motion approximation (dashed lines). The numbers 0, 1, 2, 3, 4 are the number of steps. The Brownian motion approximation for no change is truncated at 1. With steps of more than 4 there is no differences between the stepwise model and the approximation. X-Axis is in $\log_{10}$

Hasegawa (1989). MIGRATE does not allow for recombination within a locus and therefore may over-estimate variability because of recombination, but this bias is not explored well, if in doubt I suggest to try to run MIGRATE, simulated high recombination rate data leads to difficulties with convergence. Applications of recombination tests beforehand may work well, but most of these recombination recognition program use the 4-gamete test that is based on the infinite sites model and therefore will overestimate the importance of recombination.
Like `dnaml`, MIGRATE also allows for different evolutionary rates, mutation categories and autocorrelation, although any use of these additional features can slow done to program to a crawl, but this may change in the future as computers double their speed roughly every 2 years.

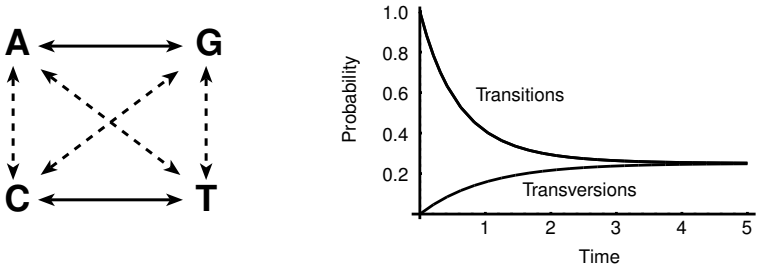


Figure 3.4: Left: Sequence mutation model. Transitions are are shown in black lines, transversion are shown with dotted lines. Right: The probability that a transition or transversion is occurring during some time. The shown graph uses equal base frequencies, but the used model does not need this restriction.

3.3.2 Single nucleotide polymorphism data (SNP)

We use a rather restrictive ascertainment models for SNPs *Kuhner* et al. (2000). A better approach than using SNPs is the use of short reads which may or many not contain SNPs. I find that SNPs are an inferior datatype because commonly researchers are adding criteria such as a minor SNP allele must occur at a frequency higher than x, and singletons are excluded etc.

1. We have found ALL variable sites and use them even if there are only a few members of another alleles present. In principal it is as you would sequence a stretch of DNA and then remove the invariant sites. Each stretch is treated as completely linked. You can combine many of such "loci" to improve your estimates.

This is certainly not how people develop SNPs, but currently the closest we can come up with. The SNP coding is otherwise exactly the same as the coding for DNA data.

3.4 Combining multiple loci

MIGRATE calculates all loci estimates independently, the multi-locus estimate is not a simple average over all loci, but takes into account the likelihood or posterior distribution for each parameter at each locus. Loci with flat likelihood curves or flat posteriors will not contribute as much as those with strongly peaked distributions. MIGRATE offers different treatment in the mutation menu of the parameter menu for the mutation rate among loci:

```
Mutation rate among loci
(C)onstant    All loci have the same mutation rate [default]
(E)stimate    Mutation rate
(V)arying     Mutation rates are different among loci [user input]
(R)elative    Mutation rates estimated from data
```

The **Constant** assumption forces each locus to have the same identical mutation rate. Try this first, because it is the least complicated and most often gives fine results. The **Estimate** is the most difficult and needs dated samples, without sample dates do not use this option. The last two options, **Varying and Relative**, are probably the best ones to try if you really need variable mutation rates. When you know the relative differences of mutation rates in your data, you can specify them. Alternatively, let MIGRATE estimate the relative mutation rate using your data. For sequences MIGRATE calculates a simple Watterson's effective population size estimate over all samples and for each locus and then uses that to calculate a relative mutation rate. With microsatellites and allozyme data MIGRATE counts the number of alleles and uses those as a measure of relative mutation rates. The mean of these rates is 1.0.

4 Data file specification

In detail specification of the data format, without reading and using this information your analysis will most likely faulty.

The data needs to be in a certain form; for us, the following formats were most convenient, but you need to edit your data into this form. There are some programs that can write MIGRATE files, for example the program MSAANALYZER (*Dieringer* and *Schlotterer*, 2003) that can generate MIGRATE datafiles from excel spread sheets for microsatellite data. The following formats are discussed in detail:

- DNA or RNA sequence data (single locus and multilocus)
- Single nucleotide polymorphism data (two formats)
- Microsatellite marker data (MIGRATE uses **REPEATNUMBERS by DEFAULT!!!!!!!**)
- Allozyme data (or other infinite allele mutation model marker)

General data format

Syntax: a token is either a word, a collection of words, or a character or a number:

$< token >$ the token between the the "angle-brackets" is obligatory

$[token]$ in square brackets are optional.

$\{token\}$ are obligatory for some

$< token1|token2 >$ choose one of the token kind of data. If this is too abstract, look at the examples further down.

A range of numbers in a "word" token as in `<individual1 10-10>` means that this token needs to be 10 characters long. The characters for any word token can normally include special characters, punctuation, and spaces, the token for the individual name "`Ind1 02 @`" is legal. An explanation of the individual parts follows at the end of this

section. The most common data file for allozyme data or microsatellite data would look like this (examples follow):

```
<Number of populations> <number of loci> \{delimiter between alleles\} \[project title 0
{\#@M <msat1-repeatlength> <msat2-repeatlength> .....}
<Number of individuals> <title for population 0-79>
<Individual 1 10-10> <data>
<Individual 2 10-10> <data>
....
<Number of individuals> <title for population 0-79>
<Individual 1 10-10> <data>
<Individual2 10-10> <data>
....
```

The delimiter is needed for microsatellite data and the project title is optional. The line starting with `#@M` is not necessary when the data consists of allozyme data or microsat repeat numbers. The line allows to automatically calculate the number of repeats from the fragment length. The `data` will be described in the following sections. **The population name must start with a alphabetical character (not a number). The individual name has to be 10 characters by default** (same as in PHYLIP), but can be changed to another constant in the parmfile, even to a length of 0. [This is one of the most common errors, make sure that your individual names are 10 characters, it does not matter whether they are all alphanumeric, spaces are fine]

For sequences or SNPs, the syntax is slightly different, the following case is for non-interleaved sequence data.

```
<Number of populations> <number of loci> [project title 0-79]
<number of sites for locus1> <number of sites for locus 2> ...
<Number of individuals locus1> <\#ind locus 2> ... <\#ind loc n> <title for population 0
<Individual 1 10-10> <data locus 1>
<Individual 2 10-10> <data locus 1>
....
<Individual 1 10-10> <data locus 2>
<Individual 2 10-10> <data locus 2>
....
<Number of individuals>  <\#ind locus 2> ... <\#ind loc n>  <title for population 0-79>
<Individual 1 10-10> <data locus 1>
<Individual 2 10-10> <data locus 1>
....
<Individual 1 10-10> <data locus 2>
<Individual 2 10-10> <data locus 2>
....
```

For each locus one can give different number of individuals, if there is only one number then the program assumes that all loci have the same number of individuals. If there a fewer numbers than loci the last number will substitute for the number of individuals at the other loci. It is important that the population name does not start with a number!

MIGRATE version older than 4.0 supported interleaved sequence formats, Ivstopped supporting this and have therefore removed its description, reformat your data to a non-interleaved format before you translate it into the MIGRATE format. I typically use PAUP* *Swofford* (2003) to export a non-interleaved PHYLIP formatted datafile and use that to change into the MIGRATE format.

A data type called **HapMap** is available for SNP data that allows less cumbersome input of SNP data than old versions of MIGRATE, but see current imput format. You still can use a single site as a locus (SNP), but with many loci this will be difficult to manage. The HapMap data type uses this format:

```
<Number of populations> <number of loci> [project title 0-79]
<Any Number> <title for population 0-79>
<Position on chromosome locus1> <TAB><allele><TAB><number><TAB><allele><TAB><number><TAB
<Position on chromosome locus2> <TAB><allele><TAB><number><TAB><allele><TAB><number><TAB
....
<Any Number>  <title for population 0-79>
<Position on chromosome locus1> <TAB><allele><TAB><number><TAB><allele><TAB><number><TAB
<Position on chromosome locus2> <TAB><allele><TAB><number><TAB><allele><TAB><number><TAB
....
```

The current format assumes that each SNP is biallelic. <allele> contain the nucleotide and the <number> contains the number of individuals with that specific allele, the total number is the sum of both, and is currently not necessary, but I may use this later to accommodate slight extension of this format, currently the total number is read from the program but not further used. This format will extend to more useful analyses that take into account the position on the chromosome, but this is currently not used.

Summary of the individual tokens

<Number of populations> Number of populations. Range: $1, 2, 3, \ldots, n$ where n is a smallish number, remember that the default MIGRATE run estimates n^2 parameters.

<Number of Loci> Number of unlinked loci. Range: $1, 2, 3, \ldots, \ell$ where ℓ can be a large number.

<Delimiter> can be any character that does not occur in some other function in the data set, examples: @ , . /

<Number of individuals> Number of individuals within one population. Range: $1, 2, 3, \ldots, m$. For exploring MIGRATE I suggest to use around 10 to 20 individuals, much less (for example 1 or 2) or more (for example 1000) will make the analysis more difficult and need more experience and patience.

<Title for population> Title for the population, the first letter must **not be a Number**!

<Individual> Remember that the default for individual names needs 10

characters. Ideally, each individual name is unique and the first letter of the individuals name represents a code for the populations, for example $0, 1, 2, ..$

<Data> See examples for the different data types.

<Number of sites> Number of linked sites. Range: $1, 2, 3, \ldots, S$

<Position on Chromosome> Location on genome measured in sites [not functional yet]

<Allele> For SNP data this is one of the nucleotides: A, C, G, T.

<Number> For SNP data this is the number of <Allel> at that specific site in the sample.

<Total> For SNP data this is the total number of samples at the specific site.

4.0.1 Examples of the different data types

The examples in this section look like real data, but they are only for the demonstration of the syntax, if you try run this "data" it will deliver often very strange values, I have added a "usable" test set of simulated data in the examples directory, see the file examples/README for more information.

Allozyme data (infinite allele model)

The data is given in genotypes, any printable character with ASCII code bigger than 33 ('!') and smaller than 128 can be used. '?' is reserved for missing data. You can use multi-character coding when you use a delimiter (see the examples for microsatellites). If there is enough interest I can work on a input using gene frequencies, although I prefer to work on more interesting things than adjusting input files.

Most simple example with a single locus, 2 population and 5 total individuals.

```
 2 1 Migration rates between two Turkish frog populations
3 Akcapinar (between Marmaris and Adana)
PB1058    ab
PB1059    ab
PB1060    b?
2 Ezine (between Selcuk and Dardanelles)
PB16843   ab
PB16844   bb
```

Example with 2 populations and 11 loci and with 3 and 2 individuals per population, respectively (this data set is only an example of syntax, analyzing this dataset would not make much sense).

```
 2 11 Migration rates between two Turkish frog populations
3 Akcapinar (between Marmaris and Adana)
```

```
PB1058    ee bb ab bb bb aa aa bb ?? cc aa
PB1059    ee bb ab bb bb aa aa bb bb cc aa
PB1060    ee bb b? bb ab aa aa bb bb cc aa
2 Ezine (between Selcuk and Dardanelles)
PB16843   ee bb ab bb aa aa aa cc bb cc aa
PB16844   ee bb bb bb ab aa aa cc bb cc aa
```

Same example, but with a different syntax that allows multicharacter allele names (see last locus!). The delimiter is specified as the third parameter in the first line, the delimiter cannot be a standard alphabet character.

```
 2 11 / Migration rates between two Turkish frog populations
3 Akcapinar (between Marmaris and Adana)
PB1058    e/e b/b a/b b/b b/b a/a a/a b/b ?/? c/c Rs/Rf
PB1059    e/e b/b a/b b/b b/b a/a a/a b/b b/b c/c Rs/Rs
PB1060    e/e b/b b/? b/b a/b a/a a/a b/b b/b c/c Rs/Rs
2 Ezine (between Selcuk and Dardanelles)
PB16843   e/e b/b a/b b/b a/a a/a a/a c/c b/b c/c Rf/Rf
PB16844   e/e b/b b/b b/b a/b a/a a/a c/c b/b c/c Rf/Rs
```

4.0.2 Microsatellite data

DEFAULT INPUT SYNTAX

The third argument on the first line has to be a delimiter character, for example a **"."**. The data is given in genotypes. Each individual has two alleles. Alleles are coded as **REPEAT NUMBERS**, so for example your sequence

```
Flanking    msat    Flanking
region              region
------------------------
ACCTATAGCACACACACACAAATGCGA      6 CA repeats
ACCTATAGCACACACACA--AATGCGA      5 CA repeats
```

contains a microsatellite with 6 repeats. And if with a homozygote individual it needs to be coded as 6.6 or 06.06, where the "," is the delimiter. '?' is reserved for missing data.

Example:

```
 2 3 . Rana lessonae: Seeruecken versus Tal
2   Riedtli near Guendelhart-Hoerhausen
0         6.5 37.31 18.18
0         6.6 37.33 18.16
2  Tal near Steckborn
1         4.5 35.? 18.18
1         4.4 35.31 20.18
```

FRAGMENT LENGTH INPUT SYNTAX

Earlier version of The third argument on the first line has to be a delimiter character, for example a **"."**. The data is given in fragmentlength. Each individual has two alleles.

Alleles are coded as **FRAGMENTLENGTH**, so for example your sequence

```
Flanking    msat    Flanking
region              region
------------------------
ACCTATAGCACACACACACAAATGCGA      27 sites total length
ACCTATAGCACACACACA--AATGCGA      25 sites total length
```

contains a microsatellite with 6 repeats, but you only have measures of the total length, here for the first allele there are 27 sites and the second allele there are 25 sites. This format needs an additional line to tell MIGRATE that we use fragment length and that MIGRATE needs to do the translation to repeat numbers, inspect closely the line that starts with #@M in the example below. The #@M tells the program that here comes a definition of the microsatellite repeats, and the numbers force MIGRATE to assume that the loci are dinucleotide repeats (2) , ot trinuleotide with 3 or tetranulceotides with 4 nucleotides per repeat, and so forth.

And if with a homozygote individual it needs to be coded as 25.25 or 025.025, where the "." is the delimiter. A heterozygote would read 25.27, for example. '?' is reserved for missing data.

Example:

```
  2 3 . Rana lessonae: Seeruecken versus Tal
\#@M 2 2 2
2   Riedtli near Guendelhart-Hoerhausen
0         25.27 137.131 218.218
0         27.27  218.216
2  Tal near Steckborn
1         23.25 135.? 218.218
1         23.23 135.131 220.218
```

4.0.3 Sequence data

The sequence data format has received a face-lift: two new formats are now allowed (1) the old format described below and (2) the new format that is more appropriate for genomic type data.

After the individual name follows the base sequence of that species, each character being one of the letters A, B, C, D, G, H, K, M, N, O, R, S, T, U, V, W, X, Y, ?, or - . Blanks will be ignored, and so will numerical digits. This allows GENEBANK and EMBL sequence entries to be read with minimum editing. These characters can be either upper or lower case. The algorithms convert all input characters to upper case. The characters constitute the IUPAC (IUB) nucleic acid code plus some slight extensions (Table 4.1). They enable input of nucleic acid sequences taking full account of any ambiguities in the sequence.

Table 4.1: IUPAC (IUB) convention for naming nucleotide sites and ambiguous sites

Symbol	Meaning		Symbol	Meaning	
A	Adenine		B	not A	(C or G or T)
G	Guanine		D	not C	(A or G or T)
C	Cytosine		H	not G	(A or C or T)
T	Thymine		V	not T	(A or C or G)
U	Uracil		X,N,?	unknown	(A or C or G or T)
Y	pYrimidine	(C or T)	O	deletion	
R	puRine	(A or G)	-	deletion	
W	"Weak"	(A or T)			
S	"Strong"	(C or G)			
K	"Keto"	(T or G)			
M	"aMino"	(C or A)			

Most simple example with 1 population and a DNA-locus with 50 basepairs.

```
   1 1 Make believe data set using simulated data (1 locus)
50
3  Tallahassee
Peter     ACACCCAACACGGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
Donald    ACACAAAACACGGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
Christian ATACCCAGCACGGCCGGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
```

```
   1 1 Make believe data set using simulated data (1 locus) OLDFORMAT
50
3  Tallahassee
Peter     ACACCCAACACGGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
Donald    ACACAAAACACGGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
Christian ATACCCAGCACGGCCGGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
```

The new format looks very similar in its simplest version

```
   1 1 Make believe data set using simulated data (1 locus) NEWFORMAT
(s50)
3  Tallahassee
Peter     ACACCCAACACGGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
Donald    ACACAAAACACGGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
Christian ATACCCAGCACGGCCGGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
```

Same example, but now with 2 population and a single DNA-locus with 50 basepairs.

```
   2 1 Make believe data set using simulated data (1 locus) OLDFORMAT
50
3  Tallahassee
Peter     ACACCCAACACGGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
Donald    ACACAAAACACGGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
Christian ATACCCAGCACGGCCGGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
```

```
3  St. Marks
Lucrezia  ACACCCAACACGGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
Isabel    ACACAAAACACGGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
Yasmine   ATACCCAGCACGGCCGGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
```

In the new format this still looks very similar to the old format except for the line that contains the number of sites, the simple number of sites is replaced by the data type 's' and the parenthesis specifies that the locus is unlinked

More complicated example with 2 population AND with **2 loci**, the sequences are NOT interleaved, I drop the interleaved from because I find it error-prone cumbersome to change and unnecessary.

```
   2 2 Make believe data set using simulated data (2 loci) OLDFORMAT
50 46
3 3   pop1
eis       ACACCCAACACGGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
zwo       ACACAAAACACGGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
drue      ATACCCAGCACGGCCGGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
eis       ACGCGGCGCGCGAACGAAGACCAAATCTTCTTGATCCCCAAGTGTC
zwo       ACGCGGCGCGAGAACGAAGACCAAATCTTCTTGATCCCCAAGTGTC
drue      ACGCGGCGCGAGAACGAAGACCAAATCTTCTTGATCCCCAAGTGTC
2    pop2
vier      CAGCGCGCGTATCGCCCCATGTGGTTCGGCCAAAGAATGGTAGAGCGGAG
fuef      CAGCGCGAGTCTCGCCCCATGGGGTTAGGCCAAATAATGTTAGAGCGGCA
vier      TCGACTAGATCTGCAGCACATACGAGGGTCATGCGTCCCAGATGTG
fuefLoc2  TCGACTAGATATGCAGCAAATACGAGGGGCATGCGTCCCAGATGTG
```

Improved (new) format

The old format asks for the number of sites for each locus on the second line of the datafile and then needs all loci as consecutive blocks within each population. The new format still allows this old style but adds a new format that can take concatenated loci, one line per individual, To mark the new format the number of sites needs to be specified in a modified format, here a few examples:

```
OLDFORMAT 1 locus:  123
NEWFORMAT 1 locus:  (s123)
OLDFORMAT 3 loci:  123 195 2310
NEWFORMAT 3 loci:  (s123) (s195) (s2310)
NEWFORMAT 3 loci:  (s123), (s195), (s2310)
```

The last two examples show some of the difference to the old format, the example with the "," (comma) block the data like the old format, whereas the example just before uses the concatenated scheme, each individual will need 123+195+2310 sites, that are are then separated by the program into 3 independent loci because the "()" syntax suggests that these loci are independent. Other formats like "Brownian motion is specified with (b1),

SNPs are specified with (n1) or 4 linked snps are specified with (n4). For allelic data the old format is preferred and the new format may still break for Brownian motion, stepwise, or allelic models (b, m, a).

Advancement of the new format

The main advantage of the new format allows to give a very large sequence that may or may not be a concatenated list of loci, that then can be split by the specification on the 'sites' line (the second line) in the datafile The new format is triggered when the first character on the sites line is a '(', fro example `(s50)`
marks a single sequence locus with 50 sites `(s20 s30)`
marks two LINKED sequence loci with 20 sites and 30 sites. The two loci are on the same line, for example (s3 s5) looks like this

```
2 2 Make believe data set using simulated data (2 loci) NEWFORMAT
(s3  s5)
3  Tallahassee
Peter     ACA CCCAA
Donald    ACA CAAAA
Christian ATA CCCAG
3  St. Marks
Lucrezia  ACA CCCAA
Isabel    ACA CAAAA
Yasmine   ATA CCCAG
```

`(s20) (s30)` marks two UNLINKED sequence loci with 20 sites and 30 sites, respectively.

The old format that specified earlier as two unlinked loci with `50 46` can now be written as `(s50), (s46)` observe the ',' that specifies that the loci are blocked like in the old format, if both loci would be in the sample line as in the example before, then it reads `(s50) (s46)`

Here are more examples: `(s100) (s50 s40) (s10)`
The first locus has 100 basepairs, the second is a compound out of two linked loci with 50 sites and 40 sites each, and third locus has 10 sites.

Currently MIGRATE is ill equipped to run dataset with large sequences (millions of base pairs) automatically without guidance by the user how to break up these into unlinked blocks. But there are several shortcuts for very large genomic sequences, for example assume that you have sequence data of a chromosome. You could want to run 100 loci with length 500 bp distributed over the whole genomic sequence. This would be possible by `[100o500] (s21000000)` and you will also need to specify on the first line that you have 100 loci. This will take the whole sequence of 21×10^6 sites and extract 100 regularly spaced loci each 500 bp long, the same could be achieved by specifying the location of the locus in the full sequence, using something like:
`(0s500) (210000s500) (420000s500) (630000s500) ....`

Instead of an ordered set of loci one can choose a randomly set of loci using
`[100r500] (s21000000)`
This allows to run different subsets of the data, currently there is no way to use this random site subset to do model comparison because there is no possibility to force the same random set for different runs of MIGRATE.

4.0.4 SNP data

The SNP data uses the same nucleotide nomenclature as the sequence data. and the first format is the same as the sequence data but with only one site for unlinked SNPs or more than one site for linked SNPs see example, the datatype to use for this data is either 'N' for nucleotides or 'H' for HapMap. The very first letter forces as specific data model, if that first position is empty than the parmfile or the menu can specify the data type.

```
# using the old SNP data format
N 2 2 Make believe data set using simulated data (2 population and 2 loci)
1 4
3 3   pop1
ind1      A
ind2      A
ind3      A
ind1      ACAC
ind2      ACAC
ind3      ACGC
2    pop2
ind4      C
ind5      C
ind4      TGGA
ind5      TCGA
```

The HapMap format for the same data set looks like this:

```
# PRELIMINARY use this with care and let me know!
# using the HapMap data format
H 2 2 Make believe data set using simulated data
3  pop1
1         A    3    C    0    3
1000      A    3    T    0    3
1010      C    3    G    0    3
1011      A    2    G    1    3
1015      C    3    A    0    3
2    pop2
1         A    0    C    2    2
1000      A    0    T    2    2
1010      C    1    G    1    2
1011      A    0    G    2    2
1015      C    0    A    2    2
```

5 Menu and Options

Most options can be changed through the textual menu.

You can change the options in the menu (Fig. 5.1) using letters or in submenus numbers. In menu entry `Data type` you need to specify what kind of data you have and according to that type some other menu entries appear, for example: transition/transversion ratio for sequences.

```
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 +   POPULATION SIZE, MIGRATION, DIVERGENCE, ASSIGNMENT, HISTORY  +
 +   Bayesian inference using the structured coalescent           +
 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  PDF output enabled [Letter-size]
  Version 5.0.4(git:v5.0.2-30-g1306745-dirty)   [May-09-2022]
  Program started at   Fri Aug  5 11:04:35 2022


  ================================================================
  MAIN MENU
  ================================================================

  D       Data type currently set to: DNA sequence model
  I       Input/Output formats and Event reporting
  P       Parameters  [start, population model]
  S       Search strategy
  W       Write a parmfile
  Q       Quit the program


  To change the settings type the letter for the menu to change
  Start the program with typing Yes or Y
===>
```

Figure 5.1: Top menu of *Migrate*

Menu options can also be changed in the `parmfile`, but before you do that, become more experienced with the menu and its interaction with the parmfile (make some changes in the menu, save the parmfile, and then check how these changes were translated. Never

ever use an old parmfile from earlier versions to edit by hand, you will miss new options and also potential changes in the parmfile. If you want to use options of an older parmfile, load it into MIGRATE and save it using the menu option, and then manipulate the parmfile with a text editor. **migrate will overwrite currently all user comments added to the parmfile.** All possible options are shown in `parmfile` syntax, but the same items can be changed in the menu as well. All entries in the `parmfile` are not case sensitive and all options can be given with the first letter, e.g. datatype=Allele is equal to datatype=A.

5.1 Data type

If you chose `D` in the main menu then will get the data menu (Fig. 5.2). More options will appear with some choices, for example when you have dated samples you can add a datefile and will also need to specify a mutation rate estimate (Fig. 5.3). These additional options are meaningless without dated samples and should only be used with that type of ancient DNA or virus datasets.

```
  ================================================================
  DATATYPE AND DATA SPECIFIC OPTIONS
  ================================================================

  D   change Datatype, currently:                  DNA sequence model
  1   change Mutation model, currently:                  Jukes Cantor
  2   Haplotyping is turned on:                                    NO
  5   One category of sites?                             One category
  6   Site rate variation                                NO: one rate
  8   Sites weighted?                                              NO
 10   Sequencing error rate?              [0.000 0.000 0.000 0.000]
 13   Inheritance scalar set                                       NO
 14   Pick random subset per population of individuals             NO
 15   Assign individuals [?name in data] to population             NO
 16   Tip date file                    None, all tips a contemporary

  Are the settings correct?
  (Type Y or the number of the entry to change)
===>
```

Figure 5.2: Data menu

To change the data type select `1`, the other numbers show options that are relevant for the actual data type. There are several datatypes such as the following:
datatype=<Allele | Microsatellites | Brownian | Sequences | Nucleotide-polymorphisms | HapMap-SNP | Genealogies >
specifies the datatype used for the analyses, needless to say that if you have the wrong data for the chosen type the program will crash and will produce sometimes very cryptic error messages.

Allele: infinite allele model, suitable for electrophoretic markers, perhaps the "best" guess for codominant markers of which we do not know the mutation model.

Microsatellite: a simple electrophoretic ladder model is used for the change along the branches in genealogy.

Brownian: a Brownian motion approximation to the stepwise mutation model for microsatellites us used (this is **much** faster than exact model, but is not a good approximation if population sizes Θ_i are small (say below 10).

```
  ====================================================================
  DATATYPE AND DATA SPECIFIC OPTIONS
  ====================================================================

  D   change Datatype, currently:                  DNA sequence model
  1   change Mutation model, currently:                     Tamura-Nei
  2   Haplotyping is turned on:                                     NO
  5   One category of sites?                            One category
  6   Site rate variation                            YES:    4 rates
  7   Rates at adjacent sites correlated?    NO, they are independent
  8   Sites weighted?                                               NO
 10   Sequencing error rate?               [0.010 0.010 0.010 0.010]
 13   Inheritance scalar set                                        NO
 14   Pick random subset per population of individuals              NO
 15   Assign individuals [?name in data] to population              NO
 16   Tip date file                                           datefile
 17   Mutation rate per locus and year                0.000000010000
 18   How many generations per year                           0.5000

  Are the settings correct?
  (Type Y or the number of the entry to change)

===>
```

Figure 5.3: Data menu with more options that appear with dated samples, and site rate categories

Sequences: Data are DNA or RNA sequences and the mutation model used is F84, first used by Felsenstein 1984 (actually the same as in `dnaml` (Phylip version 3.5; *Felsenstein*, 1993), a description of this model can be found in *Swofford* et al. (1996).

Nucleotide-polymorphism:[SNP] the data likelihood is corrected for sampling only variable sites. We assume that the a sequence data set was used to find the SNP. It is more efficient to run the full sequence data set.

HapMap-SNP:[SNP] the data likelihood is corrected for sampling only variable sites. We assume that the a sequence data set was used to find the SNP.

Genealogies: Reads the `bayesallfile` (see INPUT/OUTPUT section) of a previous runs, currently this option simply recreates the histogram, this allows the readjust some of the printouts but its usability to create new plots is limited.

Sequence data

If you specified **datatype=Sequence** the following options have some meaning and will show up in the menu (see also details for these options in the main.html and dnaml.html of the PHYLIP distribution

http://evolution.gs.washington.edu/phylip.html)

ttratio=< r1 r2 .....>
you need to specify a transition/transversion ratio, you can give it for each locus in the dataset, if you give fewer values than there are loci, the last ttratio is used for the remaining loci → if you specify just one ratio the same ttratio is used for all loci.

freq-from-data=< Yes | No:freqA freqG freqC freqT>
freq-from-data=Yes calculates the base frequencies from the infile data, this will crash the program if in your data a base is missing, e.g. you try to input only transitions. The frequencies must add up at least to 0.9999.
freq-from-data=No:0.2 0.2 0.3 0.3 Any arbitrary nucleotide frequency can be specified.

sequence-error=$<$ {**VALUE,VALUE,VALUE,VALUE**}|**Estimate:**$1|4>$
The number has to be between 0.00 and 1.00, default is 0.00, which of course is rather far from the truth of about 0.001 (= 1 error in 1000 bases). The values are considered to be error rates for all sites and sequences. One can in principle estimate the error rate (for for all bases or 4 for each of the bases) through MCMC but this may not work well. Examples are
sequence-error={0.002,0.001,0.0004,0.005}
sequence-error=Estimate:1

categories=<Yes | No>
If you specify **Yes** you need a file named " `catfile` in the same directory with the following Syntax: number_of_categories cat1 cat2 cat3 .. categorylabel_for_each_site for each locus, a # in the first column can be used to start a comment-line. This option is very rarely used. Example is for a data set with 2 loci and 20 base pairs each

```
# Example catfile for two loci
# in migrate you can use # as comments
2 1 10         1111111112222222222
5 0.1 2 5 23 3 11111122223333445555
```

rates=< n : r1 r2 r3 ..rn>
by specifying rates a hidden Markov model is used for the sequences *Felsenstein* and *Churchill* (1996), also see the PHYLIP documentation. In the `Menu` you can specify rates that follow a Gamma distribution, with the shape parameter `alpha` of that Gamma distribution, the program then calculates the rates and the rate probabilities (**p**rob-rates).

prob-rates=< n : p1 p2 p3 ... pn>
if you specify your own **rates** you need also to specify the probability of occurrence for each rate. MIGRATE is using, like PHYLIP, Laguerre quadrature points to find the discrete rates with their probability [in contrast to other programs that use discrete values at equal probabilities]

autocorrelation=<Yes:value | No>
if you assume hat the sites are correlated along the sequence, specify the block size, by assuming that only neighboring nucleotides are affected you would give a value=2. [this option may not work in version 4.x]

weights=<Yes | No>
If you specify **Yes** you need a file `weightfile` with weights for each site, the weights can be the following numbers 0-9 and letters A-Z, so you have 35 possible weights available.

```
# Example weightfile for two loci
11111111112222222222
1111112222AAAA445XXXX5
```

inheritance-scalars={value1, value2, ...} The inheritance scalar is relative to the locus that is set to 1.0. If that locus is a nuclear marker and the species is diploid then all Θ are equivalent to $4N_e\mu$, if that locus is a segment of mtDNA then all Θ are equivalent to $N_e\mu$ (maternal inheritance, sex ratio 1:1). If you have 3 loci, for example in this order: a nuclear marker, a mtDNA marker, and an X-linked marker then the input for this option is:
inheritance-scalars={1.0, 0.25, 0.75 }
This expresses all loci as $\Theta = 4N_e\mu$; A second example: if you have two loci, the first is Y-chromosome segment and the second is X-linked and you would want to express all in Θ_Y then
inheritance-scalars={1.0, 3.0 }
or if you want to express in Θ_X then
inheritance-scalars={0.333 1.0}
Use for the reference locus the scalar 1.0 and all other scalars relative to that.

random-subset=<NO | number> MIGRATE can randomly subsample each population. Picking the number specified in the **random-subset**. If the population sample has fewer individuals than the specified number, all samples are taken for that population.

tipdate-file= <NO | YES:datefile >
IF YOU HAVE ONLY CONTEMPORARY DATA DO NOT USE THIS OPTION.
The `datefile` contains sampling-dates for the individuals (the tips of the genealogy). An example is this: `tipdate-file=YES:datefile.bison3`
The datefile format is close to the `infile` format but for obvious content reasons not identical, in generalized form it looks like this:

```
 <Number of populations> <Number of loci>  <Title>
<Number of individuals>  <Population title>
<individual1 1-10>      <Date>
<individual2 1-10>      <Date>
<individual3 1-10>      <Date>
....
<Number of individuals>  <Population title>
<individual4 1-10>      <Dual Date>
```

```
<individual5 1-10>      <Dual Date>
....
```

The individual names MUST match the individual names in the `infile` and all names MUST be unique, this is a stringent requirement that is only needed when you use a datefile to guarantee that the right dates and sequences are matched.

The date must be given as a date measured backwards in time (dual time), so if a bison died 164 BC and you are able to extrac DNA from the bones then you should specify that the bison died 2172 years ago (in 2008), MIGRATE will adjust so that the smallest date will be set to date zero. Here an example using the mentioned syntax:

```
 2 1  Bison priscus dated samples
3  Alaska
a2172     2172
a2526     2526
a4495     4495
2 Siberia
s14605    14605
s23040    23040
```

In the example the dates are the years before present, but in principle they can be any units as lon as the mutation rate per 'year' and the generation-per-year is on the same scale.

mutationrate-per-year= {<mutationrate1>,<mutationrate2>,...}
For example: mutationrate-per-year={0.0000005}
IF YOU HAVE ONLY CONTEMPORARY DATA DO NOT USE THIS OPTION.
If you do not know the mutation rate, guess and try out to estimate the mutation rate in the analysis but depending on your data this may be a taxing analysis. For the moment use the mutation rate per generation and not year, see below.

generation-per-year= <value>
IF YOU HAVE ONLY CONTEMPORARY DATA DO NOT USE THIS OPTION.
The `datefile` needs additional information about the spacing of the samples in time, the number of generations per year helps to get this spacing, but we also need the mutation rate (see above). Example: `generation-per-year=1.000000`. Currently the generation time setting needs further tests, a generation time of 1.0 works, but other settings may fail; for the moment just use 1.0, and translate the results in years if needed.

Microsatellite data

Options that are used when the data are microsatellite repeat markers. MIGRATE uses repeat numbers internally, the infile can specify whether the data is in repeat numbers or in fragmentlength. MIGRATE does not use models that behave differently with very small or very large numbers of repeats, It assumes that the mutation rate for a change from, say,

5 repeats to 6 is the same as from 245 to 246.

Stepwise mutation model: If the **datatype=Microsatellite** is used, the following options have some meaning:

include-unknown=<YES | NO>
The default is bfNO. Alleles that are marked with a "?" are stripped from the analysis with include-unknown=NO. Using YES leaves the "?" in the analysis, under some circumstances this might be the preferred way, but for most situations the unknowns can be safely stripped from the analysis.

micro-threshold=value
specifies the window in which probabilities of change are calculated if we have allele 34 then only probabilities of a change from 34 to 35-44 and 24-34 are considered, the probability distribution is visualized in Figure 3.2 the higher this value is the longer you wait for your result, choosing it too small will produce wrong results. If you get -Infinity during runs of migrate then you need to check that all alleles have at least 1 neighbor fewer than 10 steps apart. If you have say alleles 8,9,11 and 35,36,39 then the default will produce a probability to reach 11 from 35 and as a result the likelihood of a genealogy will be -Infinity because we multiply over all different allele probabilities.
Default is **micro-threshold**=10

usertree=<NO | RANDOM >
The default is **NO** and MIGRATE calculates a starting tree using a UPGMA tree that uses a very simply distance matrix between the samples and then constrains this topology to follow a coalescent.

With the keyword **RANDOM** one can generates a random starting tree with "coalescent time intervals" according to the start parameters. This is generally a bad choice, but in conjunction of many short chains and the **replicate=YES:number** option [number is bigger than 1, see below]. This can help to search the parameter space more efficiently.

For these following options see under *Sequence data* above.

random-subset=<NO | number>

tipdate-file= <NO | YES:datefile >

mutationrate-per-year= {<mutationrate1>,<mutationrate2>,...}

generation-per-year= <value>

Brownian motion approximation: If the **datatype=Brownian** is used, the following options have some meaning:

include-unknown=<YES | NO>
The default is bfNO. Alleles that are marked with a "?" are stripped from the analysis

with include-unknown=NO. Using YES leaves the "?" in the analysis, under some circumstances this might be the preferred way, but for most situations the unknowns can be safely stripped from the analysis.

usertree=<NO | RANDOM >
The default is **NO** and MIGRATE calculates a starting tree using a UPGMA tree that uses a very simply distance matrix between the samples and then constrains this topology to follow a coalescent.

With the keyword **RANDOM** one can generates a random starting tree with "coalescent time intervals" according to the start parameters. This is generally a bad choice, but in conjunction of many short chains and the **replicate=YES:number** option [number is bigger than 1, see below]. This can help to search the parameter space more efficiently.

For these following options see under *Sequence data* above.

random-subset=<NO | number>

tipdate-file= <NO | YES:datefile >

mutationrate-per-year= {<mutationrate1>,<mutationrate2>,...}

generation-per-year= <value>

Allozyme data

include-unknown=<YES | NO>
The default is bfNO. Alleles that are marked with a "?" are stripped from the analysis with include-unknown=NO. Using YES leaves the "?" in the analysis, under some circumstances this might be the preferred way, but for most situations the unknowns can be safely stripped from the analysis.

usertree=<NO | RANDOM >
The default is **NO** and MIGRATE calculates a starting tree using a UPGMA tree that uses a very simply distance matrix between the samples and then constrains this topology to follow a coalescent.

With the keyword **RANDOM** one can generates a random starting tree with "coalescent time intervals" according to the start parameters. This is generally a bad choice, but in conjunction of many short chains and the **replicate=YES:number** option [number is bigger than 1, see below]. This can help to search the parameter space more efficiently.

For these following options see under *Sequence data* above.

random-subset=<NO | number>

tipdate-file= <NO | YES:datefile >

mutationrate-per-year= {<mutationrate1>,<mutationrate2>,...}

generation-per-year= <value>

No special variables, but see **Parmfile specific commands**.

Nucleotide polymorphism

Similar to **sequence data**.

5.2 Input/Output formats

This group of options specifies input file names and various output file options. Currently we only support the Bayesian Approach (BA, Fig. 5.4). The numbering in the menus are not 1,2,3,4,... because I wanted to keep the same numbers for the options that are shared between the two approaches the same.

```
INPUT/OUTPUT FORMATS
--------------------------------------------

INPUT:
 1   Datafile name is                                  twoswisstowns
 2   Use automatic seed for randomisation?                       YES

OUTPUT:
 5   Print indications of progress of run?                       YES
 6   Print the data?                                              NO
 7   Outputfile name is                                      outfile
                                                         outfile.pdf
12   Print genealogies?                                         None
15   Save logging information?                                    NO
19   Show event statistics                 mighistfile (all events)
     Events are recorded every                   every sample step
     Histogram bin width                                  0.001000
20   Record parameter change through time?              skylinefile
     Histogram bin width                                  0.001000


Are the settings correct?
(type Y to go back to the main menu or the letter for the entry to change)
```

Figure 5.4: Input/Output menu of *Migrate*

5.2.1 Input formats

infile=filename
If you insist to have a datafile names other than `infile`, you can change this here, if you do not specify anything here, it will use any file with name `infile` present in the execution directory, if there is no `infile` than the program will ask for the datafile and you can specify the path to it (this may be hard on Macs and Wintel machines). If you use this option, do **N**OT use spaces or "/" or on Macs ":" in your filename. The default is obviously **infile=infile**

random-seed=<Auto | Noauto | Own:seedvalue>
The random number seed guarantees that you can reproduce a run exactly. I you do not specify the random number seed (**seed=Auto**) the program will use the system clock. With **seed=Noauto** the program expects to find a file named `seedfile` with the random number seed. With **random-seed=Own:seedvalue** you can specify the seed value in the parmfile (or in the menu).
Example for own seed:
random-seed=Own:21465 If you want reproducible runs you should replace the **Auto** seed with your own starting number (there are no requirement for the starting number perhaps except 0,] MIGRATE uses the Mersenne-Twister algorithm to generate random numbers). The default is **r**andom-seed=Auto. If you use **random-seed=Own:seedvalue** do nor forget to change the seed for different runs, otherwise the sequence of random numbers is always the same and the result will look the same on the same machine.

Caution: if you run MIGRATE in a simulation study you should set the random number yourself, the AUTO option might produce the same random number seed for runs that are started in the same second: this is quite common under batch-queue systems, when you run the same date from the same seed, you will get always the same result. I tried to improve this by getting a better seed automatically but this is somewhat machine dependent.

title=titletext
if you wish to add an informative title to your analysis, you can do it here or in the infile, the infile will override the title specified here. The length of the title is maximal 80 characters. Example: **title=Migration parameter estimation of populations A and B of species X**.

5.2.2 Output formats

progress=<Yes|No|Verbose> Show intermediate results and other hints that the program is running. Prints time stamps and gives a prognosis when the program eventually will finish, but this is a rather rough guide and sometimes gets fooled. An analogy, the system knows how far to drive and how far we have already driven and

the time, but no clue about how many speed bumps (many migration events) and accidents are ahead of us.

Verbose adds more hints (at least for me) and information. The default is **progress=Yes**

print-data=<Yes|No>
Print the data in the `outfile`. defaults is **print-data=No**. If you run your data for the first time trhough MIGRATE turn this option on, because it helps to find problems with data-reading. Especially with microsatellite data it is possible that the program runs but the loci are incorrectly read.

outfile=filename
All output is directed into this file, the default name is outfile. If you use this option, do **NOT** use spaces or "/" or on Macs ":" in the filename. The default is obviously **outfile=outfile**

print-trees=<All | None | Last | Best>
print genealogies into `treefile`. Remember these trees contain migration events, `treeview` *Page* (1996) and `FigTree` *Rambaut* (2006) can display such trees, although the migration events do not show on these displays, other program might crash. We have a program `eventree - ET for short` that can display all the events on the tree, the program can be downloaded from the Migrate website.

None: `treefile` is not initialized and no trees are printed, this is the fastest and the one I recommend.

All: will print all trees (you want to do that only for ridiculously small datasets with too short chains or you have **many Gigabytes** of free storage).

Last: Only the trees of the last long chain are printed, Still you will need lots of space.

Best: Prints the tree with the highest data-likelihood for each locus. This is slow! And does not give a lot of information, except if you are more interested in the best tree for each locus than in the best parameter estimate.

Default is **print-trees=None**

logfile=<NO | YES:logfilename>
Records the output to the screen into a file when turned on, otherwise the screen output will be lost. On windows systems this may be the only option to see what is going on the because the screen buffer is only 80 lines.

mig-histogram=<NO | <ALL | MIGRATIONEVENTSONLY >:binsize:mighistfilename>
Records the frequencies of migration events (with MIGRATIONEVENTSONLY) or of all migration events and coalescence events (ALL) over time using *binsize*, the binsize is not optimal because you need to fix it before you know the range of times. A value 10 to 20$\times$ smaller than the average population size Θ is a good start. The output is a histogram of frequencies for each parameter, and a summary table of the average

frequencies and a table of the frequency of the location of the root of the genealogy.

skyline=<NO | YES:binsize:skylinefilename>
If you have only contemporary data, do not trust this output. This options depends on the mig-histogram option, it uses the same binsize and needs some of its data structures, therefore do turn on the mig-histogram=ALL.... before attempt to use this option. With this option MIGRATE will present the changes of parameters through time, this method uses a different approach than BEAST and is may be more crude but can represent migration parameters and can summarize over multiple loci.

5.3 Start values for the Parameters

The Parameter menu allows to change the meaning of some of the parameters and allows to set start parameters

```
  PARAMETERS
  --------------------------
  Start parameters:
  1   First parameter values are?
                          combined start-parameter report not finished

 Gene flow parameter and Mutation rate variation among loci:
  2   Use M for the gene flow parameter                  YES [M=m/mu]
  3   Mutation rate is                                       Constant

  Structured coalescent model and combination of localities:
  4   Sampling localities                                    default
  5   Model is set to                      Full migration matrix model
  6   Geographic distance matrix:                                 NO


  Are the settings correct?
  (Type Y to go back to the main menu or the letter for an entry to change)
===>
```

Figure 5.5: 'Start value for the parameter' menu of *Migrate*

Start parameters

theta=<Prior:{percentvalue}| Own:{value1,value2, ...} | Normal:{mean,std}| Uniform:{minimum, maximum} >
The menu option "*Use a simple estimate of theta as start?*" allows to specify a start value for the mutation scaled population size Θ. The 'prior' option is the default, it is

set to 10 (=10% of the prior value), this seems to be OK for many data sets, but this depends strongly on your prior, for example if you set an expoential prior with bounds at 0.0 and 10^10 then start value may be crazy high for your data. Setting prior ranges very wide is usually a bad idea. This option is in principle not important because the MCMC run should be long enough so that the starting values do not matter. In praxis good values of start parameters allow much faster convergence than bad ones. Simulations have shown that starting from too low values typically increases the run-length considerably, whereas to high values seem more to help than hurt, although if the start values are very large and the data is not strong then MA can fail without a clear signal of failure; MIGRATE will return a large parameter estimate that does not reflect the data very well. BA is much less vulnerable to this problem. The start genealogy depends on the start parameters because even with a random topology the times are constrained to come from a coalescence process with parameters set equal to the start parameters defined here.

migration= $<$ **Prior:**$\{$**percentvalue**$\}$|**Own:Migration matrix** | **Normal:**$\{$**mean,std**$\}$| **Uniform:**$\{$**minimum, maximum**$\}$ $>$
The menu option *Use a simple estimate of migration rate as start?* allows to specify a start value for the migration parameter. The 'prior' option is the default, it is set to 10 (=10% of the prior value), this seems to be OK for many data sets, but this depends strongly on your prior, for example if you set an expoential prior with bounds at 0.0 and 10^10 then start value may be crazy high for your data. Setting prior ranges very wide is usually a bad idea. The values for **Own** are given in terms of $4N_em$ which is 4 $\times$ effective population size $\times$ migration rate per generation. The default is **migration=FST**. The **migration matrix** is a n by n table with - on the diagonal and can look like this for four populations `migration=OWN:{ - 1.0 1.1 1.2 0.9 - 0.8 0.7 2.1 2.2 - 2.3 1.4 1.5 1.6 - }`
or like this

```
migration=OWN:{ -   1.0 1.1 1.2
                0.9 -   0.8 0.7
                2.1 2.2 -   2.3
                1.4 1.5 1.6 - }
```

`See note on start values above under` Θ.

Gene flow parameter and mutation rate variation among loci

The gene flow parameter can be presented as xNm or M. M is the mutation scaled immigration rate m/μ that represents the importance of variability brought into the population by immigration compared with the variability created by mutation, m is the fraction of the new immigrants of the population per generation. xNm represents the number of immigrant per generation scaled by x where x depends on the data: $x = 1$ for haploid, uniparental inheritance (mtDNA, Y), $x = 2$ for haploids (bacteria), $x = 3$ for the X

chromosome in the mammal X-Y systems, $x = 4$ for diploid organisms (nuclear DNA), etc.

use-M=<YES | NO>

mutation=<Constant | Estimate| Varying | Relative | Data >
If there are more than one locus the program averages the parameter distributions over all loci (this is different from the average of the most likely parameter values, loci that contribute more peaked parameter distributions are weighted more heavily than parameter distributions that have very flat distributions.]). The mutation rate over all loci can be manipulated in a couple of ways, This options should not be used for first trials with MIGRATE. If you do not have DATED samples, the option 'Estimate' WILL NOT WORK. If you suspect that your data has regions of highy different mutation rates, I suggest to use the 'Relative' or 'Data' (both are the same). The menu presents you with these choices:

```
(C)onstant   All loci have the same mutation rate [default]
(E)stimate  Mutation rate
(V)arying     Mutation rates are different among loci [user input]
(R)elative    Mutation rates estimated from data
```

The **Estimate** flag ("estimate mutation rate") allows for the variation of the mutation rate of each locus proposing new mutation rate values from the prior distribution. For almost all dataset do not use this! Because you will need dated samples to estimate the mutation rate. The options **Varying** allows you to input your own mutation rate modifier. MIGRATE is modifying your values so that the average rate will be 1.0.

The option **Relative** estimates a rough rate modifier for the mutation rate using the data. For sequence data the Watterson estimator is used to get relative rates, this takes into account the different numbers in the sample. For microsatellite and allozyme the allele counts are used to generate a rough value of the rate for each locus. These rates average to 1.0.

With **Constant** no special calculations are done. The summarizing step is simply finding the best parameters by maximizing the sum of the log-likelihoods of each locus. The default is **mutation=Constant**

5.3.1 Migration model

If you do not specify anything the posterior probability densities of all $n \times n$ parameters (n mutation-scaled population sizes, and $n(n-1)$ mutation-scaled immigration rates are found) are found. This options has many complications and is key for comparing population models. One can

- set particular immigration scenarios (for example, symmetric or assymetric migration, groups of migration rates with the same value)
- force the estimation of population splitting times using a model that estimates the

mean and standard deviation of population splits.

- set some connections to zero, this allows to explore stepping stone models or other non-island-model type scenarios.

custom-migration=<NONE | migration-matrix>
The migration matrix contains the migration indicators (or divergence indicators) from population j to i on row i, and the Θ are on the diagonal. The migration matrix can consist of connections that are

- 0: not estimated
- m: mean value of either Θ or $\mathcal{M}$.
- s: symmetric migration [symmetric $\mathcal{M}$ not xNm]
- S: symmetric migration [symmetric ξNm not $\mathcal{M}$]
- c: constant value (together with `migration=OWN...` or `theta=OWN...`)
- * or x: no restriction

(the ξ (xi) means a multiplier: 4 for diploid nuclear markers, 2 for haploid markers, 1 for haploid markers transmitted only through one sex, such as mtDNA and Y chromosomes)

The values can be spaced by blanks, newlines A few examples for 4 populations:

Full model: **custom-migration={****
****
****
****}**
N-island model: **custom-migration={m m m m**
mm mm
m mmm
mmmm}
Stepping Stone model with symmetric migrations, and unrestricted Θ estimates:
custom-migration={*s00 s*s0 0s*s 00s*}
Source-Sink (the first population is the source (Figure 5.6)):

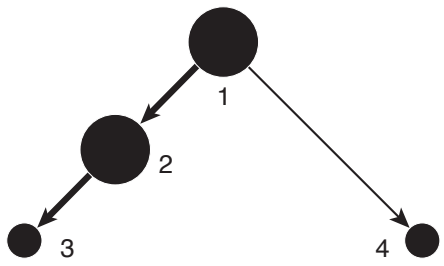


Figure 5.6: Source-sink example

custom-migration={*000000**0*00*}**

5.3.2 Geographic distance between locations

You can specify a distance matrix between your populations. the distance file has the same syntax as a PHYLIP distance file [see example below].

geofile=<NO | YES:filename>
The distance matrix contains the distances between pairs of populations, if you choose for example distance units in kilometers you will get migration rate estimates that are scaled as M = immigration rate / (mutation * kilometer), if you restrict the migration rate to an average value for all connections between population you are calculating a dispersion coefficient based on discrete populations. This coefficient should be in the limit the same as the one calculated from a isolation by distance population model.
There is no requirement that the distances from i to j are the same as from j to i, although interpretation might be difficult with an unequal distance matrix. the default filename for this distance file is *geofile*.

Example *geofile*

```
     3
Ermatingen0.0 10.4 12.4
Schachen  10.4 0.0 1.0
Heiden    12.4 1.0 0.0
```

5.4 Search strategy

This section is the key to good results and you should not just use the defaults, for guidance how I myself would do this check out the section **how long to run**.

5.4.1 Maximum likelihood inference

```
 SEARCH STRATEGY

 0   Strategy:                          Maximum Likelihood
 1   Number of short chains to run?                    10
 2   Short sampling increment?                         20
 3   Number of recorded genealogies in short chain?   500
 4   Number of long chains to run?                      3
 5   Long sampling increment?                          20
 6   Number of recorded genealogies in long chain?   5000
 7   Number of genealogies to discard at
     the beginning of each chain? [Burn-in]         10000
 8   Combine chains or runs for estimates?             NO
 9   Heating:      YES (  4 chains, swap interval is   1)

-------------------------------------------------------------
Obscure options (consult the documentation on these)

10   Sample at least a fraction of new genealogies?    NO
11   Epsilon of parameter likelihood             infinity
12   Use Gelman's convergence criterium?      YES:Summary


 Are the settings correct?
 (Type Y to go back to the main menu or the number for a menu to change)
===>
```

Figure 5.7: 'Search strategy' menu with the Maxim likelihood approach

The terminology of **short** or **long** chains is arbitrary, actually you could choose values so that short chains are longer than the "long" chains. Anyway, Markov chain Monte Carlo (MCMC) approaches tend to give better results when the start parameters are close to the maximum likelihood values. One way to achieve this is running several short chains and use the result of the last chain as starting value for the new chain. This should produce better and better starting values, if the short chains are not too short.

Strategy
With version 2.0 you have a choice of either using a maximum likelihood procedure or a Bayesian approach, on nice data both method will work about the same, for some example runs it seems that the profile likelihoods and the Bayesian posterior

distribution agree quite fine on the distribution of the parameter value. The options specific to the Bayesian approach are explained in the next section.

Number of short chains to run? (short-chains=value
we run most of the time about 10 short chains, which is enough if the starting parameters are not too bad. Default is **short-chains=10**.

Short sampling increment? (short-inc=value)
The sampled genealogies are correlated to reduce the correlation between genealogies and to allow for a wider search of the genealogy space (better mixing), we sample not every genealogy, the default is **short-inc=20** means that we sample a genealogy and step through the next 19 and sample then again.

Number of steps along short chains? (short-steps=value)
The default number of genealogies to sample for short chains is 500. But this may be to few genealogies for your problem. If you big data sets it needs normally bigger samples or higher increments to move around in the genealogy space.

Number of long chains to run? (long-chains=value)
I run most of the time 3 long chains. The first equlibibrates and the last is the one we use to estimate the parameters. Default is **long-chains=3**.

Long sampling increment? (long-inc=value)
The default is the same as for short chains.

Number of steps along long chains? (long-steps=value)
The default number of genealogies to sample for long chains is 5000. I often choose the "long" chains about 10 times longer than the "short" chains.

Number of genealogies to discard at the beginning of each chain? (burn-in=value)
Each chain inherits the last genealogy of the last run, which was created with the old parameter set. Therefore the first few genealogies are biased towards the old parameter set. When **burn-in** is bigger than 0, the first few genealogies in each chain are discarded. The default is **burn-in=10000**.

Combine chains for estimates The use of this option is recommended for difficult data sets. It allows to combine multiple chains for the parameter estimates when you use **replicate=YES:LongChains**. With **replicate=YES:number** where number is, well, a number bigger than 1. (e.g. replicate=Yes:5), you run the program "number" times and the results of their last chains are combined, The method of combination of chains is the same as in *Kuhner* et al. (1995b) and is based on the work by *Geyer* (1991). The LongChain option does not need much more time than the single chain option, but the full replication needs exactly "number" times a normal run. But is sampling the search space much better than any other option, I use this often in conjunction with random starting trees (randomtree=YES).

Heating (heating=<NO | YES | ADAPTIVE <:waitnumber:{cold,warm,hot,boil,....}>

This allows for running multiple chains and swap between them, when these chains are run at different temperatures, the "hotter" chains explore more genealogy space than the "cold" chains. An acceptance-rejection step swaps between chains so that the the "cold" chains will sample from peaks on the genealogy surface proportional to their probability. This scheme is known as MCMCMC (Markov coupled Markov chain Monte Carlo), it is based on the work of *Geyer* and *Thompson* (1995) and uses for four or more chains at different temperatures, the hotter chains move more freely and so can explore other genealogies, this allows for an efficient exploration of data that could fit different genealogies, and should help to set the confidence intervals more correctly than a single chain path could do. You need to set the temperatures yourself because there is no default. The ADAPTIVE heating scheme manipulates these temperatures according to their swapping success. If a neighboring temperature pair is not swapping after 1000 trials the temperature difference between them is lowered by 10%, if a pair is swapping more than 10 times in a 1000 the gap is increase by 10% (these values are arbitrary, but cannot be changed in the menu, yet). Adaptive heating is definitely no cure-it-all, I typically prefer the static heating, but it helps find good values to try for the static heating scheme, I have seen pathological behavior of long adaptive runs where all chains essentially converged to values very very close to 1.0 (the cold chain) and stopped swapping.

If you use a STATIC heating scheme then you need to experiment a little because you want that the different chains swap once in a while, but not too often and certainly more often than never. The swapping seems to depend on how good the data describes a given genealogy. I would start with 4 chains and temperatures that are **{1. 1.5 3.0 10000.0}**. The temperatures are ordered from cold to boiling, the coldest temperature **MUST** be 1 (one). The default for the heating option is **heating=NO**. If you use this option sampling will be at least 4 times slower, except if you have a multiprocessor machine and a POSIX compliant thread-library (often called with slight variations but containing word parts such as pthread, thread, linuxthread), then you can compile the program using "make thread", this will improve speed somewhat, but lately I do not gain more than 170% CPU usage out of this. It is probably easier and faster to use all cores on new computers using the parallel version of MIGRATE.

The **waitnumber** is the number of trees to wait before the differently heated chains are check whether to swap or not. I normally use 1. I have little experience whether, say, using 10 improves mixing over using 1.

Obscure options

If you are not experienced with MCMC or run *Migrate* for the first, second, ... time, do not bother about the options here.

Sample at least a fraction of new genealogies? (moving-steps=<Yes:ratio | No >)
With some data the acceptance ratio is very low, for example with sequence data with more than 5000 bp the accpetance ratio drops below 10% and one should increase the length of the chains. One can do this either by increasing the **long-inc**, or **long-steps** or by using **moving-steps**. The ratio means that at least that ratio of genealogies specified in **long-steps** have to be new genealogies and if that fraction is not yet reached the sampler keeps on sampling trees. In unfortunate situation this can go on for a rather long period of time. You should always try first with the default **moving-steps=No**. An example:
You specified **long-steps=2000**,and **long-inc=20** and the acceptance-ratio was only 0.02, you have visited 40,000 genealogies of which only 800 are new genealogies so that you have maximally sampled 800 different genealogies for the paramter estimation. In a new run you can try **moving-steps=Yes:0.1**, the sampler is now extending the sampling beyond the 40000 genealogies and finally stopping when 4000 new genealogies were visited.

Epsilon of parameter likelihood (long-chain-epsilon=value)
The likelihood values are ratios

$$\frac{L(\mathcal{P})}{L(\mathcal{P}_0)} = \frac{1}{n}\sum_i \frac{\mathrm{p}(G_i|\mathcal{P})}{\mathrm{p}(G_i|\mathcal{P}_0)} \qquad \text{(Beerli and Felsenstein, 1999)}$$

When the Likelihood values are very similar then the ratio will be close to 1, or 0 when we use logarithms. This means that the sampler is not improving drastically between chains: (a) it found the maximum likelihood estimate or (b) it is so far from the maximum likelihood estimate that the surface is so flat that all likelihood values are equally bad. using a smaller value than the default **long-chain-epsilon=100.00** for example a value of 1.0 would guarantee that the sampler keeps on sampling new long chains as long as that log-likelihood-difference drops below 1.0. In some cases this will never happen and the program will not stop.

Gelman's convergence criterium If you specify "Yes" then the number of last chains get extended until the convergence criterium of Gelman is satisfied (the ratio should be smaller than 1.2 for **all** parameters. This can take a very long time. [In the parallel version this fails, turn it off there [this is a bug, but I had not time to find and fix it]).

5.4.2 Bayesian method

File for recording parameters? (bayesfile=<NO | YES:bayesfile>) this file contains the raw histogram for all parameters and all loci and their combination, figure 5.9 shows the first few lines of an example, see under section **Bayesian posterior explained** further uses of this file.

```
 SEARCH STRATEGY

   0   Strategy:                                        Bayesian Inference
   1   File for recording posterior distribution?                       NO
   2   File for recording all parameter values?                         NO
   3   Number of bins of posterior [Theta,M]?                  200, 200
   4   Plotting type of posterior distribution? up to ~100% percentile
   5   Frequency of tree updates vs. parameter updates?              0.50
   6   Proposal distribution?          Theta:Slice Mig:Slice Rate:Slice
   7   Prior distribution?             Theta:Unif. Mig:Unif. Rate:Unif.
   8   Number of long chains to run?                                    1
   9   Sampling increment?                                             20
  10   Number of recorded steps in chain                             5000
  11   Number of steps to discard at
       the beginning of chain? [Burn-in]                            10000
  12   Running multiple replicates:                                    NO
  13   Heating:                              STATIC (  4 parallel chains)
  14   Sampling at least fraction of new genealogies:           0.000000
  15   Convergence diagnostic for replicates:                YES:Summary


  Are the settings correct?
  (Type Y to go back to the main menu or the number for a menu to change)
===>
```

Figure 5.8: 'Search strategy' menu with the Bayesian approach

File for recording all parameter values? (bayes-allfile=<NO | YES:number:bayesallfile>) this file contains the raw histogram for all parameters and all loci and their combination, figure 5.10 shows the first few lines of an example, see under section **Bayesian posterior explained** for further uses of this file. This file can be very large depending on your options, it is still hard to work with files larger than 10 GB, so choose you settings carefully, there will be samples×loci×replicates sets of n^2 parameters and some additional values. If you need more samples to get good results and your data is highly autocorrelated increase the long-inc options (see there). If you specify this option (recommended) the memory foot print of the program is smaller than when this option is set to NO. This is important particularly for the parallel MIGRATE runs.

Number of bins of posterior (bayes-posteriorbins=<thetabins Mbins <ratebins>>) The number of bins for the posterior needs to be pre-specified (to save memory). The default for Θ, M is 200 bins. This number is probably to small if the range of the prior distribution is very large. If the PDF histograms look course rerun after increasing the binsizes. The ratebins are used when the mutation rate modifier with many loci is estimated in the Bayesian analysis, this may sometimes fail, because there is little information about rate differences among loci in some datasets.

Plotting bins of posterior (bayes-posteriormaxtype=<TOTAL | P100| P99| MAXP99 >)

```
# Raw data for the histogram of the posterior probabilities for all parameters
# and loci  produced by the program migrate-n 2.0.3
# (http://evolution.gs.washington.edu/lamarc/migrate.hml)
# written by Peter Beerli 2004, Tallahassee,
# if you have problems email to beerli@csit.fsu.edu
#
# The HPC values are indicators whether the parametervalue is in the
# highest-posterior credibility set, a 0 means it is outside and a 1 means
# the value is inside the credibility set.
#
# Delta for Theta and M 0.001000 0.001000 9.995000 9.995000
# ------------------------------------------------------------
# Locus Parameter 50%HPC  95%HPC (parameter-value count) frequency
# ------------------------------------------------------------
1 1 0 0 0.002499 327 0.001635
1 1 0 0 0.003498 1634 0.008169
1 1 0 1 0.004498 4612 0.023058
1 1 0 1 0.005497 8970 0.044846
1 1 1 1 0.006497 13576 0.067874
1 1 1 1 0.007496 17320 0.086592
1 1 1 1 0.008496 19492 0.097451
1 1 1 1 0.009495 20537 0.102676
1 1 1 1 0.010495 19504 0.097511
```

Figure 5.9: First few lines of a bayesfile: the header explains the columns

The posterior distribution often covers only a short range of the prior distribution, therefore displaying the **TOTAL** range of the prior distribution is often not advised, the P99 presents 99% of the posterior distribution, cutting of 1% of the posterior, this is a good way to visualize posterior distributions with very long (thin) right tails. P100 takes 99.99% of the values and excludes strange outliers. MAXP99 is cutting of at 99% credibility, but using the parameter with the highest value for Θ, and M, in principle this forces the same scale in the output for the parameters (this needs more testing because I most often use P100).

Frequency of tree updates versus parameter updates (bayes-updatefreq=< value >)
The *value* specifies the ratio of genealogy updates and parameter updates, 0.5 means that the genealogy is updated roughly every second time, and one of the parameters is updated every second time. A value of 1.0 means that the parameters are never updated, A value of 0.0 is not advised because the genealogy does not adjust the migration events and so does not really test the parameter distribution for a specific tree.

Proposal distribution
bayes-posterior=< < THETA | MIG | RATE > < SLICE | METROPOLIS > >)

There are two ways the generate posterior distributions: SLICE and METROPOLIS. METROPOLIS is using the standard Metropolis-Hastings algorithm that proposes a new state not taking into account the data and then accepting or rejecting using the fit if the data to the old and new state. For some data the rejection rate is very high and many computer cycles are wasted because the MCMC chain does not move. SLICE sampling uses the current posterior distribution (taking into account the data) to generate a new state, every new state is compatible with the data, therefore the acceptance ratio is always 1.0. This comes at a price because the calculations are more demanding than the MH algorithm, and therefore may be slower. On data with lots of information SLICE sampling is great, but fails with poor data. SLICE is the default in MIGRATE.

Examples:

```
bayes-proposals= THETA SLICE Sampler
bayes-proposals= MIG SLICE Sampler
bayes-proposals= RATE SLICE Sampler
```

Prior distribution
bayes-priors=< < THETA | MIG | RATE > < PRIORSPECIFICATION > >)
There are several prior distribution available, but the list is still short. For each prior distribution you need to specify additional parameters:

Distribution	parameter 1	param
Uniform	Minimum	Maxi
Exponential	Minimum	Me
Windowed Exponential	Minimum	Me

Examples:

```
bayes-priors= THETA EXPPRIOR: 0.000000 0.250000 0.500000
bayes-priors= MIG WEXPPRIOR: 0.000000 500.000000 1000.000000 100.000000
bayes-priors= RATE UNIFORMPRIOR: 0.010000 100.000000 5.000000
```

Number of long chains to run? (long-chains=<value>)
Use 1 long chain because multiple long chains will do little to help the analysis, if you want to combine over replicated runs use the replicate option.

Sampling increment? (long-inc=<value>)
Samples are taken every *value* cycle, the default is 20.

Number of recorded genealogies in chains? (long-steps=value)
The default number of genealogies to sample for long chains is 50000. With the default increment this means 1,000,000 genealogies will be visited This is short for many datasets.

Number of genealogies to discard at the beginning of each chain?

(burn-in=value)
The chain is not equilibrated at the beginning of the run, and we discard those aberrant values and trees. The default is **burn-in=10000**.

Combine chains for estimates The use of this option is recommended for difficult data sets. It allows to combine multiple chains for the parameter estimates when you use **replicate=YES:number**; where the number is bigger than 1. (e.g. replicate=Yes:5). You run the program "number" times and the results are combined (similar to MrBayes). The option **replicate=YES:number** works for both Bayesian inference and Maximum Likelihood. The work is number times more than without replication. For ML there is another option: **replicate=YES:LongChains** that allows the combination of long chains only, this is the same method as used by *Kuhner* et al. (1995b) and is based on the work by *Geyer* (1991). The LongChain option does not need much more time than the single chain option.

Replication allows a better sampling of the search space than the single chain. in particular when used on parallel cluster computers. I use this often in conjunction with random starting trees (randomtree=YES).

Heating (heating=<NO | YES | ADAPTIVE <:waitnumber:{cold,warm,hot,boil,....}>
This allows for running multiple chains and swap between them, when these chains are run at different temperatures, the "hotter" chains explore more genealogy space than the "cold" chains. An acceptance-rejection step swaps between chains so that the the "cold" chains will sample from peaks on the genealogy surface proportional to their probability. This scheme is known as MCMCMC (Markov coupled Markov chain Monte Carlo), it is based on the work of *Geyer* and *Thompson* (1995) and uses for four or more chains at different temperatures, the hotter chains move more freely and so can explore other genealogies, this allows for an efficient exploration of data that could fit different genealogies, and should help to set the confidence intervals more correctly than a single chain path could do. You need to set the temperatures yourself because there is no default. The ADAPTIVE heating scheme manipulates these temperatures according to their swapping success. If a neighboring temperature pair is not swapping after 1000 trials the temperature difference between them is lowered by 10%, if a pair is swapping more than 10 times in a 1000 the gap is increase by 10% (these values are arbitrary, but cannot be changed in the menu, yet). Adaptive heating is definitely no cure-it-all, I typically prefer the static heating, but it helps find good values to try for the static heating scheme, I have seen pathological behavior of long adaptive runs where all chains essentially converged to values very very close to 1.0 (the cold chain) and stopped swapping.

If you use a STATIC heating scheme then you need to experiment a little because you want that the different chains swap once in a while, but not too often and certainly more often than never. The swapping seems to depend on how good the data describes a given genealogy. I would start with 4 chains and temperatures that are **{1. 1.5 3.0 10000.0}**. The temperatures are ordered from cold to boiling,

the coldest temperature **MUST** be 1 (one). The default for the heating option is **heating=NO**. If you use this option sampling will be at least 4 times slower, except if you have a multiprocessor machine and a POSIX compliant thread-library (often called with slight variations but containing word parts such as pthread, thread, linuxthread), then you can compile the program using "make thread", this will improve speed somewhat, but lately I do not gain more than 170% CPU usage out of this. It is probably easier and faster to use all cores on new computers using the parallel version of MIGRATE.

The **waitnumber** is the number of trees to wait before the differently heated chains are check whether to swap or not. I normally use 1. I have little experience whether, say, using 10 improves mixing over using 1.

Sampling at least fraction of new genealogies (moving-steps=<NO | YES:value >) This allows to specify that a minimum number of different genealogies need to be sampled, it is expressed as the ratio of sampled genealogies. If the frequency is not reached at the end of the specified number of samples, MIGRATE will continue until the ratio is satisfied, with high numbers the program may run forever. I rarely use this option.

Convergence diagnostic for replicates (gelman-convergence=< NO | YES:<Sum | Pairs > > This collects information about the convergence rate of two replicated chains (use two or more replicates). *Sum* reports the an average value over all whereas *Pairs* using the pairs of replicates to. Version 3.0 has some difficulties with this option and I hope to fix this in the next minor release, but on some machine and under some conditions the diagnostic fails.

5.4.3 Parmfile specific commands

Important parmfile options

menu=<Yes|No>
defines if the program should show up the menu or not. The default is **menu=Yes**.

end
Tells the parmfile reader that it is at the end of the parmfile.

Options to change the lengths of words and texts

If you change these, you should understand why you want to do this.

nmlength=number
defines the maximal length of the name of an individuum, if for a strange reason you need longer names than 10 characters (e.g. you need more than 10 chars to

characterize an individual) and you do not need this very often then set it to a higher value, if you have no individual names you can set this to zero (0) and no Individual names are read. the default is **nmlength=10**, this is the same as in PHYLIP.

popnmlength=number
Is the length of the name for the population. The default is **popnmlength=100**

allelenmlength=number
This is only used in the infinite allele case. Length of an allele name, the default should cover even strange lab-jargons like `Rvf` or `sahss` (*Rana ridibunda* very fast, *Rana saharica* super slow) The default is **allelenmlength=6**

```
# Migrate debug 3.0 (Peter Beerli, (c) 2008)
# Raw results from Bayesian inference: these values can be used to generate
# joint posterior distribution of any parameter combination
# Writing information on parameters (Thetas, M or xNm)
# every 2 parameter-steps
#
# --  Steps
# --  Locus
# --  Replicates
# --  log(Posterior)
# --  log(prob(D|G))
# --  log(prob(G|Model))
# --  log(prob(Model))
# --  Sum of time intervals on G
# --  Total tree length of G
# Order of the parameters:
# Parameter-number Parameter
#@      1    Theta_1
#@      2    Theta_2
#@      3    M_(2,1)
#@      4    M_(1,2)
#
# --  Thermodynamic temperature = 1.000000
# --  Thermodynamic temperature = 1.500000
# --  Thermodynamic temperature = 3.000000
# --  Thermodynamic temperature = 1000000.000000
# --  Marginal log(likelihood) [Thermodynamic integration]
# --  Marginal log(likelihood) [Harmonic mean]
#
#$ ------------------------------------------------------------------
#$ begin  [do not change this content]
#$ Model=****
#$ Mode2=****
#$ 1 2 4 0 1 1
#$ pop00
#$ pop01
#$ end
#$ ------------------------------------------------------------------
#
# remove the lines above and including @@@@@, this allows to use
# Tracer (http://tree.bio.ed.ac.uk/software/tracer/) to inspect
# this file. But be aware that the current Tracer program (October 2006)
# only works with single-locus, single-replicate files
# The migrate contribution folder contains a command line utility written
# in PERL to split the file for Tracer, it's name is mba
# @@@@@@@@
#Steps  Locus   Replicate       lnPost  lnDataL lnPrbGParam     lnPrior treeintervals   tr
100     1       1       -22365.119577   -22620.671234   255.551656      -17.034386      95
200     1       1       -22367.961876   -22622.328216   254.366340      -17.034386      95
300     1       1       -22368.867271   -22618.681322   249.814051      -17.034386      95
```

Figure 5.10: First few lines of a bayesallfile: the header explains the columns, the data section is truncated at the right and bottom

6 How to run migrate

If you have compiled and installed the program successfully (see Installation) and your data is in a good format (section data format) and perhaps has the name infile, just execute

Command	Parameters	Comments
`migrate-n`		No option will take the default `parmfile` if present
`migrate-n`	parmfile.test	opens the file `parmfile.test` if present otherwise creates a new file that can be save through the menu
`migrate-n`	parmfile.test -menu	forces the program to show the menu
`migrate-n`	parmfile.test -nomenu	forces the program to NOT show the menu and start running immediately (use the -nomenu option for batch scripts and batch queue system.

On some systems you need to call MIGRATE using `./migrate-n`.

On most graphical systems you can start MIGRATE by double-clicking its icon, but the results are different among the different computer systems (Linux, MacOS 10, Windows). On Macs home directory and that is most likely not the location where your files sit. It is actually easier to open the Terminal.app (in /Applications/Utilities) and learn a couple of shell commands (a minimal set of cd, mv, cp will probably do for a start) (see for example this online tutorial http:). Within the terminal window you change to the directory with the data and then execute the program that either is in the same folder using the commands above. For windows double-clicking opens also a terminal window that is located at the same directory location as the icon, if your data is also in that same location your are set, but you can use the "Run..." command from the Startup menu to open a terminal window and then use chdir, copy, rename to operate the windows shell similarly to the UNIC shell.

Without any **parmfile**, *Migrate* will display a menu, in which you can change all the sensible options. For hints how to use the parmfile, look into section **Menu and Options** or the `parmfile`. Once you know how to customize the options with the **parmfile** you will probably more often edit the parmfile than making the changes in the menu. Be careful, some complex options are most easily set through the menu.

7 Bayesian inference

From a practical viewpoint we can think of Bayesian inference of a combination of knowledge: the prior knowledge and the knowledge gained through the data and model. The prior knowledge is used to to treat the parameters of interests as random variables with a distribution that is typically independent of our investigation, the prior distribution. The posterior distribution is the product of the prior distribution and the probability of the data given the parameters (the likelihood). The prior distribution needs to cover the interesting part of the range of the parameters, essentially the posterior distribution should fit within the range of the prior distribution. It is important to inspect the posterior for probably truncation by the prior, if that occurs ou should rerun the analysis. If the prior is much larger than the parameter region of interest, in many circumstances the analysis will take a long time because most values proposed from the prior do not fit well with the data and are rejected.

7.1 Prior distribution

Currently there are three distributions available: uniform, exponential with boundaries, exponential with boundaries and windowing:

Uniform prior You need to specify a lower and an upper bound, this prior distribution is similar to other programs, such as those by Hey and Nielsen (2004). Uniform assume that all parameter values are equally likely, an assumptions that often is not justified.

Exponential prior with boundaries This proper prior distribution (it integrates to 1) needs a lower and and an upper bound and a mean, if one specifies 0.0 and ∞ as boundaries, this distribution is the same as a simple exponential distribution. Preliminary runs show that this distribution is superior (aka converges faster) than the uniform distribution prior. Typically the boundaries are chosen so that there is a large set of possible values in between, the method is picking randomly in this range and so from one step to the next large differences in parameter values can occur, this large differences might lead to a larger rejection rate.

Exponential prior with boundaries and window Same as exponential prior with boundaries except that you need to specify an additional parameter that specifies the window size in from which changes in parameters are drawn. The chain will less often reject parameter values because they will be closer to the last value. This prior distribution seems to produce the best results so far, but it needs some fidgeting with the window. If the window is too small very long chains need to be run to explore the whole distribution, if the window is

too large than the method reduces to the exponential prior with boundaries.

7.2 Proposal distribution: Slice sampling versus Metropolis-Hastings sampling

MIGRATE allows to use two different proposal functions for the evaluation of the parameters, but only one for the evaulation of genealogies: Metropolis-Hastings sampling *Metropolis* et al. (1953) and Slice sampling (*Neal*, 2003). Metropolis-Hastings (MH) sampling is standard in most applications in population genetics, but Slice-sampling is not. I know that Paul Lewis (University of Connecticut) is working on a phylogeny program that uses slice-sampling but I have not see the program in the wild. Paul helped me to understand the slice-sampling method in 2006 at the molecular evolution workshop in Woods Hole. Slice sampling uses the data and the prior distribution to choose a new prior value, because the data already is comaptibel with this new value a MH-rejection step is not necessary and the new value is always accepted. In contrast to that, MH-sampling picks a value from the prior and then uses the data later in the rejection-acceptance step to accept or reject the new value. Experiments have shown that slice-sampling converges typically faster and produces smoother posterior distributions with less steps on the MCMC chain.

7.3 Posterior distribution

MIGRATE prints a file *bayesfile* that contains the raw histogram values for all parameters, the columns in that file allow to use graphing program such as GNUPlot (http://www.gnuplot.info/) to plot the distribution. My favored program to plot such graphs is GMT (General mapping tool, http://gmt.soest.hawaii.edu/) that produces postscript output, in the contribution directory I added a shell script (sorry, no MS Windows utility) that uses GMT and that produces posterior distributions that display the 95% credibility set.

MIGRATE also allows to save the raw parameter values that are used for the posterior distribution (*bayesallfile*). This file contains all information necessary to recreate the posterior histograms from scratch, This file also is compatible with the program TRACER (*Rambaut*, 2007) when you analyze a single locus without replication. There is a command line utility in the contribution directory. This utility *mba* allows to separate the large bayesallfile into files per locus and replicates. It also allows the assembly of different files, that can then be feed back to MIGRATE to recreate the posterior histograms. If you run MIGRATE on a cluster in parallel use turn on this option because the memory footprint of MIGRATE is much smaller than when this option is turned off.

7.4 Prior distributions: choice and problems

to come

8 Model selection

[This section is not finished] MIGRATE allows to calculate the probability of the model using three approaches:

1. Akaike's information criterion (AIC) for maximum likelihood inference An option the parmfile allows to turn on a search for all migration model that are subsets of the model that was used to sample genealogies [this may break on several models with low number of estimated parameters]. This option may use a very long time (longer than you want to wait) when there are more than 4 (!) populations. The number of migration models increases hyperexponentially with number of populations, AIC tests with more than 6 populations will take forever. These tests are only approximate because only the full model was evaluated through the MCMC run.
2. Bayes factors Bayes factors evaluate the merit of hypotheses and models in a Bayesian context. BF do not need to compare nested hypotheses (necessary for likelihood ratio tests). Evaluating Bayes factors is problematic because the marginal likelihoods needed to calculate the BF are difficult to evaluate. In a Bayesian inference program we normal only need to record the parameter values to construct the posterior distribution (histogram). For the marginal likelihood we need to estimate the denominator of the Bayes formula, we can integrate by recording all priors and likelihoods. Two methods are implemented in MIGRATE:
   a) Harmonic mean estimator: described by Kass and Raftery (1996) . This method is know to be fast but inaccurate. It is implemented in many other programs (BEAST/Tracer, MrBayes)
   b) Thermodynamic integration: described by Gelman and Meng (2003). This method needs multiple chains that run at different temperatures (use static heating because the other methods are not well explored yet). This methods can be very accurte but time consuming.

9 Performance of migrate

Markov chain Monte Carlo programs are difficult to use and despite what people tell you very error-prone. This chapter tries to convince you that MIGRATE *often is doing the correct thing, and when something goes wrong that you perhaps can find out why and how it went wrong.*

Markov chain Monte Carlo samplers have the proven property that when they are run infinitely long they converge to the correct value, but since we cannot run the program infinitely long, we are interested how many samples we need to get before we start to get "acurate" result. This is true for maximum likelihood and Bayesian inference modes of the program. Despite the huge literature about measures when to stop sampling, there is still no good universal criteria available. MIGRATE reports some measures, such as the effective sample size of an MCMC run, or the Gelman-Rubin statistic. The problem of difficulty to converge can be divided into three simple categories:

1. Programming errors, typically programs of this complexity will always contain some errors, programmers certainly try to make every effort to make sure that there are no errors in the main calculations, but testing is typically very difficult especially when interactions among multiple options, different hardware need to be tested.
2. The sampler was not run long enough, this is data dependent and some general guidelines could be given, but NSF panels do not seem too keen to fund projects that would do that. To my knowledge, no study has explored effects of sample size, sequence lengths/variability of sequence for more than a single population (*Pluzhnikov* and *Donnelly*, 1996; *Felsenstein*, 2005; *Carling* and *Brumfield*, 2007). You have to explore this with your own data.
3. The assumptions of the model are not met, all data will violate some of the assumptions but typically the method is quite tolerant.

I will discuss some ways to investigate these three sources of problems in the following paragraph, highlighting the potential source of error.

The program is sampling form the right distribution: running the sampler with no data (e.g. sequence data with all "?" data) should result in the distribution $\mathrm{p}(G|\mathcal{P}_0)\mathrm{p}(D|G)$, the one we sample from [checks **(1)**]. With Bayesian inference the uninformative data runs will return the prior distribution [checks **(1)**].

Large simulation studies show that we can recover parameters and population structure that was used to create the data [checks **(1,2)**]. Such simulations need to be planned very carefully because silly parameter combination may suggest that the method does not work, but we perhaps would hope that under biological useful parameter ranges the program should deliver good results, an example of a study where the parameter range was not optimal is a paper by *Abdo* et al. (2004). Real data may have difficulties to deliver consistent results, the most common source of this problem seems that either the model is heavily violated (non-neutral loci, non-random mating, very high rate of recombination). For many data sets this seems not to be a problem, so.

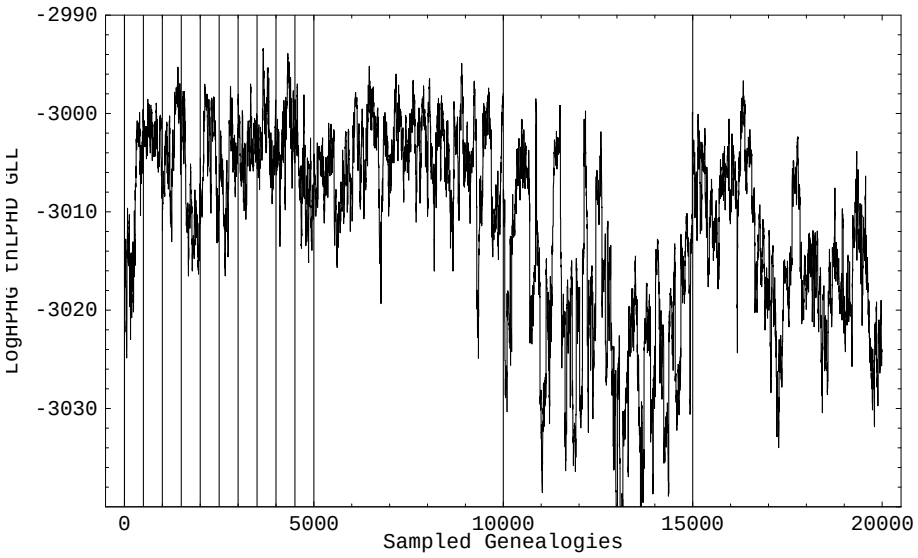


Figure 9.1: Data likelihood $\mathrm{p}(D|G)$ for all sampled genealogies: A sample run of migration estimation using 2 populations, the very long vertical lines mark chain boundaries (10 short and 3 long chains). Totally, 10 short chains $\times 500$ sampled genealogies $+3$ long chains $\times$ sampled 5000 genealogies were sampled out of total 400,000. The values for not recorded trees are not shown.

The program is sampling many different genealogies; one can show this by plotting a curve showing on the x-axes all sampled trees and on the y-axis the likelihood of the genealogy (in our case this is $\mathrm{p}(D|G)$, Figure 9.1). A plot of a sequence of $\mathrm{p}(\mathcal{P}|G_i)\mathrm{p}(D|G_i)$ is not useful because the genealogies contain different number of time intervals, and they are **not** comparable.

One can show that starting from random start parameters, the estimates converge rather quickly after a few short chains (Figure 9.2), the updating of the start parameters over several short chains moves the estimates to the proper region and the remaining uncertainty is only driven by the often huge uncertainty about the parameter estimates in the data, the likelihood surface is flat for many parameter combinations and the data. [checks **(2)**]

Comparison with other programs produce similar results. I compared MIGRATE with GENETREE (*Bahlo* and *Griffiths*, 2000) and with `fluctuate` (*Kuhner* et al., 1998). The comparison with

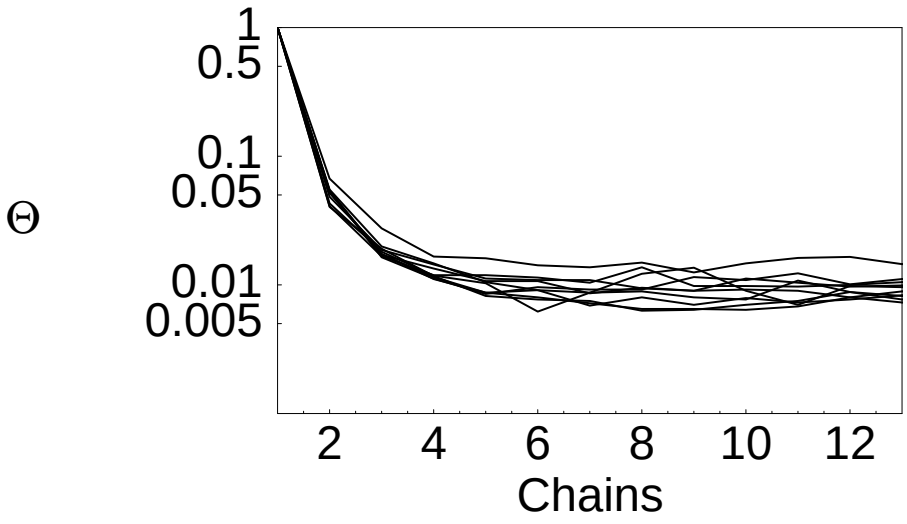


Figure 9.2: Convergence to the true parameter region. Ten runs were started from a $\Theta = 1.0$. The data was generated using a $\Theta = 0.01$. Totally, 10 short chains $\times 500$ sampled genealogies $+3$ long chains $\times$ sampled 5000 genealogies were sampled out of total 400,000.

GENETREE used two populations (England and Ghana: 2.5 kb sequence data for the beta-globin locus (*Harding* et al., 1997)) and the results were very similar. For my paper on n-population I have worked out a 100-locus data set simulation that shows that GENETREE and MIGRATE deliver the same estimates, and approximative confidence intervals, although GENETREE is very slow compared to MIGRATE for that specific data set (*Beerli* and *Felsenstein*, 2001).

The comparison with `fluctuate` was for one population, yes you can run MIGRATE with only one population, and for a data set created using a $\Theta = 0.01$ MIGRATE delivered $\Theta = 0.0123$ with a 50% confidence interval of 0.08 to 0.017, while `fluctuate` delivered a point estimate of $\Theta = 0.0119$.

In 2007, *RoyChoudhury* and *Stephens* published a new method to infer population-scaled mutation rates, their findings helped me to find a problem with my microsatellite estimator and now accuracy and speed are very similar to their estimator (Figure 9.3 *Beerli*, 2007). [checks **(1,2)**]

MIGRATE Version 2.2 and newer print out statistics that help to assess whether the program was run long enough: (1) effective sample size [ESS] and (2) Rubin-Gelman statistic to assess convergence. I am not a strong believer of such measures because they only show the worst problems. For example effective sample sizes of 1000 may seem a lot but it certainly depends on the number of other parameters and the correlation among parameters. The program TRACER (Rambaut et al. 2005) flags effective sample sizes below 100; this is very low for population genetic

purposes, I suggest that you strive to get at least 1000 or more for all parameters including the likelihood of the genealogies.

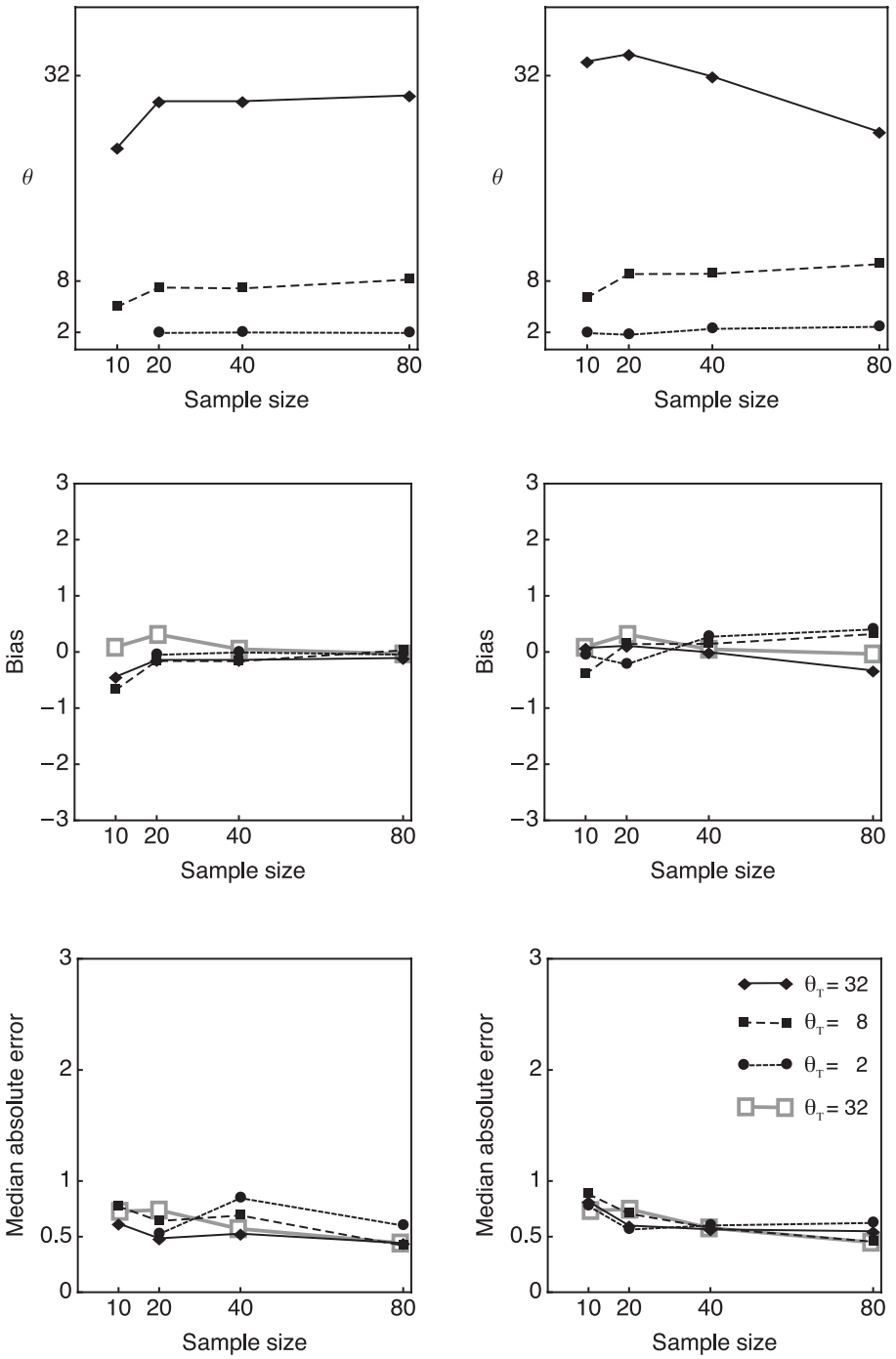


Figure 9.3: Mutation-scaled population size estimated from microsatellite data. Bias and absolute error for MIGRATE version 2.3. Left column: using the stepwise mutation model. Right column: using the Brownian motion approximation, Scale and calculations of bias and absolute error are the same as in Figure 1 in *RoyChoudhury* and *Stephens* (2007). The open squares are the values for θ=32 from their paper.

10 Quick guide for achieving "good" results with migrate

10.1 Monitoring progress

The output for Bayesian inference is terse, but the same rules apply, except that you should run only one long run because the prior distribution will deliver many different "driving" values, convergence issues are still present but less severe as in the ML approach. Under Bayesian inference the effective sample size and autocorrelation are printed out and give a good idea about how well the run succeeded. The example below is a long parallel run of multiple microsatellite loci and replication for a total of 180,000,000 updates.

MCMC-Autocorrelation and Effective MCMC Sample Size

Parameter	Autocorrelation	Effective Sampe Size
Θ_1	0.87513	1325797.86
Θ_2	0.88456	1251936.23
Θ_3	0.90215	1178151.10
$M_{2->1}$	0.64889	3212601.80
$M_{3->1}$	0.63842	3735886.06
$M_{1->2}$	0.62695	3456407.75
$M_{3->2}$	0.58838	3624469.48
$M_{1->3}$	0.65747	3546529.66
$M_{2->3}$	0.64678	3024254.09
Ln[Prob(DIG)]	0.78159	2235300.81

10.2 Run time and accuracy

If you have looked in the menu `Search Strategy` then you saw that we distinguish between short and long chains. Since the MCMC process is going from a not so good estimate (the first guess, you specify in `Start values for Parameters`) to a better estimate along a "gradient" on the likelihood surface, the success in recovering the best parameters is driven by the steepness of this surface. This means if there is few information in the data, the likelihood surface will be flat and the estimation process need a long time to wander to a peak (if at all) . The short chains allow for a burn-in period in which the the trees and the parameters can equilibrate, for the final estimate we use only the last of the long chains. The necessary length of these chains is specified by the number of individuals, length of sequences and variability of the data. There are no good estimates what a good length for the final chains should be

For MIGRATE it seems that in simulated datasets with around 20 individuals and 10 "electrophoretic" loci the truth can be recovered.

During my simulations for the paper on MIGRATE (*Beerli* and *Felsenstein*, 1999), I detected problems with the accurate estimation of the migration rate with start to be obvious with very long sequences (say above 1000bp). The first tree is constructed using an UPGMA topology and a Fitch algorithm to insert the migrations. This process will insert a minimum of migrations onto the tree. If now the sequences define a good topology for your guessed start parameters the program will tend to be stuck with this starting tree. This is fine for estimating the population size, but the migrations are not well distributed on the tree. I recommend that you run longer chains and watch the acceptance-rejection, if the program finds about 200 new trees for short chains and about 2000 trees for long chains or more then the estimation process should be fine. If in your initial run you see acceptance ratios of only around 2% you should definitely increase the length of the chains, or use the option **moving-steps**. When after some runs you see that the program returns hugely different values, for example the profile likelihood curves exclude the parameter estimates of other runs, you should also consider running multiple chains at different temperatures or use replication (see **Search Strategy**). Most likely, there are sets of genealogies that are not that well connected and with short chains the program will settle in one solution. Currently there is no way to check which of the independent runs fits the data better because the reported likelihoods are relative and not absolute and this makes it impossible to compare different runs.

10.3 Quick guide for achieving "good" results with `migrate`

Of course this is not a fool proof guide, then it's easy to give advice with data simulated using the same sequence model as the inference program.

FIRST: make sure that your data is correct. Miscounts of individuals, sequence length,

number of loci etc can produce funny errors.

- Set parameters in the **Search Options** to very low values, e.g to something below 100 for sampling increment and the chains to something like 2, also Turn off the profile and plot option, but set `print the data` in the **Input/Output** menu.
- run the program an check if the number of individuals read is correct, and if all the data was read, and if the program produces numbers in the output. If the program crashes before the menu there is an error in the `parmfile`, if it crashes shortly after the menu most likely there is some error in the infile. If it crashes at the end, most likely there is a programmer's bug :-(.
- Once it is clear that the program is able to run, use the default options to start a first run. If you have written a `parmfile` you should rename or destroy it.

Monitor the progress by looking at the intermediate parameter estimates:

- Check the log on the screen or the logfile, if the data-likelihood of the start tree for each chain is always improving then consider to lengthen the increment between the sampled genealogies (e.g. `short-inc=100`) or supply your own distance matrix (`distfile` option), or give own starting values or run more short chains (e.g. `short-chain=20`).
- Gelman's convergence criterium: My implementation of this criteria is not completely correct, then MIGRATE is using two consecutive chains to calculate the criterium, whereas Gelman used chains with "overdispersed" starting points. If the values are close to 1 (Gelman uses $R < 1.2$) then we can assume that the chains are sampling from the stationary distribution and that our parameter estimates are OK, but of course, this is no guarantee for success then when the sampler is sampling only around one probability mountain and does not know that another much higher mountain exist, the results will be wrong.

But, besides monitoring progress, I would:

- Run MIGRATE with the default values using F_{ST} to find the start parameters.
- Rerun, using the obtained parameter estimates of the last run. Be careful not to take this advice too literally: start parameters of zero (0.0) are very bad starting points for parameters where you expect nozero values, if the preliminary run suggests a parameters is zero, use some arbitrary value: for example for Θ and DNA data I would use 0.005
- If the results do not change much , perhaps you can stop. Otherwise increase the length of the chains, increasing the increment (e.g. **long-inc** does not increase memory usage, but run-time. You can also increase the number of sampled genealogies (**long-sample**). E. g. increase it by a factor of 10.
- Change the random number seed and check if you get similar results.
- Use the heating scheme if you get wildly different results and have low acceptance ratios.
- Run with `replicates=YES:10` and perhaps also using `randomtree=YES`, but beware this will run 10x longer then your single run.

- Microsatellite and Electrophoretic data should experiment with lowering the number of sampled genealogies (if they have many loci), because otherwise the runs will take forever, try to run migrate on a parallel machine (based on MPI) that would distribute the loci onto different machines, read the chapter "Parallel migrate".

11 Presentation of results

11.1 Maximum likelihood inference

There are several differences between the Maximum likelihood analysisand Bayesian analysisoutput. The outfput exists typically in two files a textfile called outfile (default name) and a PDF called outfile.pdf, for changing these names consult the input/output menu. The maximum likelihood analsysi writes to a PDF file but because of time constraints I never completely finished that transition (perhaps next year), therefore for Maximum likelihood analysisuse the textfile as the main output, EXCEPT if you are interested in the distribution of the migration and coalescence events through time, that is only plotted into the PDF (see below).

Contents of the output in `outfile`: Some of the output options vary according to the datatype. $+$ = always present, o = optional, Default = $\star$

Item	Description	Status
List of options	all used options are specified	$+$
Summary of data	(Too) short data summary	$+$
Dataset	Print of the dataset	o
MCMC estimates	List of the estimated parameters for each locus and the mean	$+$
Shape α	Estimation of the shape parameters α for the variation of the mutation rate	o
F_{ST}table	Table of the possible start values generated with a F_{ST}estimator	o
plots	plot of the likelihood surface in outfile	o$\star$
	plot of the likelihood surface into mathfile	o
α-histogram	Table of shape values versus log(likelihood), α is varying whereas the other parameters are held constant at the maximum of the surface.	o
Profiles	Profile likelihood tables	o$\star$
Percentiles	Percentiles table, summary of profile tables	o$\star$
Event histograms	Distribution of events over time	o

The F_{ST} calculations are based on mean differences in populations compared to mean differences between populations, for more information you should consult *Maynard Smith* (1970); *Nei* and *Feldman* (1972); *Beerli* and *Felsenstein* (1999), .

11.1.1 Walk through an outfile

The following output pieces are from `outfile.seq` in the `example` directory.

Title and Options

```
 ============================================
   An example with sequence data
 ============================================
  MIGRATION RATE AND POPULATION SIZE ESTIMATION
  using Markov Chain Monte Carlo simulation
  ============================================
  Version 2.0.3

  Program started at Sat Dec 18 19:56:23 2004
         finished at Sun Dec 19 01:22:31 2004

Options in use:
---------------
Datatype: DNA sequence data
Random number seed (with internal timer)          1103417783
Start parameters:
   Theta values were generated  from the FST-calculation
   M values were generated from the FST-calculation
Migration model:
   Migration matrix model with variable Theta
Mutation rate is constant for all loci
Analysis strategy is                          Maximum likelihood
Markov chain settings:
   Short chains (short-chains):                          10
      Trees sampled (short-inc*samples):              20000
      Trees recorded (short-sample):                   1000
   Long chains (long-chains):                            3
      Trees sampled (long-inc*samples):              200000
      Trees recorded (long-sample):                   10000
   Averaging over replicates:                             2
   Static heating scheme
      4 chains with  temperatures
       1.00, 1.57, 2.71, 5.00
      Swapping interval is 1
   Number of discard trees per chain:                 10000
Print options:
   Data file:                               infile.check-mig
   Output file:                                      outfile
   Print data:                                            No
   Print genealogies:                                     No
   Plot data: Yes, to outfile and mathfile
              Parameter: {Theta, M}, Scale: Log10, Intervals: 36
              Ranges: X-    M: 0.000100 - 100.000000
              Ranges: Y-Theta: 0.000100 - 100.000000
   Profile likelihood: Yes, tables and summary
             Percentile method
             with df=1 and for Theta and M=m/mu
```

This is the title and options part. Don't cut away the options, so you will still know a few weeks later with what kind of options and how long you run the program.

Summary of the data

```
Summary of data:
---------------
Datatype:                                        Sequence data
Number of loci:                                              2

Population                                      Locus  Gene copies
-------------------------------------------------------------------
  1 Tallahassee                                   1     20
                                                  2     20
  2 Sopchoppy                                     1     20
                                                  2     20
  3 St._George Island                             1     20
                                                  2     20
Total of all populations                          1     60
                                                  2     60


Empirical Base Frequencies
-----------------------------------------------------------
Locus     Nucleotide                          Transition/
          ------------------------------  Transversion ratio
          A       C       G       T(U)
-----------------------------------------------------------
   1      0.2515  0.2730  0.2283  0.2472       2.00000
   2      0.2465  0.2350  0.2627  0.2557       2.00000
```

The data summary is (too) short, and self explanatory, you can also print the data (not shown). Print the data the first time you use the program with your data and check if it was read correctly: I control the first and the last individual in a population and check a few sites at both ends of the sequence. If the program crashes shortly after the start, almost certainly the data contains some trouble. The most common error is having the wrong number of individuals and/or number of sites, or having miscounted the number of characters in the individual name.

Parameter estimates

```
=============================================================================
MCMC estimates
=============================================================================
Population [x] Loc.  Ln(L)   Theta    M [m/mu] [+=receiving population]
                             [xNe mu]    1,+     2,+     3,+
-------------- ---- -------- -------- ------------------------------------
 1: Tallahasse  1 1   11.406  0.04326 ------- 22.9291  0.0000
                1 2    0.875  0.04006 -------  0.0000  0.0000
                1 A    1.754  0.04009 -------  0.0000  0.0000
                2 1    2.463  0.03418 -------  0.0000 11.8760
                2 2    3.246  0.04276 -------  4.1693 12.4264
                2 A    4.158  0.04214 -------  3.8056 12.1477
                All   20.696  0.04330 -------  8.6572  0.0000
 2: Sopchoppy   1 1   11.406  0.01553  5.6584 -------  3.2363
                1 2    0.875  0.01996 12.0396 -------  0.0000
                1 A    1.754  0.01993 12.0679 -------  0.0000
                2 1    2.463  0.00918  0.0000 ------- 14.9951
                2 2    3.246  0.01485  0.0000 ------- 16.6994
                2 A    4.158  0.01444  0.0000 ------- 16.5949
                All   20.696  0.01283  0.0000 -------  2.0536
 3: St._George  1 1   11.406  0.00969  0.0000  0.0000 -------
                1 2    0.875  0.01125  0.0000  5.8414 -------
                1 A    1.754  0.01124  0.0000  5.8325 -------
                2 1    2.463  0.01174 13.0240  0.0000 -------
                2 2    3.246  0.01025 20.5578  0.0000 -------
                2 A    4.158  0.01039 19.7503  0.0000 -------
                All   20.696  0.01088  3.4871  3.2021 -------

Comments:
 The x is 1, 2, or 4 for mtDNA, haploid, or diploid data, respectively
There were 10 short chains (1000 used trees out of sampled 20000)
  and 3 long chains (10000 used trees out of sampled 200000)
  Static heating with 4 chains was active
  COMBINATION OF 2 MULTIPLE RUNS)
```

This is the main output of the program. For each population there is a list of all estimates for each locus and each replicate and their over-all-replicate and over-all-loci estimates. The replicate summary estimates are not simple averages but use a method devised by Geyer (1994: reverse logistic regression). The summary over loci is summing up the likelihood curves under the assumption that each locus is independent of the other (a rather save assumption as long one is not working with multiple mtDNA or Y-chromosome loci).

11.2 Bayesian inference

11.2.1 Walk through an outfile

The main output of a Bayesian run contains of the following table that summarizes the posterior distribution and an acceptance ratio table. The table of the posterior distribution is characterized for each locus and each parameter and percentiles, median, mode, and mean. The posterior distribution over all loci is also presented graphically (Figure 11.1).

Bayesian Analysis: Posterior distribution table

Locus	Parameter	2.5%	25.0%	Mode	75.0%	97.5%	Median	Mean
1	Θ_1	0.08961	0.11681	0.14801	0.17921	0.29281	0.16721	0.17748
1	Θ_2	0.00001	0.00321	0.01361	0.02401	0.13441	0.02161	0.07699
1	Θ_3	0.00001	0.00001	0.01361	0.02401	0.54560	0.02481	0.12032
1	Θ_4	0.00641	0.02241	0.03441	0.04321	0.07201	0.03761	0.03801
1	$M_{2->1}$	0.0	0.0	1.0	6.0	18.0	7.0	2.6
1	$M_{3->1}$	0.0	0.0	1.0	6.0	16.0	7.0	1.7
1	$M_{4->1}$	0.0	0.0	1.0	6.0	16.0	7.0	2.1
1	$M_{1->2}$	0.0	6.0	29.0	56.0	182.0	51.0	64.6
1	$M_{3->2}$	0.0	0.0	1.0	52.0	518.0	53.0	150.2
1	$M_{4->2}$	0.0	0.0	1.0	56.0	406.0	57.0	159.8
1	$M_{1->3}$	0.0	0.0	1.0	20.0	88.0	21.0	27.3
1	$M_{2->3}$	0.0	0.0	1.0	36.0	268.0	37.0	158.2
1	$M_{4->3}$	0.0	0.0	23.0	160.0	454.0	161.0	291.1
1	$M_{1->4}$	0.0	0.0	1.0	12.0	38.0	13.0	11.6
1	$M_{2->4}$	0.0	0.0	1.0	18.0	94.0	19.0	26.4
1	$M_{3->4}$	0.0	0.0	1.0	44.0	160.0	45.0	56.6
1	$\mu\ [10^{-4}]$	3.23685	5.03505	6.60848	7.73235	9.08100	7.05802	6.54878

Bayesian Analysis: Posterior distribution over all loci

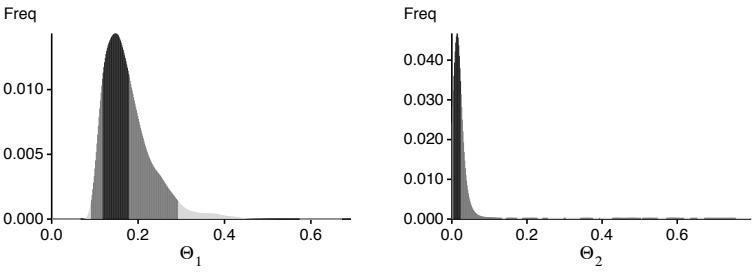


Figure 11.1: Table and Figure example of a Bayesian posterior distribution.

11.3 Histograms over time

11.3.1 Events through time

MIGRATE allows to investigate the pattern of events through time, the histograms represent the frequency of recorded events during the MCMC run, the location of these events in time are determined by the data (that is what we want to see) but depends on the length of runs, and how well the genealogies were explored (that is what we want to have no influence!). MIGRATE is assuming the all the events in every time units come from the same prior distribution (BA) or driving value (MA). For simulated data from populations that are constant in size through time and that exchange migrants at a constant rate, we expect distributions that look similar exponential decay. If either the data was generated by a process that is not constant through time the histogram will look different (figure 11.2). The time is measure in unites of generation / mutation rate per generation (and site) . MIGRATE also prints out tables that report average time

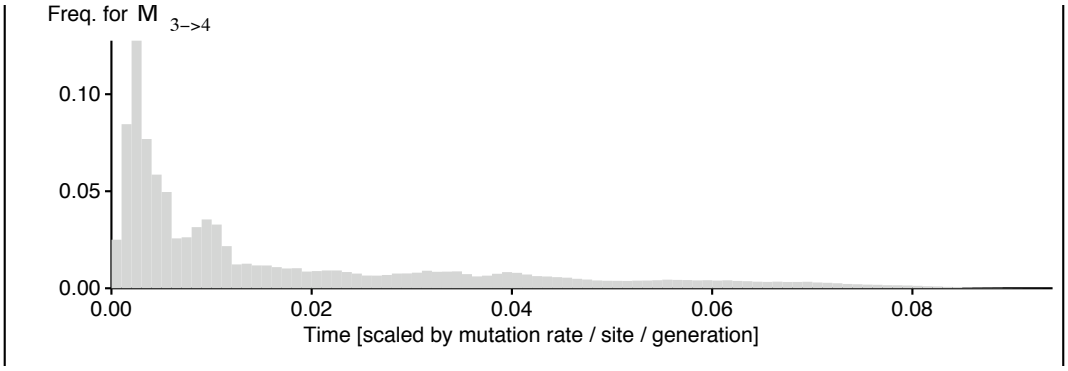


Figure 11.2: Frequency of migration events between between two populations through time. Today is left on the graph; units are generation per mutation rate

for migration and coalescence events for all events and for the most recent common ancestors, and that supplies the probability in which population the sample originated (Figure 11.3). This seem to work fine with equal sample sizes , but may be skewed with unequal sample size (for example for two populations: 100 and 10). Unequal sizes may need much longer run time to say some thin with confidence. In addition, a single locus may not give really relevant results, use multiple loci if you can.

11.3.2 Skyline plots

MIGRATE has its own version of skyline plots *Strimmer* and *Pybus* (2001); *Shapiro* et al. (2004); *Drummond* et al. (2005). MIGRATE reports averages and standard deviation of expected parameter values calculated from the genealogy. The proposal for all timeintervals uses the constant population size and migration rate, so it is different from *Drummond* et al. (2005) and certainly

Summary statistics of events through time

Locus 1 Population		Time			Frequency
From	To	Average	Median	Std	
1	1	0.013473	0.010500	0.010571	0.497562
2	2	0.012101	0.011500	0.006743	0.035628
3	3	0.007849	0.005500	0.006085	0.050622
4	4	0.005770	0.003500	0.008057	0.194836
2	1	0.024648	0.021500	0.018571	0.006902
3	1	0.029659	0.020500	0.020595	0.002664
4	1	0.025750	0.014500	0.020790	0.004337
1	2	0.028152	0.023500	0.016714	0.009083
3	2	0.018129	0.010500	0.017686	0.019865
4	2	0.018140	0.010500	0.018299	0.024827
1	3	0.035386	0.033500	0.018594	0.002991
2	3	0.015806	0.007500	0.016983	0.027621
4	3	0.015217	0.007500	0.016953	0.063453
1	4	0.033605	0.025500	0.018438	0.004742
2	4	0.021297	0.013500	0.021340	0.016607
3	4	0.017979	0.008500	0.020123	0.038260

Time and probability of location of most recent common ancestor

Locus 1 Population	Time			Frequency
	Average	Median	Std	
1	0.081112	0.081500	0.004877	0.708145
2	0.075595	0.073500	0.007884	0.005840
4	0.078512	0.078500	0.005191	0.286015

Figure 11.3: Left: Tables of frequencies and average time for all events. Right: Table of the probability of the location of the most recent common ancestor.

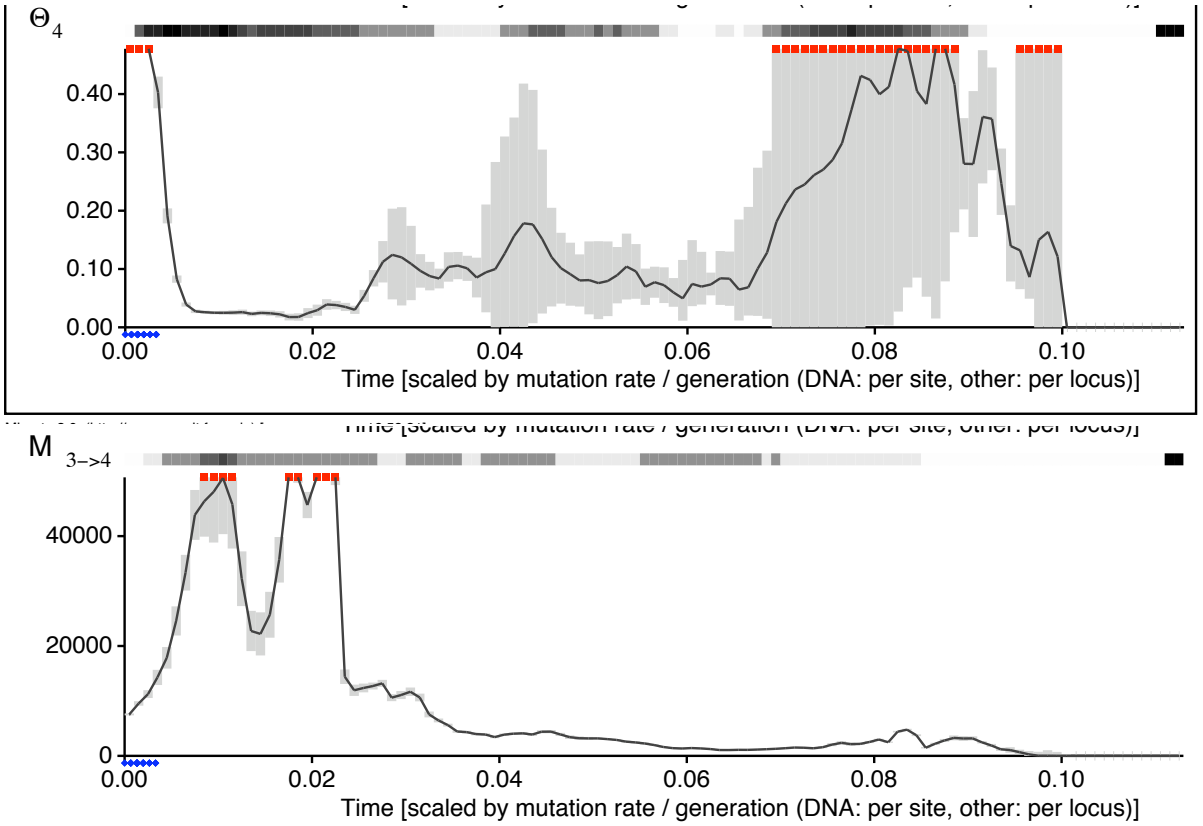


Figure 11.4: Skyline plot of a population that recently increased strongly , the time is in units of mutation-scaled generations. Top: population size, bottom: one example immigration rate into the population shown on top.

needs more evaluations. MIGRATE can summarize over multiple loci, take into account several data types, and reports the parameters changes through time also for migration parameters. MIGRATE has several short-comings: for example it assumes that the mutation rate is constant per locus, which make affect results for some data sets, but because MIGRATE is typically used for populations within species or very closely related species, I hope that the mutation rate of a specific locus will not change considerably.

A legend for these plots is printed toward the end:

```
Skyline plots:
Skyline plots visualize the changes of population sizes and migration rates through time
(today is on the left side and time is measured into the past. The time scale is in units of
expected mutations per generation. To calculate the absolute time scale you must supply an
mutation rate per year and the duration of a  generation in years in the data option.
You can calculate the absolute time by multiplying the scale by generation time times
mutation rate per year (per site for DNA; per locus for all other datatypes).
With estimated mutation rate only the combined rate modifier is plotted.
[this will change to  mutation rate plot].
The gray bars cover one approximate standard deviation up and down from the expected value.
The bar with different shades of gray on top of each plot indicates the number of values that
to calculate the expected value, white means there are very few and black means.
that there were man thousands of samples per bin.
On some plots one can see red squares below the grayscale bar, these suggest that either the
upper quantile and/or the main value was higher than the visible part of the  axis.
Event histograms:
All accepted events (migration events, coalescent events) are recorded and their frequency
are shown as histograms over time with recent time on the left side. The frequency plots of
populations with constant size and constant immigration rates show histograms that are simila
to exponential distribution, if the populations come from a divergence model without migratio
then the frequency of migration events can show a peak in the past.
```

12 Output that is not part of the outfile

MIGRATE writes the raw data that is used to generate the histograms and tables in the PDF and the textfile into several files, such as the *bayesfile*, *bayesallfile*, *mighistfile* and the *skylinefile*. Each file contains a header that gives you some idea what the values mean and you can process these files by yourself using graphing programs (or TRACER). I highlight here a use of the the print-tree option.

12.1 Potential genealogy plots

MIGRATE allows to record the best genealogy visited in the course of the MCMC run, this treefile contains migration events and currently only the program `eventtree` (ET) (Palczewski and Beerli unpubl. – popgen.scs.fsu/et) can plot these events. Remember this is not necessarily the best possible tree for the data, but the most likely visited tree, in tests with small dataset we could show that with real species tree MIGRATE recovers the topology, but because it does not optimize branch length, will make errors on the length of the branches, it also assumes a clock.

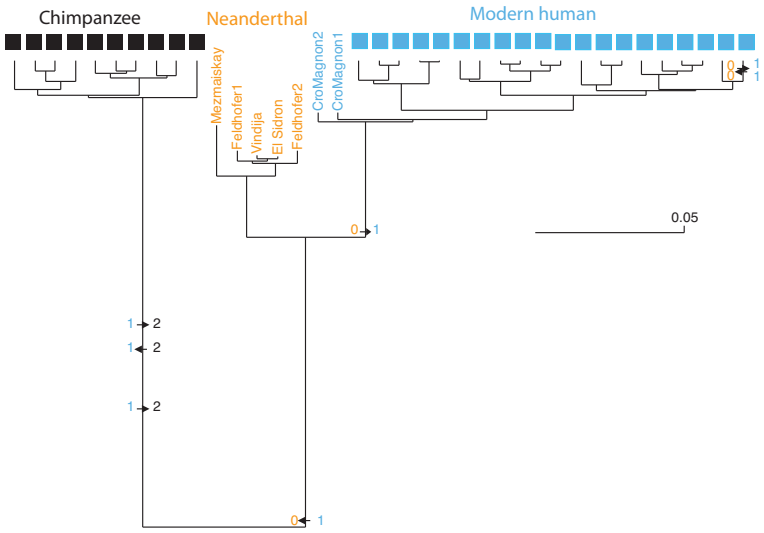
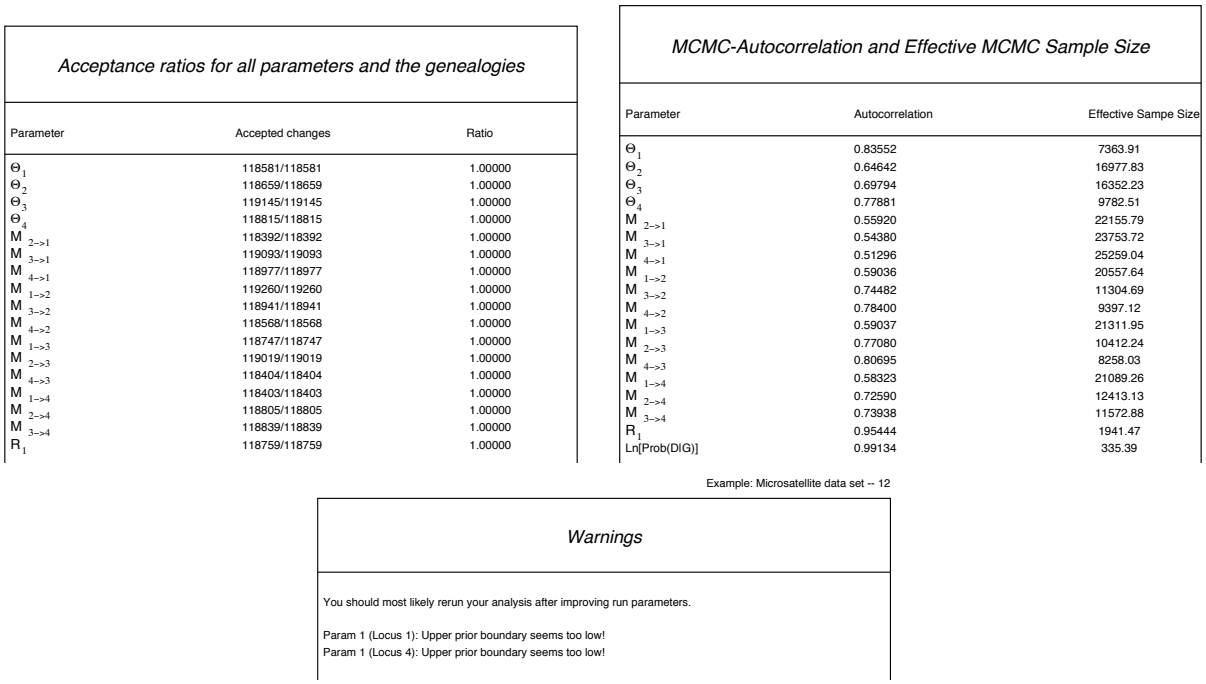


Figure 12.1: Best visited genealogy of a 3 population run with Neanderthals, modern humans, and chimpanzees. The arrows on the tree mark migration events – there is very little power to pinpoint these migration events and the events shown are a haphazard sample of many possible migration events that happen to occur on the topology that is most compatible with the data, The color was added using Adobe Illustrator.

13 Diagnostics

MIGRATE prints out several diagnostics, these diagnostics are not sufficient to judge whether your data was run successfully, but you should run the program minimally two times to compare the results and not trust the diagnostics. The acceptance/rejection ratios for all parameters (BA) and the genealogy (MA, BA) give some idea about how many new parameters or trees are in the MCMC sample, if the ration is very low the autocorrelation will be high and the effective sample size of trees and parameters will be low. For MA a statistic described by Gelman and Rubin *Kass* et al. (1998) can be used to get some idea about convergence. The Gelman-Rubin statistic is broken for some of the analysis option, but I believe that multiple runs from different start settings (different start parameter and random tree) are a great way to explore the behavior of the MCMC run(s).

The last page of the output can contain **Warnings** that suggest whether some parameters did not converge or not (Figure 13.1).



Acceptance ratios for all parameters and the genealogies

Parameter	Accepted changes	Ratio
Θ_1	118581/118581	1.00000
Θ_2	118659/118659	1.00000
Θ_3	119145/119145	1.00000
Θ_4	118815/118815	1.00000
$M_{2->1}$	118392/118392	1.00000
$M_{3->1}$	119093/119093	1.00000
$M_{4->1}$	118977/118977	1.00000
$M_{1->2}$	119260/119260	1.00000
$M_{3->2}$	118941/118941	1.00000
$M_{4->2}$	118568/118568	1.00000
$M_{1->3}$	118747/118747	1.00000
$M_{2->3}$	119019/119019	1.00000
$M_{4->3}$	118404/118404	1.00000
$M_{1->4}$	118403/118403	1.00000
$M_{2->4}$	118805/118805	1.00000
$M_{3->4}$	118839/118839	1.00000
R_1	118759/118759	1.00000

MCMC-Autocorrelation and Effective MCMC Sample Size

Parameter	Autocorrelation	Effective Sampe Size
Θ_1	0.83552	7363.91
Θ_2	0.64642	16977.83
Θ_3	0.69794	16352.23
Θ_4	0.77881	9782.51
$M_{2->1}$	0.55920	22155.79
$M_{3->1}$	0.54380	23753.72
$M_{4->1}$	0.51296	25259.04
$M_{1->2}$	0.59036	20557.64
$M_{3->2}$	0.74482	11304.69
$M_{4->2}$	0.78400	9397.12
$M_{1->3}$	0.59037	21311.95
$M_{2->3}$	0.77080	10412.24
$M_{4->3}$	0.80695	8258.03
$M_{1->4}$	0.58323	21089.26
$M_{2->4}$	0.72590	12413.13
$M_{3->4}$	0.73938	11572.88
R_1	0.95444	1941.47
Ln[Prob(DIG)]	0.99134	335.39

Example: Microsatellite data set -- 12

Warnings

You should most likely rerun your analysis after improving run parameters.

Param 1 (Locus 1): Upper prior boundary seems too low!
Param 1 (Locus 4): Upper prior boundary seems too low!

Figure 13.1: Acceptance Ratios, Effective sample size and autocorrelation, and Warnings of a run.

14 Installation

14.0.1 Binaries

On UNIX system unpack with `tar xvfz migrate.[system].tar.gz` or `gunzip -c migrate5.1.[system].tar.gz | tar xf -`. This builds a directory `migrate-5.1` with a subdirectory `examples`, the files `README, HISTORY`, and the programs `migrate` and `migrate-n`. The program can be moved to a location like `/usr/local/bin` and the documentation (HTML files are in documentation/migratedoc) to your HTML directory (e.g. `/usr/local/etc/httpd/htdocs`). On Powermacs or Windows machines double click the archive and a folder system similar the UNIX directories above will be created.

14.0.2 Source

The program is known to compile on every UNIX machine that has a decent ANSI compatible compiler. And on the following non-UNIX machines: INTEL (Windows 2000, xt, vista?).

UNIX (Linux, BSD Unix, MacOSX)

1. **gunzip -c migrate5.1.tar.gz** | **tar xf** - or
   tar xfz migrate5.1.tar.gz this creates a directory "migrate-5.1" with "src", "contribution", and "examples" in it.
2. `cd migrate-5.1`
3. `./configure`
   (this scripts checks your system and will report functions the program needs, if a function is not, it will report an error, which I need to know. I assume that your machine has `gcc` installed, but `configure` tries to be smart about other compilers: on SGI and DEC ALPHA without `gcc` it will use the native cc compiler with the approriate options. You can force this behavior with bash shell: CC=cc ./configure, in csh shell: env CC=cc ./configure
4. `make`
   (please report warnings and especially errors) This produces an optimized binary for your computer, if your computer has multiple CPUs or core, you can try to compile using `make thread`, this produces a binary that can use multiple processors for heated runs, this is good but parallel runs on such computers use more CPU cycles.

The result should be a binary `migrate` in the migrate directory. If you have a multiprocessor machine that has the POSIX thread library installed (the configure script searches for libpthread and pthread.h) try to use `make thread`, this will allow to run the heated chains in parallel and so should speed up the program if you use heating.

5. `make install`
(this will install the program and man-page into usr/local/bin, /usr/local/man/man1 ; you need to be root to do this; this step is not necessary)

15 Parallel migrate

This text describes how you can improve the performance of MIGRATE when you have more than one locus and more than one computer at your fingertips. You can parallelize migrate runs (1) using a virtual parallel architecture with a message-passing interface (MPI) or (2) by hand. The hand-version works but is cumbersome, the MPI-version runs fine on clusters of MacOSX workstations, dedicated clusters of Linux machines, AIX parallel machines (Regatta; SP3, SP4).

15.1 I. Using the standard Message passing interface (MPI)

1. Secure as many computers for the analysis as you have loci or parameters in your dataset. Make sure that all computers can talk to each other. Currently my program will only work if they are a flavor of UNIX (e.g. LINUX or MACOSX). Of course, you need an account on all the machines.

   - Download OpenMPI from http://www.openmpi.org [I use version 1.2.5] (Macintosh computers with the Leopard operating system – MacOS 10.5 have this already built-in, use `fastmigrate-n`)
   - install on all machines (if this is to complicated for you ask a sysadmin or other guru to help:
     `./configure` There are several options that may or may not be helpful in your environment
   - prepare a file "hostsfile" according to the specs in the openmpi distribution, the master node needs to be the first machine mentioned. my "hostsfile" looks like this:
     ```
     ciguri node=2
     zork node=1
     nagual node=32
     ```
   - `make sure that you can access all machines [using ssh] without the need to specify a password, see man ssh-keygen and man ssh if you have firewalls installed on your individual systems then you would need to allow the individual machines to open/request "random" ports on the other machines. On MacOSX machines this is a common problem because the machines may run local firewalls.`

- `change into the migrate-2.4/src/ directory configure and then use "make mpis-pretty"`

2. If your machines have no cross-mounted file system, you need to make sure that the program is all in the same path e.g. /home/beerli/migrate-test/migrate-n and on EVERY machine.
3. Compile migrate, you need to follow the instructions in README. Essentially you need to do
   `./configure`
   `make mpis`

   The configure command sets up the Makefile etc. `make mpis-pretty` compiles for parallel machines.
4. Try run the following command from the src/example directory

   - `mpirun -np 7 -host hostfile ../migrate-n parmfile.testbayes -nomenu` [6 loci will be analyzed at at the same time, the log is not very comprehensive because all 7 processes write to the same console, 7 because there is one master-node who does only scheduling and summarizing, 6 worker-nodes do the actual tree rearrangements and the likelihood calculations. the number you specify has nothing to do with the physical computers, OpenMPI can run several nodes on a single CPU but best is to use not more nodes than there are CPU-cores.
   - send comments how it worked for you and improvements for my [currently] too short and confusing guide.

16 Frequently asked questions

This section will increase when I get more feedback. The order of the questions/answers is probably random or historical.

16.1 Questions

16.1.1 General

1. I cannot find the program executable? I double-click the program icon but nothing happens?
2. The program crashes! Your program has a bug!
3. The program crashes with large but not with small data sets, what is wrong?

16.1.2 About the datafile

1. I need more input about how microsatellites are coded in migrate?
2. How can I code haploid data for MIGRATE?
3. Can I use haplotype frequencies as input?
4. Can I use gene frequencies as input?

16.1.3 About options and how to run

1. It runs with the default number of chains etc. Has it run long enough?
2. How long does it run?
3. Can migrate run on multiple machines in parallel?

16.1.4 About reading the outfile

1. I have haploid data, what is Θ?

2. I have mtDNA sequence data, what is Θ?
3. Why are the Likelihood values different between runs?
4. Why do I have positive numbers in the Ln(L) column?
5. I have problems to understand what are the Null-hypothesis and the alternative hypothesis in the likelihood ratio test section.
6. I run migrate several times and get inconsistent estimates.
7. I run migrate and the population sizes are strangely high.
8. I run MIGRATE using Θ and M parameters but I want to calculate $2Nm$ (my data is a haploid lichen)?

16.2 Answers

16.2.1 General

1. **I cannot find the program executable? I double-click the program icon but nothing happens?**
   Some binary distributions contain migrate-n as command line tool and they need to be started from a Terminal program [or shell]. A typical migrate run on Macosx operating systems involves to start of the Terminal.app (for tutorials about this see `http://www.macdevcenter.com/pub` and then change to the directory where the data resides, and then start `migrate-n`.
2. **The program crashes! Your program has a bug!**
   Sure, this program most likley has some bugs, but more likely is that the `infile` is not correct, and without more detail about what went wrong there is little hope for help.
3. **The program crashes with large but not with small data sets, what is wrong? [System description... + part of log]**
   - General: Most often mistakes in the `infile`, such as wrong number loci or populations or individuals or number of sites or using few characters for the individual names, let the program crash almost immediately after the menu. Check the infile carefully and compare with the data file specifications.
   - on Macintoshes: the preferred memory consumption of `migrate` is set to 20MB RAM, for larger problems, such as many populations or many loci or long chains, this can produce cryptic crashes (e.g. `Error in calloc() in file broyden.c line xxx`). Try increase the memory. You single-click the icon of `migrate`, go to the `File` menu and choose `Get Info` and in there `Memory`. Set the `preferred Size` to some higher value. If you have 128 MB RAM and your System is consuming already around 30 MB, you can set the program up to something like 80 MB, but

then if you run other programs it will swap parts of the RAM into virtual memory. If some part of `migrate` are swapped into onto disk by the virtual memory manager the program will most likely not finish because because the program is slowed down to a crawl.

- on Windows: I need to know about this, but so fare the latest binaries, seem to have no trouble with preset memory.

16.2.2 Data file related

1. **I need more input on how to code microsatellites for migrate** Assume you analyze two individuals for two msat loci (locus1 is a CA repeat, locus 2 a TCA repeat) that look like this

```
     Primer 1     msat locus 1
     -------------              ------
1A   ATTAGACATTGTGCACACACACACACATTGGAC
1B   ATTAGACATTGTGCACACACACACACATTGGAC
2A   ATTAGACATTGTGCACACACACACACATTGGAC
2B   ATTAGACATTGTGCACACACA------TTGGAC

     Primer 2     msat locus 2
     -------------              ------
1A   ATTAGACATTGTGTCATCATCATCATCATCATCATCATCATCATCATCATCATCATTGGAC
1B   ATTAGACATTGTGTCATCATCATCATCATCATCATCATCATCATCATCA---TTGGAC
2A   ATTAGACATTGTGTCATCATCATCATCATCATCATCATCA------------TTGGAC
2B   ATTAGACATTGTGTCATCATCATCA------------------------------TTGGAC
```

in a migrate infile this would code (where / is a free chosen delimiter you can choose any character, but I recommend something like / . , or similar)

```
  1 2 / Example of how to code migrate
2 Example population with 2 diploid indiviudals
1A1B______ 7/7 14/13
2A2B______ 7/4 10/4
```

2. **I have haploid allelic data, how should I structure my infile** Unfortunately, I was biased towards diploid data for microsatellite and enzyme electrophoretic data and you need to fake diploids for the infile. Your microsatellite exampled data look like this:

```
      Locus1 Locus2 Locus3 Locus4 Locus5
Ind1   11      45     14     15     89
Ind2   11      47     13     15     67
Ind3   11      43     13     15     67
Ind4   12      47     13     15     73
Ind5   11      45     13     15     89
```

And your infile should look like this

```
   2  5  . Example input for haploid microsatellite data
5  Fake diploid population 1
Ind1      11.?      45.?     14.?     15.?     89.?
Ind2      11.?      47.?     13.?     15.?     67.?
Ind3      11.?      43.?     13.?     15.?     67.?
Ind4      12.?      47.?     13.?     15.?     73.?
Ind5      11.?      45.?     13.?     15.?     89.?
4 Fake diploid population 2
..data not shown..
```

Or

```
   2  5  . Example input for haploid microsatellite data
3  Fake diploid population 1
Ind1Ind2   11.11 45.47 14.13 15.15 89.67
Ind3Ind4   11.12 43.47 13.13 15.15 67.73
Ind5????   11.?  45.?  13.?  15.?  89.?
4 Fake diploid population 2
..data not shown..
```

The "?" are removed for the analysis (But recognize that in sequence data the ? are not removed.

3. **I have triploid (polyploid) allelic data, how should I structure my infile** Unfortunately, I was biased towards diploid data for microsatellite and enzyme electrophoretic data and you need to fake diploids for the infile. Your microsatellite example data look like this:

```
           Locus1     Locus2
Ind1   11.11.12   45.45.45
Ind2   11.12.12   47.45.45
Ind3   11.10.10   43.45.?
Ind4   12.12.12   47.45.47
Ind5   11.11.10   45.45.43
etc.
```

And your infile should look like this

```
   2  2  . Example input for triploid microsatellite data
5  Fake diploid population 1
Ind1      11.11   45.45
Ind12     12.11   45.45
Ind2      12.12   47.45
Ind3      11.10   43.45
Ind34     10.12    ?.47
Ind4      12.12   47.45
Ind5      11.11   45.45
Ind5x     10.?    43.?
4 Fake diploid population 2
..data not shown..
```

4. **Can I use haplotype frequencies as input?** No, input formats are a rather arbitrary matter, and I decided that you need to input each single sequence

of genotype. In principle, it would be easy to add a "frequency" input mode, but currently I have no time to do that. But keep asking for it, if this is so important to you.

5. **Can I use gene frequencies as input?** No, not yet, this is on the todo list, but has a rather low priority. To circumvent the problem, you can create artificial genotypes for the infile. The genotypes themselves are not important. A simple script that assigns alleles to individuals will do, this can be written in almost any scripting language from excel (yikes!), word-macro (yikes!), Perl, C, C++, applescript, Mathematica, ... for throw away programs I use Python[1], Mathematica[2], or C[3].

16.2.3 About options and how to run

1. **It run with the default number of chains etc. Has it run long enough?**
   this depends on the number of populations you want to analyze. If you have one it will be almost certainly enough. But if you try to analyze 6 or more it almost certainly will not. You need to experiment a little with the length of chains. See chapter 3 (Accuracy of results).

2. **How long does it run?**
   With `progress=Yes` [do not use progress=Verbose] the program tries to estimate the length of a run from the work it has done so far, after the first short chain this may be rather imprecise, but you may realize that you need to wait minutes or days (just imagine you estimate the time to travel from Spokane to Seattle in a car and estimate when you will arrive only using the distance and time you have finished already). The time calculated is only based on the genealogy search, and does not include the time to create the plots for each locus and population. Therefore, if you have many populations and many loci you can expect to wait longer than the time stamp indicates. There is an additional time estimate for the profile-likelihoods.

3. **Can migrate run on multiple machines in parallel?**
   Short answer: YES. **Long Answer 1:** If you use the `heating` option and your machine is a symmetric multiprocessor machine and you compiled with `make thread` or `make` on MacOS 10.6 then the program will utilize n processors. This will improve the heated search by about a factor of n , also the performance degrades somewhat the more threads are running concurrently. **Long Answer 2:** Yes, on UNIX systems (inclusive MacOSX) you can use a parallel virtual machine, for example OpenMPI (see their website: http://www.openmpi.org) and compile migrate with "configure; make mpis" (or similar see by typing "configure") you need the MPI libraries that come with the above

[1] freely available for Windows, Mac, and UNIX, check http://www.python.org
[2] nice,but not free software
[3] freely available for almost all systems see http://www.gnu.org [Free Software Foundation]

environment (see HOWTO-PARALLEL). Or you can do it yourself manually. See the file HOWTO-PARALLEL.

16.2.4 About reading the outfile

1. **I have haploid data, do I have to multiply my Θ, $\mathcal{M}$ and $4Nm$?**
   The Θ you get with haploid data is $\Theta = 2N_e\mu$. Comparing with other values for haploid data should be fine, but you need to multiply when you compare it with a $Theta$ from diploid data.

2. **I have mtDNA data, do I have to multiply my Θ, $\mathcal{M}$ and $4Nm$?**
   See question above, but in most vertebrates mtDNA is only passing through the maternal lineages and is haploid, for a comparison with diploid data you should multiply by 4.

3. **Why are the likelihoods between runs different?**
   The likelihoods are really ratios

$$\frac{L(\mathcal{P})}{L(\mathcal{P}_0)} = \frac{1}{m}\sum_{i}^{m} \frac{\mathrm{p}(D \mid g_i)\ \mathrm{p}(g_i \mid \mathcal{P})}{\mathrm{p}(D \mid g_i)\ \mathrm{p}(g_i \mid \mathcal{P}_0)}.$$

   and we run several chains and update the $\mathcal{P}_0$ between chains. For a comparison we would need that the second last chain of each run delivers exactly the same parameters, which we then would use for the comparison. A possibility is to run only one long chain in each run with some given parameters $\mathcal{P}_0$. This not really recommended if the start values are not very close to the true parameters.

4. **Why do I have positive numbers in the Ln(L) column?**
   See also question before. the Ln(L) is actually a ratio (see Beerli and Felsenstein 1999, we have a derivation of this ratio in the appendix, but this can be found in statistics books that talk about MCMC) In our case we try to maximize

$$L(\text{parameters}) = \sum_{i}^{\text{all trees}} \text{Coalescence-Prior(tree}_i\text{| parameters)Data-Likelihood(data| tree}_i\text{)}.$$

   its MCMC derivation is

$$\frac{L(\mathcal{P})}{L(\mathcal{P}_0)} = \frac{1}{m}\sum_{i}^{m} \frac{\text{Coalescence-Prior(tree}_i\text{|parameters)}}{\text{Coalescence-Prior(tree}_i\text{|driving parameters)}}.$$

   In fact, the $ln(L)$ should be rather close to 0.0, but this is dependent on the number parameters (I think) that produce noise, with many parameter it will be not very close to 0.0, but with just one param (single population) the value is more like 0.00x, with 16 parameter it seems more like 5-30. If you have more than one locus then it is likely that when they produce rather different results, that the value will go negative.

5. **I have problems to understand what are the Null-hypothesis and the alernative hypothesis in the likelihood ratio test section?** The easiest way to answer is with an example: Assume you just run `migrate-n` and got the following results: $\Theta_1 = 0.003$, $\Theta_2 = 0.05$, $4N_1m_{21} = 0.5$, and $4N_2m_{12} = 3$. Tis assumes that you changed in the parameter setting to estimate xNm instead of $\mathcal{M}$. Before version 2.0 the default was to estimate xNm, now the default is to estimate $\mathcal{M}$. I assume that you want to test whether the population sizes are the same or not and if the migration rates m are the same or not. This would ask for a Null-hypothesis so that $\Theta_1 = \Theta_2$ and $\mathcal{M}_{21} = \mathcal{M}_{12}$ [$\mathcal{M} = m/\mu$]. Recognize that we would use here $\mathcal{M}$ and **not** $4Nm$, with your specific parameter setting, the LRT input expects Nm values. The Alternative hypothesis is then $\Theta_1 \neq \Theta_2$ and $\mathcal{M}_{21} \neq \mathcal{M}_{12}$. For this above test you can specify the LRT-input in several ways:

   - l-ratio=MLE:m, m, m, m [easiest]
   - l-ratio=MLE:0.0265, 0.0265, 3.0,3.0

   For the second example, you need to calculate by hand first the $\mathcal{M}$ and then from that recalculate the $4Nm$ when the $\mathcal{M}$ are the same, I used the averages.

   If the run would be default run with estimates $\Theta_1 = 0.003$, $\Theta_2 = 0.05$, $\mathcal{M}_{21} = 166.66$, and $\mathcal{M}_{12} = 60$. then the LRT would look like this:

   - l-ratio=MLE:m, m, m, m
   - l-ratio=MLE:0.0265, 0.0265, 113.33,113.33

6. **I run migrate several times and get inconsistent estimates.** If the profile confidence intervals of a run exclude other runs, then you should run the program longer by increasing short-inc and long-inc and short-sample and long-sample. In addition you should try to do replicates (for example replicate=YES:10) and also use heating (heating=YES:1:1,1.5,3,10000), if you still have problems I would like to hear about this. I have seen datasets were people tried to estimate several parameters with very short sequences that when run properly delivered confidence intervals with rather unwelcome confidence intervals from close to zero to very large values ($>10^{10}$).

7. **I run migrate and the population sizes are strangely high.** If the likelihood surfaces are very flat than migrate might err onto regions that deliver to high population sizes. if this happens in a short chain than the program will rarely be able to return to more reasonable values. You need replication and heating (see question about inconsistent estimates above). I am biasing starting in version 1.5 towards the driving parameters (the parameters you use to run a chain), so that it will be harder for the program to climb to ureasonable high values, but it will go there if your data suggests such values. Although I do not believe that $\Theta > 10$ are reasonable [remember our Θ is **site** and not by locus for sequence data.], your data might violate assumptions of migrate (and also of FST) that make it hard to get correct estimates.

8. **I run MIGRATE using Θ and M parameters but I want to calculate $2Nm$ (my data is a haploid lichen)?**

   To calculate 2Nm for the haploid lichen-forming fungus for population 1, I have to multiply the

theta of population 1 by the IMMIGRATION rate into population 1. Is that correct?

In other words, if a M matrix is set up as in the Migrate manual,

```
-          M2->1         M3->1              -     10     100
M1->2         -          M3->2    =     20     -      30
M1->3         M2->3         -              90     3      -
```

and the thetas are
theta1=0.01,
theta2=0.001,
theta3=0.0001

This is what I should get for the 2Nm:
$2Nm[2->1] = theta1 * M2->1 = 0.01 * 10$
$2Nm[3->1] = theta1 * M3->1 = 0.01 * 100$

So the final 2Nm matrix would look like
- 0.1 1
0.02 - 0.03
0.009 0.0003 -

Bibliography

Abdo, Z., Crandall, K. A., and *Joyce, P.*, 2004 Evaluating the performance of likelihood methods for detecting population structure and migration. Molecular Ecology **13:** 837–851.

Bahlo, M. and *Griffiths, R. C.*, 2000 Inference from gene trees in a subdivided population. Theoretical Population Biology **57:** 79–95.

Beerli, P., 2004 Effect of unsampled populations on the estimation of population sizes and migration rates between sampled populations. Molecular Ecology **13:** 827–836.

Beerli, P., 2006 Comparison of Bayesian and maximum likelihood inference of population genetic parameters. Bioinformatics **22:** 341–345.

Beerli, P., 2007 Estimation of the population scaled mutation rate from microsatellite data. Genetics **177:** 1967–1968.

Beerli, P., 2009 How to use MIGRATE or why are Markov chain Monte Carlo programs difficult to use? In *Population Genetics for Animal Conservation*, edited by G. Bertorelle, M. W. Bruford, H. C. Hauffe, A. Rizzoli, and C. Vernesi, volume 17 of *Conservation Biology*, pp. 42–79, Cambridge University Press, Cambridge UK.

Beerli, P., Ashki, H., Mashayekhi, S., and *Palczewski, M.*, 2022 Population divergence time estimation using individual lineage label switching. G3 Genes|Genomes|Genetics **12**.

Beerli, P. and *Felsenstein, J.*, 1999 Maximum-likelihood estimation of migration rates and effective population numbers in two populations using a coalescent approach. Genetics **152:** 763–73.

Beerli, P. and *Felsenstein, J.*, 2001 Maximum likelihood estimation of a migration matrix and effective population sizes in n subpopulations by using a coalescent approach. Proceedings of the National Academy of Sciences of the United States of America **98:** 4563–4568.

Carling, M. D. and *Brumfield, R. T.*, 2007 Gene sampling strategies for multi-locus population estimates of genetic diversity (θ). PLoS One **2:** 160.

Chib, S. and *Greenberg, E.*, 1995 Understanding the Metropolis-Hastings algorithm. American Statistician **49:** 327–335.

Di, Rienzo A, Peterson, AC, Garza, JC, Valdes, AM, Slatkin, M, and *Freimer, N.*, 1994 Mutational processes of simple-sequence repeat loci in human populations. Proceedings of the National Academy of Sciences of the United States of America **Blank:1991:CMG:** 3166–70.

Dieringer, D. and *Schlotterer, C.*, 2003 microsatellite analyser (msa): a platform independent analysis tool for large microsatellite data sets. Molecular Ecology Notes **3:** 167–169.

Drummond, A. and *Rambaut, A.*, 2007 Beast: Bayesian evolutionary analysis by sampling trees. BMC Evolutionary Biology **7:** 214.

Drummond, A. J., Rambaut, A., Shapiro, B., and *Pybus, O. G.*, 2005 Bayesian coalescent inference of past population dynamics from molecular sequences. Mol. Biol. Evol. **22:** 1185–92.

Felsenstein, J., 1993 PHYLIP: phylogenetic inference package version 3.5c. Distributed over the Internet: http://evolution.genetics.washington.edu/phylip.html .

Felsenstein, J., 2005 Phylip (phylogeny inference package) version 3.6. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle. .

Felsenstein, J. and *Churchill, G. A.*, 1996 A hidden markov model approach to variation among sites in rate of evolution. Molecular Biology and Evolution **13:** 93–104.

Geyer, C. J., 1991 Estimating normalizing constants and reweighting mixtures in Markov chain Monte Carlo. Technical Report 568, School of Statistics, University of Minnesota.

Geyer, C. J. and *Thompson, E. A.*, 1995 Annealing Markov-chain Monte-Carlo with applications to ancestral inference. Journal of the American Statistical Association **90:** 909–920.

Hammersley, J. M. and *Handscomb, D. C.*, 1964 *Monte Carlo Methods*. Methuen and Co., London.

Harding, R. M., Fullerton, S. M., Griffiths, R. C., Bond, J., Cox, M. J., Schneider, J. A., Moulin, D. S., and *Clegg, J. B.*, 1997 Archaic african and asian lineages in the genetic ancestry of modern humans. American Journal of Human Genetics **60:** 772–789.

Hastings, W. K., 1970 Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57:** 97–109.

Hey, J., 2010 Isolation with migration models for more than two populations. Molecular Biology and Evolution **27:** 905–20.

Hudson, R. R., 1991 Gene genealogies and the coalescent process. Oxford Surveys in Evolutionary Biology **7:** 1–44.

Hudson, R. R., 2002 Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics **18:** 337–338.

Kass, R. E., Carlin, B. P., Gelman, A., and *Neal, R. M.*, 1998 Markov chain Monte Carlo in practice: A roundtable discussion. American Statistician **52:** 93–100.

Kimura, M. and *Ohta, T.*, 1978 Stepwise mutation model and distribution of allelic frequencies in a finite population. Proceedings of the National Academy of Sciences of the USA **75:** 2868–2872.

Kingman, J., 1982a The coalescent. Stochastic Processes and Their Applications **13:** 235–248.

Kingman, J., 1982b On the genealogy of large populations. In *Essays in Statistical Science*, edited by J. Gani and E. Hannan, pp. 27–43, Applied Probability Trust, London.

Kingman, J. F., 2000a Origins of the coalescent. 1974-1982. Genetics **156:** 1461–1463.

Kingman, J. F. C., 2000b Origins of the Coalescent: 1974-1982. Genetics **156:** 1461–1463.

Kuhner, M. K., Beerli, P., Yamato, J., and *Felsenstein, J.*, 2000 Usefulness of single nucleotide polymorphism data for estimating population parameters. Genetics **156:** 439–447.

Kuhner, M. K., Yamato, J., and *Felsenstein, J.*, 1995a Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. Genetics **140:** 1421–1430.

Kuhner, M. K., Yamato, J., and *Felsenstein, J.*, 1995b Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. Genetics **140:** 1421–30.

Kuhner, M. K., Yamato, J., and *Felsenstein, J.*, 1998 Maximum likelihood estimation of population growth rates based on the coalescent. Genetics **149:** 429–434.

Mashayekhi, S. and *Beerli, P.*, 2019 Fractional coalescent. Proceedings of the National Academy of Sciences **116:** 6244–6249.

Maynard Smith, J., 1970 Population size, polymorphism, and the rate of non-darwinian evolution. American Naturalist **104:** 231–237.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and *Teller, E.*, 1953 Equation of state calculations by fast computing machines. Journal of Chemical Physics **21:** 1087–1092.

Nath, H. B. and *Griffiths, R. C.*, 1993 The coalescent in two colonies with symmetric migration. Journal of Mathematical Biology **31:** 841–851.

Neal, R., 2003 Slice sampling. The Annals of Statistics **31:** 705–767.

Nei, M. and *Feldman, M. W.*, 1972 Identity of genes by descent within and between populations under mutation and migration pressures. Theoretical Population Biology **3:** 460–465.

Nielsen, R., 1998 Maximum likelihood estimation of population divergence times and population phylogenies under the infinite sites model. Journal of Theoretical Population Biology **53:** 143–151.

Notohara, M., 1990 The coalescent and the genealogical process in geographically structured population. Journal of Mathematical Biology **29:** 59–75.

Page, R. D., 1996 Treeview: an application to display phylogenetic trees on personal computers. Comput Appl Biosci **12:** 357–8.

Pluzhnikov, A. and *Donnelly, P.*, 1996 Optimal sequencing strategies for surveying molecular genetic diversity. Genetics **144:** 1247–1262.

Rambaut, A., 2006 Figtree. http://tree.bio.ed.ac.uk/software/figtree/.

Rambaut, A., 2007 Tracer v1.4. http://tree.bio.ed.ac.uk/software/tracer/.

RoyChoudhury, A. and *Stephens, M.*, 2007 Fast and accurate estimation of the population-scaled mutation rate, theta, from microsatellite genotype data. Genetics **176:** 1363–1366.

Shapiro, B., Drummond, A. J., Rambaut, A., Wilson, M. C., Matheus, P. E., Sher, A. V., Pybus, O. G., Gilbert, M. T. P., Barnes, I., Binladen, J., Willerslev, E., Hansen, A. J., Baryshnikov, G. F., Burns, J. A., Davydov, S., Driver, J. C., Froese, D. G., Harington, C. R., Keddie, G., Kosintsev, P., Kunz, M. L., Martin, L. D., Stephenson, R. O., Storer, J., Tedford, R., Zimov, S., and *Cooper, A.*, 2004 Rise and fall of the Beringian steppe bison. Science **306:** 1561–5.

Strimmer, K. and *Pybus, O. G.*, 2001 Exploring the demographic history of DNA sequences using the generalized skyline plot. Molecular Biology and Evolution **18:** 2298–305.

Swofford, D., 2003 PAUP*. phylogenetic analysis using parsimony (*and other methods). version 4.

Swofford, D., Olsen, G., Waddell, P., and *Hillis, D.*, 1996 Phylogenetic inference. In *Molecular Systematics*, edited by D. Hillis, C. Moritz, and B. Mable, pp. 407–514, Sinauer Associates, Sunderland, Massachusetts.

17 History and persistent problems

HISTORY OF MIGRATE

[people] in brackets helped with resources or reported problems.

Version numbers: for example 5.0.4
5: major version [addition of forces, big rearrangements]
0: subversion [new options, or large bug fixes]
4: fix minor problems [fixing errors, wording etc]

2025

May 1 Migrate 5.0.7 Replaced libharu 1.x with libharu 2.x because the older library had some issue with memory when creating huge PDF files (such as for dataset with large number of loci. Rebuilding the compile procedure.

2024

March 30 Migrate 5.0.6 Fixed an issue with specifyfing specific prior for individual parameters when using the parallel version.

March 15 Migrate 5.0.5 Fixed an issue with specifyfing specific prior for individual parameters.

2022

May 9 Migrate 5.0.4 Fixed an issue with assignment, in preparation of a software note on that.

March 30 Migrate 5.0.3 Bug fix with printing out a summary statistic when using SNPS, if the SNPs showed more than two states, calculation of the expected heterozygosity wrote beyond allocated memory.

January 30 Migrate 5.0.2 Bug fixes with inheritance-scalar, malformatted options led to a an inheritance scalar of 0.0, leading to errors in the analysis. Now both option versions: inheritance-scalars:<NO|YES:{....}> and inheritance-scalars={...} will work.

2021

November 6 Migrate 5.0.1 clean up and separate the work in progress

	from the more established material (mittag-leffler, and haplotyping) are avaiable via Makefile flag -DNEWVERSION
March 20	Migrate 5.0 merging of the Mittag-Leffler material with growth and unreleased version 4.5. The official version now includes exponential growth and assignment.
	2020
Summer 2020	Migrate 4.5 clean up and verification of the earlier version never released.
	2019
June 1	Migrate 4.4.4, changes to reduce the number of warnings with clang -weverything.
April 1	Migrate 4.4.3, updated the README and was able to compile for windows, no change of the code base.
March 21	Migrate 4.4.3, fixed a problem in the calculation of the P(G|param) when events are impossible, a test triggered an abort of the run, now it silently sets the probability of such a case to zero and discards the trial.
January 30	Migrate 4.4.2, revisited the SNP handling, now if a 'snp' with only one allele is entered no postitive log likelihoods appear, this may lead to new issues when users use ambiguity codes because they are treated as uncertainty and not as an indication of a heterozygote, I suggest to code heterozygots as two haplotypes. I added also an average locus-length to calculate the number of invariants sites better; fixed the problem that mutationmodels that allow for different base frequencies always calculated the base frequencies from the data.
	2018
November 29	Migrate 4.4.1, fixed an error in the Makefile.
November 26	Migrate 4.4.0 same as 4.2.14 (but to disambiguite version number problems.
November 17	Migrate 4.2.14 (revised), the program made several changes during 2018 that werer then reverted to 4.2.14 version because the intermediates introduced potential errors, this version will be renamed to 4.4 so that we can have a clean slate and do not have issues with the distribution because currently 4.2.14 does exist in two slight modified versions maybe eventually I learn not to make a mess with git. [this version was used to produce the figures in the Syst Bio paper.
	2017
April 16	Migrate 4.2.14: System to make the examples in the draft paper; revert to normalorig set of function for the speciat normal distribution.

```
March 25      Migrate 4.2.13: Choose between three different divergence
              distribution: Exponential, Normal, and Weibull distribution
              the Exponential estimates the mean divergence time, the
              the Normal the mean and the stadard deviation, and the
              Weibull, the mean and the spread parameter k.
March 6       Migrate 4.2.12; revision of the prior framework, now it is
              easy to specify different priors for every variable, this
              also improved the results for immigration with divergence;
              more simulation runs are now needed. The divergence system
              with the Weibull distribution and Expoential looks promising,
              but the Normal distribution is still in bad shape.
-------------------------------------------------------------------------
                                    2016
-------------------------------------------------------------------------
November 26   Migrate 4.2.11 trying to harmonize my code between version
              4.2.9 on the cluster that seems to generate better results
              than my 4.2.10 code, there are still differences with this
              check-in (git tag 4.2.11), code differences are now minimal
              but differences in results are still too big!
November 6    Migrate 4.2.10 more testing, in particular the numerical
              properties of erf(x)/erf(y) that lead to -inf, inf, nan issues
              when calculating the p(G|param).
August 24     Migrate 4.2.9 first attempt to release cleaning, fixing
              menus, and checking divergence dates.
June 24       Migrate 4.2.8 small changes to accomodate more stringent
              compiler warnings.
April 1       Migrate 4.2.7 addition of groups for migration patterns,
              for example 4 populations 1-2-3-4 can be described so that
              one migration rate for flow from left to right and another
              for flow from right to left using this option set
              custom-migration={xa00 bxa0 0bxa 00bx}; testing and improving
              calculations of lineages in genealogy (detected ties in the
              the timelist).
January 20    Migrate 4.2.6 improving numerical stability with many loci
              there are still issues (if you receive very different averages
              and modes, rerun the program with more steps, if the problem
              persists let me know (with infile and parmfile)
-------------------------------------------------------------------------
                                    2015
-------------------------------------------------------------------------
October       Migrate 4.2.4-5 improving stability
August 10     Migrate 4.2.3 addition of hyper-prior for prior gamma distribution
              checking and revising code for divergence
June 18       Migrate 4.2.2a and Migrate 3.6.11 missing -1 leading to a crash
              in printing population names found (thanks cipres-team) and fixed.
June 13       Migrate 4.2.1a fixing a problem with assignment probabilities
              where individuals to assign are not present for all loci.
May 24        Migrate 3.6.10 memory overrun resolved when data file numpop#
              and relabel numpop# do not match.
May 2         Migrate 3.6.9 the threaded versions crashed with msat data
              and fragment length translation because of an early access
              to the threadsafe random number generator which did not have
```

```
              the locks allocated: fixed.
March 30      Release of Migrate 4.2.0a, addition of a recover option, that
              allows 'checkpointing' once the first run using bayes-allfile
              failed, using the recover option [see in saved parmfile] one
              can continue and potentially finish the run.
March 26      Migrate 3.6.8 Fixed stupid typo in last update that lead to
              not calculating the allele frequency tables.
March 23      Release of Migrate 4.1.4a, fixed an error in the divergence
              calculation; changed the snp allele reporting so that it can
              work with large number of linked snps per locus. Fixed the snp
              issue in 3.6.7, too.
February 22   Release of Migrate 4.1.3a, added an option to do terse PDF
              output for very large number of loci, changed the example
              directory (see README in there). During January I also updated
              the legacy/old version to 3.6.6.
-------------------------------------------------------------------------
                                    2014
-------------------------------------------------------------------------
December 2    Release of Migrate 4.x, current version 4.1.0a, all updates
              are now done in this version, although I will do bugfixes in 3.6.5
              for a while to accomodate users.
October 18    Migrate 3.6.5 fixes a problem with the mean over many loci.
              When the program was not collecting enough samples then
              occasionally the error could lead to incorrect histograms.
              If you run 3.6.4 upgrade to 3.6.5.
-------------------------------------------------------------------------
                                    2013
-------------------------------------------------------------------------
December 15   Migrate 3.6.4/4.0 reevaluated the combination of loci in the Bayesian inference
              and found a bug that affects marginal likelihoods that are evident with small
              differences between models, this seems to be an issue with comparison based on
              sequence data (e.g. the Bayes factor tutorial). This problem was introduced wit
              the correction of the marginal likelihood in 3.6.
Setpember 29  Migrate 3.6.3 Added the hapmap datatype to the datatype menu.
September 19  Migrate 3.6.2 Fix a problem with a potential infinite loop in the
              parameter proposal step that stopped some snp runs, fixed a reporting
              glitch with random-subset.
August 24     Migrate 3.6.1 Random-subset failed on windows, now fixed
August 20     Migrate 3.6 Problem with combining loci found and fixed, the
              correction for the overuse of the prior failed when the posterior
              is concentrated and sampling is sparse, leading to an overcorrection
              that in turn led to spurious peaks that became important with more
              and more loci.
June 10       Migrate 4.0a Assignment on a per allele basis works, Haplotyping
              also works.
June 6        Migrate 3.5.4 problem with UTF file type checking is causing
              problems on some machines, UTF checking is disabled for now,
              make sure that your data file does not contain any other
              characters than ASCII.
April 28      Migrate 3.5.3 incorrect printing of random subset list in pdf file
              is fixed.
April 15      Migrate 3.5.2 a bug in the parallel code the lead to a 1 Byte
```

```
              memory overrun is fixed.
March 11      Migrate 3.5.1 'fixed' the skyline plots, now the plots
              should look OK again. Some minor formatting issues in the options.
March 3       Migrate 3.5 added a Gamma distributed prior, added the
              prior distribution to the histogram in the PDF, to show how
              different the posterior is from time prior. Changes in the
              calculation of Nm (in contrast to M [m/mu]) that should give
              fewer problems than the earlier implementation. Running with
              'no data' now works for all cases.
January 26    Migrate 3.4.6 Fixed a reporting bug that did increase the
              printed accepted and tried samples when in a parallel run
              the number of nodes was smaller than replicates*loci.
January 4     Migrate 3.4.5 compilation on windows using cygwin works now.
January 2     Migrate 3.4.4 fixes a problem for maximum likelihood
              introduced in 3.3.1, the autotune feature introduced a memory
              overrun with likelihood, but not with Bayesian inference.
              Changed printout presenting relative mutation rate from data.
-------------------------------------------------------------------------
                                    2012
-------------------------------------------------------------------------
December 17   Migrate 3.4.3 fixes a crash when the burn-in is zero.
December 11   Migrate 3.4.2 Electrophoretic marker data was not read correctly
              due to a programming error (the program crashed), now fixed
              [thanks to Chester Sands for reporting]
December 6    Migrate 3.4.1 A report suggested that on some cluster computers
              the generation or random number seeds leads to a stall because
              the queried device does not deliver enough randomness and
              therefore blocks, since we do not need cryptographically usable
              random numbers the reader was changed to the non-blocking
              random device, this does not affect windows machine because
              those use the system clock (inferior).
November 24   Migrate 3.4 added a few commandline options (-version, -help)
              the -version command is used to test the fastmigrate-n
              installation for macintosh distribution
November 10   Migrate 3.3.5 when using Nm instead of M and also using a
              restricted custom migration matrix sometimes warnings were
              triggered, the problem was found and fixed (values for
              undefined parameters were recorded bu not used).
October 25    Migrate 3.3.4 Under some condition a problem with the combination
              of loci lead to empty posterior distribution diagrams and
              huge negative numbers for the marginal likelihood. My underflow
              protection failed on, well, for some runs. This fix still needs
              more review but looks good so far. I skip the version 3.3.3
              because I had an internal one that was released on some binary
              distributions. Some reporting during the run has changed.
August 26     Migrate 3.3.2 Fixed a memory problem that made it problematic to
              run datasets with many populations (and parameters).[Romain Mayor]
August 22     Migrate 3.3.1 Change of data distribution system for parallel
              version, this allows to faster start with many (>500) loci.
              Inclusion of autotuning for Metropolis-Hastings acceptance ratio
              this has promoted a change in the defaults for the proposal:
              now it is back to Metropolis-Hastings instead of Slice sampling.
```

```
Jul 22          Migrate 3.3.0 Bayes factor subsystem revised and improved.
                Version bump because of changes in menu ordering.
June 13         Migrate 3.2.21 does fix a bug that lead to problems when you
                analyze reduced custom migration matrices using parallel
                version and also did not save the bayesallfile [this should
                fix the infamous empty posterior histogram problem]]
May 20          Migrate 3.2.20 revision of the inheritance scalar system.
                the earlier versions did not work well.
May 10          Migrate 3.2.19 allowes now #M and #@M in the data file to
                estimate microsatellite repeat lengths from fragment lengts.
-------------------------------------------------------------------------
                                    2011
-------------------------------------------------------------------------
October 8       Migrate 3.2.17 fixed a problem when all parameters are set
                to average (m) and no intermediate file for Bayesian inference
                is used. Fixed a problem for windows that caused a crash when
                intermediate files for Bayes inference (bayesallfile) was
                used.
July 1          Migrate 3.2.16 removed some printing in the MPI version and
                fix of a condition that lead to print ? instead of numbers
                for the allele frequency tables (msats and iam datasets)
June 10         Migrate 3.2.15 found a mistake that was incorporated on
                May 30 that resulted in incorrect sampling of genealogies
                when the option fast-likelihood=NO and DNA data was used.
                The output then showed near zero acceptances for genealogies
                and extremely low ESS for the genealogies. This is fixed.
June 8          Migrate 3.2.14 reissue of the last update because I pushed
                older version onto the website, to avoid confusion the correct
                version is labeled 3.2.14.
May 30          Migrate 3.2.13 Fixed two memory problem that lead to
                occasional crashes in the likelihood inference module.
                Addition of testsuite that allows testing and comparing
                speed of different compilations and settings.
May 26          Migrate 3.2.12 Fixed warnings for Ubuntu Linux.
May 19          Migrate 3.2.11 I have started to use a testsuite to check
                more regularly on small glitches: the testsuite directory
                contains datafiles, a python script to run automated tests,
                the parameter warning system had a glitch with ML and
                parallel runs, now fixed.
May 14          Migrate 3.2.10 fixing and underflow problem with snp data
                when the number of individuals are large, this particularly
                affected model choice runs because the hottest chain often
                failed because snps did not use scaling of the conditional
                likelihood values on the genealogy; the fix introduces such
                a scaler with the same option that is used with the sequence
                data.
May 1           Migrate 3.2.9 Streamlining code
March 21        Migrate 3.2.8 Cleaning of little nuisances detected by the
                static analyzer in xcode, investigation of strange posterior
                with parallel runs and custom migration matrices and replicates.
                Revised the "Warning" section, to "Suggestions".
January 12      Migrate 3.2.7 Parallel Migrate had difficulties with Maximum
```

```
              likelihood printing of profiles tables (it crashed),
              temporary fix (the exchange of data among computer nodes
              needs to be more streamlined (time frame depends on funding)).
-------------------------------------------------------------------------
                                    2010
-------------------------------------------------------------------------
December 5    Migrate 3.2.6 Parallel Migrate broke for Bayesian inference
              when the reporting file bayesallfile was set to NO: fixed.
November 27   Migrate 3.2.5 a memory overflow in reporting large migration
              matrices in the ML-mode is fixed [Daniel T. R.]
November 21   Migrate 3.2.4 cleaning up of printing and using inheritance
              scalars. First trial distribution of a general fastmigrate
              python script.
November 5    Migrate 3.2.3 the PDF library was not able recognize the shipped
              zlib.h, hopefully fixed now.
October 31    Migrate 3.2.2 clean up problems with zlib in the configure file.
              No change in functions in the program.
October 25    Migrate 3.2.1 addition of a compression scheme for the recording
              of all parameter values in the Bayesian inference (simple add
              a .gz to the filename of the bayesallfile filename and it will be
              compressed, the source distribution now distributes also the zlib
              library and configure can be forced to use that, default is the
              system, but that may be slow. Little tests with very large datasets
              have been done for the compression scheme. Printing of trees
              now also works for microsatellite trees. Currently the printing of
              ALL trees using the parallele migrate does not work (use the single
              cpu version).
-------------------------------------------------------------------------
                                    2010
-------------------------------------------------------------------------
October 5     Migrate 3.2 addition of new microsatellite reading method
              with an additional inputline in the infile, Migrate can now
              use fragment-length as input. Even decimal values (as delivered
              by the .... machine as raw data can be used when the additional
              input lines specifies the repeat lengths, ambiguous alleles will be
              assigned to repeats using a simple probability model that assigns to
              to the lower repeat number with a triangular probability distribution,
              crossing 0.5 when the actual fragment-length is exactly in the middle
              of two repeat alleles. Addition of a system to report problematic
              configuration or runtime issues, this is work in progress and may need
              some user feedback. Currently migrate reports problems with upper bounds
              of prior specifications and too low Effective sample sizes.
September 15  Migrate 3.1.10 fix of a glitch with replicates and marginal
              likelihoods and replication (replication was not divided out)
              [thanks to Anders].
September 8   Migrate 3.1.9 reestablish the reading of old bayesallfile to
              recreate the output of an older run, I am still working on the
              tool to combine several old bayesallfiles so that users can run
              loci independently on different machines and then later get a
              combined estimate [thanks to Chris Drummond for helping to squash
              several of the key problems]
August 9      Migrate 3.1.8 memory fixes so that very large datasets
```

```
                (>100 populations) have a chance to run [whether these converge
                I do not know].
July ~15        Migrate 3.1.7 Cosmetic changes so that my Bayes factor tutorial
                and the migrate menu match.
May 24          Migrate 3.1.6 turned heating back on that got turned off to
                find a problem in 3.1.4. [Yuma More]
May 14          Migrate 3.1.5 Fixed problem in combining multiple loci when the
                Bayesian inference was using exponential prior distributions.
April 30        Migrate 3.1.4 problem with not allowing very very low migration
                rates anymore fixed [my fix on March 21 was obviously to
                brush.
March 21        Migrate 3.1.3 fixed two issues that affected microsatellite
                data (migration rates had tendency to be close to zero,
                population sizes overcompensated for that, this problem was
                introduced with 3.0.8/3.1).
February 6      Migrate 3.1.2 after way too many ours of porting the new
                migrate version to windows (relearning what I thought was
                standard on decent platforms). Many little changes to
                accomodate windows.
January 15      Migrate 3.1.1 fix a bug that prevented to show results for
                the first migration rate.
January 1       Migrate 4.0alpha allows for haplotypic data using mixed
                data types (sequence, msat, snps, ... ) all with different
                mutation models. Free combination of completely linked and
                and unlinked segments. Migrate allows the use of
                libhmsbeagle to calculate likelihoods on trees using the
                GPU and other speedups [not all options are allowed yet
                because beagle is not supporting all options.
-------------------------------------------------------------------------
                                    2009
-------------------------------------------------------------------------
October 30      Migrate 3.1 stable version -- some minor errors fixed.
                Improvement and correction of the marginal likelihood
                calculations.
August 1        Migrate 3.0.8 problems with tree printing and dated samples
                reported (thanks to Trevor Bedford) hopefully fixed.
May 24          Migrate 3.0.7 Population relabeling possible, but only
                using the parmfile,
                several small problems with relabeling may still persist.
May 5           Migrate 3.0.6 Memory issue with MPI printing resolved,
                Slice sampler speed-up results in about 25% faster runs
                (for small runs).
January 8       Migrate 3.0.5 Dated samples with multiple loci should work now
-------------------------------------------------------------------------
                                    2008
-------------------------------------------------------------------------
December 8      Migrate 3.0.4 reporting of ESS and autocorrelation in MPI parallel runs
                fixed, this problem does not appear in single or thread runs and does not affe
                accuracy in the MPI runs.
December 3      Migrate 3.0.3 Bayes factors tested and seems to work fine, first exploration o
                approximation of Thermodynamic integration using only 4 chains to approximate
                integral (using Bezier curve that approximates curve from 32 chains).
```

October 22	Migrate 3.0.2 Added a configure option (at compile time) so that one can force the PDF outputfiles to be letter size (default) or A4 size.
October 20	Migrate 3.0.1 Fixed bugs in the ML calculation on parallel computers with replication (thanks Jeff Row).
August 1	Migrate 3.0 updated manual, cleaned some small things, included skyline plots, and migration events, dated samples.
July 10	Migrate 2.5.2 Internal rearrangements for preparation of skyline plot calculations. Change Makefile so that all particular Apple platforms work.
June 10	Migrate 2.5.1
May 13	Migrate 2.5 Changes of the configure/make files so that the compile choices highlight only the most important compile targets. The compile on AIX systems should work now, although some tweaking is still needed.
April	Migrate 2.4.4 Fix of a random number seed problem in the windows distribution, some earlier version delivered always the same automatic random number seed.
February 19	Migrate 2.4.3 The Bayesian inference printed wrong values when the prior was excessively large compared to the posterior, if your posteriors cover only a small fraction of the prior range you need to rerun (an indicator of the problem is the large discrepancy of the mean and mode. This problem does not affect likelihoods.
January 22	Migrate 2.4.2 The parallel code was executing the likelihood ratio test multiple times, on some systems this caused a crash, fixed.
January 9	Migrate 2.4.1 A problem with reading options for genealogy summaries fixed. Missing file in source code added.
January 5	Migrate 2.4 Several improvements moved into the mainstream program: marginal likelihood calculations, histogram of migration events over time and probability of location of most recent common ancestor, standard Mac(Intel) distribution contains now a parallel cluster utility using all avaialable CPU cores.

2007

December 16	Migrate 2.3.4 Fix of a memory problem that made long runs with some data impossible. Memory management changed considerably for genealogies.
August 26	Migrate 2.3.3 Reordering of output (the MCMC run characteristics appear now all at the end.
August 14	Migrate 2.3.2 An error that affected runs using heating with the stepwise mutation model is fixed.
July 20	Migrate 2.3.1 Problem with SNP model solved (thanks to Ivo Chelo reporting the problem).
July 12	Change of default threshold value for stepwise microsatellite mutation model
June 26	Migrate 2.3 Improvement of the speed of the stepwise model by a huge factor (reorganizing code).
January 1	Migrate 2.2.0 Improvement of memory consumption of parallel runs, some cleaning up of wording in output, addition of slice

sampling proposal mechanism, revision of manual to reflect the program better. Fix of a problem in Brownian motion datatype introduced late November in 2.1.9 (only in prerelease version). Relative mutation rate estimation from data.

2006

October 21 Migrate-2.1.9 Several minor problems with heating and custom-migration matrix under Bayesian inference are fixed.

September 8 Migrate-2.1.9 Problem with switching between ML and Bayes analysis fixed, without this fix a simple switch between methods without adjusting priors resulted in priors that did never change (resulting in an obviously wrong result).

August 8 Migrate-2.1.8 Several smaller fixes and improvements fix for wrongly reporting acceptance of swaps between heated chains when using a Bayesian framework, several minor errors: when user mix infile and parmfile in the commandline option a warning is issued, when the data looks like an xml file the data import is aborted.[skyline plots are still experimental!!!!!].

June 8 Migrate 2.1.7 reducing memory footprint for bayesian analysis rearrangments in the migration history code. Changed histogram module now can plot histograms for events (was mig-histogram) and skyline plots similar to the ones by Strimmer and Pybus (2001)[rigorous tests are still not

April 27 Migrate 2.1.6 buf fix in the profile table writer for PDF, that crashed with some datasets and some machines (Macs, Others?)

April 11 Migrate 2.1.5 working on improvement of the PDF output file, MCMC-ML table and profile tables, and percentiles tables should work now.

March 6 Migrate 2.1.4 several cleanups and streamlining of the functions related to microsatellites to gain speed, addition of an option to override the menu option in the parmfile (migrate-n -nomenu parmfile or migrate-n -nomenu parmfile).

February 2 Migrate 2.1.3 Bug that resulted in crashes in windows binary found and fixed

January 14 Migrate 2.1.2 Memory problem with multiple replicates in ML method fixed, migration-histogram problem fixed, both probably introduced on 2.1.0

2005

December 24 Migrate 2.1.1 Use of valgrind to clean up some memory problems, memory footprint should be somewhat reduced

December 4 Migrate 2.1.0 Bayesian inference overhauled and rechecked, changes in interface in parmfile interface (use write in the menu to bring your parmfile up to date). Addition of a PDF printing interface, the Bayesian analysis is mostly complete in PDF, the maximum likelihood PDF interface is still lacking major parts. The ASCII output file is still the safest resource for ML. Several minor changes to menu and option printing.

July 20 Migrate 2.0.7 Sumfile naming option fixed, MPI version does not stall anymore when number of nodes and loci match. Profile tables code cleaned.

May 14 Miggui 0.8 release of Macos 10.3+ graphical user interface programmed by Carl McIntosh.

May 13 Migrate 2.0.6 Cleaning up of Bayesian menu options,

```
              Bayesian method with multilocus-microsatellite data
              still needs more tests, if you get failures please report them.
              Several minor memory leaks and one major leak in the Bayes code
              fixed.
January 24    Migrate 2.0.5 A string buffer read error in sgets() and
              a memory leak [Kelly Gallagher, Karl Schmid] in the tree
              changing algorithm fixed. Some compilation issues on
              windows machines fixed, should work now with MS Dev Studio.
-------------------------------------------------------------------------
                                    2004
-------------------------------------------------------------------------
December 27   Migrate 2.0.4 Fixed Parmfile-reader that missed the usertree option,
              menu was not affected by this problem.
November 29   Migrate 2.0.3 prior distribution and menu/options revised,
              Migrate documentation revised and Bayesian Inference added, plotting problem w
              multiple population solved.
October 14    Migrate 2.0.2  error in option reader (LRT, theta, and Migration rates)
              found and fixed
September 17  Migrate 2.0.1 parallel runs failed on some machine entering profile calculatio
              memory problem found and fixed.
July 27       Migrate 2.0 (alpha) Test release for the Molecular Evolution workshop
              at MBL in Woods Hole, MA. Introduction of Bayesian search strategy,
              introduction of distribution of replication scheme
              among multiple computers. The parallel version now distributes loci and
              replicates over a cluster using the standard message passing interface.
July 26       Migrate 1.8.2 Bayesian version nears completion, Bayesian-version
              runs using MPI in multiple replicates and at the end summarizes
              over the loci-histogram, program bayeshist put into the contributed
              folder. Bug with recording all trees while using heating fixed [Daniel Myers]
May 21        Migrate 1.8.1 addition of replication to the parallel scheme
              now it is possible to run replicates in parallel, if there are
              enough free compute-nodes.
May 1         Migrate 1.8 changes to the parmfile writing and reading parts
              a parmfile contains now the complete syntax of all available
              options in commentlines. Fix of a memory problem when the custom
              migration matrix and the number of populations are not in sync.

-------------------------------------------------------------------------
                                    2003
-------------------------------------------------------------------------
December 13   Migrate 1.7.7 under some conditions the parallel version crashed while
              reading the sequence data into the tree
October 23    Migrate 1.7.6 with very large numbers of loci the program crashed during readi
              the data file, for some datasets this fix did not work, and because I
              had this version up for a short while on the ftp site I increase the version
              number.
October 14    Migrate 1.7.5 with very large numbers of loci the program crashed during readi
              the data file.
August 04     Migrate 1.7.4 fix accidently deleted line for ADAPTIVE heating.
June  15      Migrate 1.7.3 MPI works now on IBM regatta (SP4) machines.
February 24   Migrate 1.7.2 working on problems with heating and gamma-deviated
              mutation rates.
```

```
February 2      Migrate 1.7.1 more checking, removal of a typo bug that
                shortens long chains (introduced in restructuring process in 1.7).
January 27      Migrate 1.7 added option so that one can choose whether the
                missing data gets discarded or not for the allelic data types
January 3       Migrate 1.7 Revised printout for likelihood ratio test [bug
                testing help Peter Pearman], inclusion of AIC (Akaike's
                information criterion). Inconsistency for custom migration
                matrices filled with '0' and 'm' fixed [Peter Pearman].
                updated list of papers that cite migrate. Migrate compiles
                in parallel using MPICH (I still prefer LAM (www.lam-mpi.org)).
                Addition of Makefile specification for IBM SP3.
                (711 registrations)
-------------------------------------------------------------------------
                                    2002
-------------------------------------------------------------------------
November 29     Migrate 1.6.9 If you run microsatellite data and used
                versions 1.6.7 or 1.6.8 you need to rerun because there was
                a data reading problem in that version
                and this is fixed now [I am sorry about this].
August 12       Migrate 1.6.8 problem with reading weighfiles and
                catfile solved [Jon Seger]. Change of printing
                routines in the parallel version, all workers
                send now to the master who is the print-center.
July 12         Migrate 1.6.7 fixed problem with large sumfiles [Alex Wang].
June 25         Migrate 1.6.6 fixed a reporting problem with adaptive heating.
                Second and hopefully final fix of of "?" for *all* allele data
                types [Eric Simandle]
June 16         Migrate 1.6.5 nasty bug in microsatellite and allele code fixed:
                when "?" were present then the second replicate could end up
                with the incorrect number of tips in the genealogy and the
                program would fail. [Alex Wang, Deirdre Joy], a related problem
                that occured when only 1 individual was scored for 1
                msat allele in a population is fixed, too [Russell Pfau].
                Memory overrun with brownian mutation model and heating fixed.
June 05         Migrate 1.6.4 Problem with SNP data reading fixed, currently
                running simulations to see if SNP works.
May 23          Migrate 1.6.3 Some more minor problems in the parallel
                implementation fixed.
May 20          Migrate 1.6.2 Problem with msat data distribution in the
                parallel version fixed [Eric Simandle]
May 13          Migrate 1.6.1 Fixed a fatal bug in the profiles when using
                profile=All:FAST [Martin Damus].
April 13        Migrate 1.6 In the infile the number of individuals per locus with
                sequence data  can now be different, to accomodate different numbers
                of individuals change the population line to
                <num ind locus1> <num ind Loc2> ..... <num ind loc m> Population name
                the old syntax will still work and still assume that in a population all loci
                have the same number of sequences.
April 12        Migrate 1.5.1 Fixed a problem when using sumfiles and replicates
                the profiles had difficulties to find the maximum.
March 28        Migrate 1.5 change of maximization routine, jumps far away
                from the driving values are penalized using a normal distribution
```

```
                with mean=param_0 and std=param_0, this should help that
                with some data sets the program will refrain to jump to
                ridiculously high values, although if your data suggest such values
                the program will go there, just more slowly. Addition of adaptive
                heating (MCMCMC) to help to search the solution space better.
February 18     Migrate-1.4 Problems with custom migration matrix fixed
                but profile tables with symmetric 4Nm have problems (@#$^&%@),
                symmetric M works.
-------------------------------------------------------------------------
                                    2001
-------------------------------------------------------------------------
December 18     Migrate-1.3.3 two buffer over-runs in the parallel part
                found and fixed, this affected Linux machines but not MacOSX
December 2      Migrate-1.3.2 Profile tables work again for settings
                profile=YES:FIXED and profile=YES:QUICK, I obviously broke
                these settings in version 1.2.4 [Martin Damus, Mats Bjorklund]
November 12     Migrate-1.3.1 In parallel version: the data file is only read
                by the master node and the distributed to the worker nodes.
October 28      Migrate-n 1.3 (minor bug fixes related to plotting and parallel
                MPI execution)  changes so that profiles are now calculated
                parallelized over parameters instead of loci. This will
                reduce network traffic and should finish much faster if there
                are as many cpu as there are parameters. Improvement in the
                the ML calculation [replaced the line search with a newer version]
August 15       Migrate-n 1.2.4 fix for geographic distance file option, some older
                versions [most likely version that were newer than 1.1] were using
                a similarity matrix instead of a distance matrix.
August 9        Migrate-n 1.2.3 On Compaq alpha the program crashed with
                underflows (EXP(-1000)=NaN and not 0 as everywhere else),
                now every EXP() will be safeguarded on Compaq alphas.
August 8        Migrate 1.2.2 Several minor fixes to "beautify" outfile and
                menu printing.
July 28         Migrate-n 1.2.1 changes to the menu to incorporate AIC for
                migration model selection, needs documentation.
July 15         Migrate-n 1.2 Reworking of microsat likelihood calculation.
                It seems that my change on April 15 was only doing half
                of the job, the conditional likelihood calculation
                were not taking into account unobserved alleles above and
                below the smallest and largest repeatnumber.
                Everybody using microsatellite data should upgrade.
April 30        Migrate-n 1.1 Reworking of likelihood calculation,
                should speed up parameter estimation about 10-20%.
                Inconsistency between manual likelihood-ratio description
                and program fixed, manual is now at version 1.1.
                If you want to use likelihood ratio test you should
                run 'long' runs or then use replication or heating.
April 20        Migrate-n 1.0.4 Harmonizing the use of end-of-line
                characters,  migrate had several problems
                reading files that were moved between different
                operating systems, hope this is all fixed now.
                Fix of FAST and QUICK profile options [Raphaelle Chaix].
April 18        Migrate-N 1.0.3. More fixes, plots should work now
```

```
                [Steven Irvin], windows binary now is compiled with
                correct set of C-files and not old ones [Mats Bjoerklund].
April 15        Migrate-N 1.0.2 Serious problem in microsatellite code
                found and fixed: the stepwise-mutation model was only
                approximated due to a programming error. The probability
                of the number if steps for a given length of time was only
                too crudely approximated. Without the error reports of
                Steven Irvin and Raphaelle Chaix I would not have found
                this. Problem with "?" as the last character in a file
                transported between different operating systems fixed
                [R. Chaix]. Error in new replicate-summarizing code fixed
                [S. Irvin].
April 11        Migrate-N 1.0.1 Glitch in single locus likelihood ratio test
                fixed (technically you should not use this anyway).
                Some small formatting issues when saving parmfile.
April 10        MIGRATE-N 1.0 several memory leaks fixed, plotting
                option changed and will plot only over all loci.
                Many internal changes/speedups that should not affect users,
                although may introduce no bugs. Gamma deviated mutation rate
                among loci changed considerably but still has difficulties
                to converge. Sumfile run with "allelic" data broke,
                fixed now.
February 12     MIGRATE-N 0.9.14 some cleanup on freeing memory in profile
                evaluation. Fixed a bug in the heating code when the dataset
                is "allelic data" [Robb Brumfield].
-------------------------------------------------------------------------
                                    2000
-------------------------------------------------------------------------
December 16     MIGRATE-N 0.9.13 Speed up: memcpy() in site rate category code
                is replaced by pointer swapping [C++ newsgroup contribution],
                more precalculation of parts of the acceptance-ratio
                calculation. Can use now a geographical distance file to
                specify a kind of an isolation by distance model.
                Rearrangement of likelihood ratio test menu [Steven Irvine],
                removal of the l-ratio=LOCUS option, l-ratio=MEAN is replace
                by l-ratio=MLE, this actually should not break old parmfiles.
December 05     MIGRATE-N 0.9.12 Bug fix: I broke the infinite allele code
                and it is fixed now again [Ron Goldthwaite]
November 26     MIGRATE-N 0.9.11 Bug fix: when compiled optimized on SUNs
                the program showed odd and wrong behavior, and did not
                accept any new genealogy after a while, perhaps it is
                a compiler problem, because it does not occur on Linux
                Intel and I have not found a memory problem. A more
                restrictive array copying seems to remedy the problem.
                Additions: Adapted for compiles in a MAC OS X terminal
                window, it runs about twice as fast as in MAC OS 9.
                Addition of an option for more accurate calculations
                during the data likelihood calculations with large data
                sets where the individual probabilities underflow and the
                program crashes, this option is not necessary for many
                data sets, but slows the down run around 20-40%.
October 18      MIGRATE-N 0.9.10 Bug fix: Multiple microsatellite loci analyzed
```

```
                with Brownian motion model, populations with no data, and a
                missing value in at least one of the sampled populations
                crashed, because the whole locus was discarded, it should
                work properly now [Martin Damus]
October 13      MIGRATE-N 0.9.9 Bug fix: Parallele evaluation on symmetric
                multiprocessor machines now works with rate categories (more
                than one process wrote to the same unguarded memory).
                Further redesign to easy transition to parallele processing
                of loci.
October 6       MIGRATE-N 0.9.8 Bug fix: The reanalysis of a sumfile
                with one locus failed in the profile likelihood calculation.
                Addition of saveguards for machines (Dec Alphas)
                that return for EXP(-1000) = NaN instead of a very small
                number or zero.
                Addition of a sequencing error
                possibility [see under Datatype with Sequence data].
                Heating scheme expanded from 4 chains to n<20 chains.
July 28         MIGRATE-N 0.9.7 Bug fix: in cases where a population size
                cannot be well estimated (the likelihood surface is flat)
                the reset function failed to calculate an average size,
                and returned 0.0 which resulted in erratic behavior
                [Patricia Brito].
July 22         MIGRATE-N 0.9.6 Addition of a logfile option,
                the Gamma-deviated mutation rate among loci seems to work
                but needs more rigorous testing, so sometimes it will still
                fail.
July 11         MIGRATE-N 0.9.5 Bug fixes: the addition of a null population
                should work now for all datatypes [Martin Damus],
                under some conditions the maximizer
                gave up too quickly, and (an embarassing one) for profile
                likelihood percentiles miscalculation of percentile values:
                some of the  old percentiles were wrong, To see what impact it
                had on your conclusions see below
                correct/:1%   5%   10% 25%   50% 75%   90% 95%   99%
                wrong/old:  0.5% 2.5%  5% 12.5% 50% 87.5% 95% 97.5% 99.5%
                The old tables were using the 1,5,10... labels but calculated
                values under "wrong/old".
                [the likelihood ratio tests are not affected by this]
                The new profile tables are set so that you can generate
                99%, 95%, 90%, 50% confidence intervals.
                [mutation=Gamma is still broken, sigh]
May 30          MIGRATE-N 0.9.4
                Fixed a bug in reading and writing summary files
                (options affected were write-summary and datatype=genealogy).
                mutation=Gamma is still broken [Eric Simandle],
                do not use it.
May 12          MIGRATE-N 0.9.3
                embarrased to say but the last fixed introduced a problem,
                in the likelihood calculation, hopefully  fixed now.
                mutation=Gamma is still broken [Eric Simandle],
                do not use it.
April 22        MIGRATE-N 0.9.2
```

```
                inconsistency in likelihood calculation with replication
                fixed.
April 21        MIGRATE-N 0.9.1
                Bug in Mac-version of automatic random number seed
                generation, and in recording start migration parameters fixed,
                and migration start parameter mix up in parmfile
                fixed [all Ken Wahrheit].
                Heating scheme changed, implemented a 4 parallel chain
                heating scheme (simulated tempering) based on Geyer and
                Thompson. The Tempered transition method (Neal) will
                be reimplemented in a later version.
                Fixes: ttratio now works for different values
                [Judite Alves],
                Registered users: 423
                                  (tried to find this time all doubles)
March 3         MIGRATE-N 0.9
                First introduction of estimation of parameters
                over multiple chains or multiple runs.
                Problems: Multiple chain/runs
                with the combination of gamma deviated mutation rate
                does not work yet. Heating scheme is broken.
-------------------------------------------------------------------------
                                    1999
-------------------------------------------------------------------------
December 10     MIGRATE-N 0.8.5
                Change of defaults: plot=FALSE, moved eventloop()
                in plot routine for Macintosh.
December 2      MIGRATE-N 0.8.4
                Revision of likelihood ratio test output. Change
                of "burn-in" default from 200 to 10000.
                Minor speedups in several functions.
November 23     MIGRATE-N 0.8.3
                Revision of heating scheme. But still needs more testing.
November 5      MIGRATE-N 0.8.2
                Addition of a convergence criterium: Gelman's R,
                (use progress=verbose)
                Added material to the
                likelihood ratio test documentation.
                Several minor bugfixes (sumfile related [Tonya Bitner],
                Profile Quantile table, verbose Progress reporting)
                Registered users: 372
September 7     MIGRATE-N 0.8.1
                More cleanup of C-code, incorporation of new spline
                routine ( but this is still experimental). Improvement
                of documentation.
August 20       MIGRATE-N 0.8
                A problem with the UPGMA starting tree fixed,
                with many individuals the starting tree contained
                some silly ordering, that produced uneven number of
                migration events on this tree and needs rather a long
                time to recover from this.
                profile likelihood speed improvements when there is a
```

```
               custom-migration matrix with zeroes.
               Registered users: 322
June 4         MIGRATE-N 0.7.1
               Division by 0 bug fixed in fst-calculation, this seems
               to bother only DEC Alphas.
June 1         MIGRATE-N 0.7
               Updated documentation, several minor things, warnings and error
               reporting should be more consistent, I am adding a section to
               the manual that describes all error/warning messages [partly
               done], the plotting graphics are more flexible now, but still
               need more work. You can specify the range and type of
               axes (log-scale, std-scale), and if the migration parameter
               shall be plotted as M=m/mu or 4Nm. Fix of inconsistency
               in migration value menu input [Reinaldo Brito].
               Fix of an error in the
               profile-method=FAST (it will need now more time to finish,
               because it is doing the final maximization over all other
               parameters), if you want its old behavior, that assumes that
               Theta and M are not correlated [not a too bad assumption],
               then use profile=YES:QUICK.
March 8        MIGRATE-N 0.6.3
               Updated documentation (fixed errors in description of
               random-seed options, added important material
               to profile-likelihood) ,
               inclusion of improved man page,
               fixed configure for SGI without gcc.
Feb 14         MIGRATE-N 0.6.2
               Tree traversal debug code removed,
               this killed runs with many individuals [Lisle Gibbs]
               Configure for SGI changed, does it work?
               MIGRATE-0.4: no change.
               Registered users: 280.
-------------------------------------------------------------------------
                                    1998
-------------------------------------------------------------------------
December 29    MIGRATE-N 0.6.1 Multilocus estimates are all wrong
               in version 0.6, silly programming mistake found an fixed.
               If you have used microsatellites or electrophoretic markers
               or several sequence loci you need to rerun that analysis.
               Result table should print now nicer with population numbers
               above 3.
               MIGRATE-0.4: no change.
               Registered users: 234
Oct 29         MIGRATE-N 0.6
               Addition of datatype=n that is for single nucleotide
               polymorphism data, no simulation with this kind of data
               is yet done, so I do not know about biases etc.
               Profile tables now report 4Nm instead of m/mu for
               the migration parameters.
               Documentation changed and contains now more about
               how to read the outfile and what you can and
               cannot do with the reported log(likelihood) values
```

```
              [Mats Bjorklund].
              Binaries for OPENSTEP available [thanks to Magnus Nordborg
              giving me an account on his machine].
              Registered users: 206
Sep 1         MIGRATE-N 0.4/0.5 [was not released, was too busy with
              other things]
              FST start values work now also for microsatellite data
              but I still need to check  the correctedness of the FST table
              when the data are microsatellites.
              Fixed wrong emmigration plots. Fixed wrong start
              calculations for allelic data when a delimiter was used,
              and several minor bug fixes. Profile-method
              "uncorrelated" from version alpha.1 recovered.
              Registered users: 197
June 14       MIGRATE-N alpha.3 and MIGRATE-0.4.2
              Several minor changes in migrate-n: menu addition for
              -profile method:
              profile-method=<Spline | Percentiles | Discrete>
              Spline: uses 1-dimensional splines to find percentiles,
              faster than the "Percentiles" option but not so accurate,
              "Discrete" evaluates at "fixed" (0.02, 0.1, 0.2, 0.5, 1, 2,
              5, 10, 50) * MLE of parameter.
              -with progress=yes you can see now a rough prognosed time
              of end of sampling genealogies and if you use profiles
              an estimated time of finishing.
              -Fix of reading in intermediate results (sumfile).
              -Most importantly a (hopefully) stable compile for
              Windows, I failed to find the cause why the program
              compiled with WATCOM failed to finish with "bigger" data sets,
              it is now compiled with mingw32/gcc-win32, this is
              a windows port of the same system I am using on my workstation.
              Please report failures, I can only try a limited set of
              examples.
              Migrate-0.4.2: new windows binary (using mingw32/gcc-win32)
              Registered users: 163
May 30        MIGRATE-N alpha.2 and MIGRATE-0.4.1
              With more than 2 sequence loci, there was a problem
              with the T/T-ratio, when the ratio was not specified
              for each locus.
              Start parameter problems with microsatellite data fixed
              [Mats Bjorklund].
              Persistent problems with Windows executable
              sometimes I get floating point errors, on all other systems
              this does not occur.
              Registered users: 153
May 29        MIGRATE-N alpha.1 and MIGRATE-0.4
              Memory bug in FST calculation found and fixed
              [Daniel Yeh]
              No change of Migrate-0.4
              Registered users: 148.
May 26        MIGRATE-N and MIGRATE-0.4
              This release has the two population version (Migrate-0.4)
```

```
                and an alpha-version of Migrate-n that can solve migration
                matrix population model with unequal population sizes and
                unequal migration rates for n populations, I tried up to 10
                and the results where fine, but I am pretty sure that if
                you try to feed in all your date of 100 subpopulation it
                will (a) probably crash, but more importantly (b) will
                need TERRIBLY long to run.
                I would like to get some
                feedback about what you want to see in the outfile,
                menu etc. Registered Users: ~138.
February 25     MIGRATE 0.4 (was not put up onto the website, I was to busy)
                More complex sequence evolution models (categories,
                weights, autocorrelation etc.) should work now,
                it was broken. Cleanup of some output file lines, and
                some menu entries. The FST estimation (Remember FST is only
                used to generate start parameter values) is
                in pre 0.4 versions logically flawed. It estimates
                2 parameters per population using F_within and F_between,
                but there is only 1 F_between. Correctly, we can
                only estimate maximally 3 parameters with 1 locus for
                two populations. I added an option into the MENU and into
                the PARMFILE (fst-type=<Theta | Migration >) with which you
                can decide which parameter is considered the same for both
                populations.
                Registered users:89
-------------------------------------------------------------------------
                                      1997
-------------------------------------------------------------------------
August 20       MIGRATE 0.3.1
                Confusing menu entries for start theta and 4Nm values
                fixed [Carol Reeb], the start migration values are now
                4Nm and *not* m/mu values as before. Automatic Random
                number seed on Macs and perhaps on other Systems
                delivered sometimes negative values, now fixed
                [Carol Reeb], although I would recommend to use your
                own random number seeds: best values are 4n + 1 in the
                range of 5 .. 2147483647, so there are plenty of
                start random number seeds. Menu entry for
                usertree options should be now more clear, the usertree
                options needs a genealogy with migration events on
                it [Tony Metcalf]. Currently MIGRATE can construct
                those, or you have to do it by hand, if you need to do
                this send me email, because the doc is not updated.
                Registered users:52
June 20         MIGRATE 0.3.0.
                Brownian motion approximation to stepwise mutation
                model for microsatellites added.  Solved problems:
                Input problems with microsatellites data, major memory
                allocation problem for datasets with more than 100
                gene copies fixed [Carol Reeb].  Update of some
                citation  and FST output tables [Byron Adams].
                Persistent problems: Long sequences AND high number of
```

```
                individuals need nuch longer chains than the proposed
                default. Try ten times longer "long" chains. Or use
                the option "moving-steps".
                Registered users:38
May 12          MIGRATE 0.2.1a.
                Fixed problems: Interleaved sequence data
                should work now, last character of individual names is
                now printing, and printing of second population data
                should work, too, although the EP data printout is
                still ugly.  [Allen Rodrigo]. Memory problem with some
                Allelic data fixed.
                Registered users: 30
April 30        MIGRATE 0.2a released.
                Fixed problems or changes:  Corrections of several
                minor problems, Printing of the data fixed, but still
                ugly; Memory problem with large sequences fixed.
                Options: treefile added, can write now a genealogy
                with migrations; the option progress=Verbose for more
                information during a run, the progress=Yes gives now
                less information than before. Output: covariance
                matrix for combined loci now prints, too.  Persistent
                problems:  -Long sequences need very long chains to
                remove the starting conditions for the migration rate
                from the first tree (see documentation).
                -Microsatellites still have probably a bias downwards
                in Theta, but I need more simulations to make this
                more clear.
                Registered users: 8
March 4         First trial release of MIGRATE 0.1a
                This release is not announced widely, because I have
                to test, almost everything including all HTMLs,
                registration, and the program itself: simulations need
                time.  Registered users: 1
```