

Migrate

Manual

***Peter Beerli
Department of Genetics
Box 357360
University of Washington
Seattle, WA 98195-7360***

beerli@genetics.washington.edu

*Version 0.9
Last update: November 26, 2000
Started: January 1, 1997*

For the impatient

Reading manuals is not everybody favored task, 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.**

Good luck,
Peter Beerli Seattle, July 2000

Contents

1	Theoretical considerations	2
1.1	Maximum likelihood estimation of migration rates	2
1.2	Performance criteria	3
1.3	Data models	5
1.3.1	Infinite allele model	5
1.3.2	Microsatellite model	5
1.3.3	Sequence model	6
1.3.4	Single nucleotide polymorphism data (SNP)	7
2	Program usage	8
2.1	Data file specifications	8
2.1.1	Examples of the different data types	10
2.1.2	Microsatellite data	10
2.1.3	Sequence data	11
2.2	Additional files	12
2.2.1	Overview	12
2.2.2	Necessary input file	13
2.2.3	Optional input files	13
2.2.4	Output file	14
2.2.5	Optional output files	14
2.3	How to run	14
2.4	Menu and Options	14
2.4.1	Data type	15
2.4.2	Input/Output formats	18
2.4.3	Start values for the Parameters	22
2.4.4	F_{ST} calculation	23
2.4.5	Migration model	24
2.4.6	Search strategy	24
2.4.7	Obscure options	26
2.4.8	Parmfile specific commands	27
2.5	Likelihood ratio tests and profile likelihood	28
2.5.1	Likelihood ratio test	28
2.5.2	Profile likelihood	30

3	Accuracy of results	32
3.1	Monitoring progress	32
3.2	Run time and accuracy	33
3.3	Quick guide for achieving “good” results with <code>migrate</code>	33
3.4	How to avoid conflicts with other computer users	35
4	Presentation of results	36
4.0.1	Walk through an outfile	36
5	Frequently asked questions, errors and warnings, and troubleshooting	42
5.1	Questions	42
5.1.1	General	42
5.1.2	About the datafile	42
5.1.3	About options and how to run	42
5.1.4	About reading the outfile	42
5.2	Answers	43
5.2.1	General	43
5.2.2	Data file related	43
5.2.3	About option and how to run	44
5.2.4	About reading the outfile	45
5.3	Errors and warnings displayed by <code>migrate</code>	46
5.3.1	Errors	47
5.3.2	Warnings	48
5.4	Troubleshooting	49
6	Program distribution and installation	50
6.1	Program availability	50
6.2	Installation	50
6.2.1	Binaries	50
6.2.2	Source	50
7	Miscellaneous	52
7.1	Wish list	52
7.2	How to give credit	52
7.3	Copyright	52
7.4	Acknowledgement	53
8	Literature	54
9	Appendix	56
9.1	Mathematica plot package	56
9.2	History and persistent problems	57

Chapter 1

Theoretical considerations

1.1 Maximum likelihood estimation of migration rates

Migrate calculates maximum likelihood estimates for migration rates and effective population sizes of two populations using genetic data (Fig 1). The parameters to estimate are Θ_1 , Θ_2 , Nm_1 , Nm_2 , which are $4 \times$ effective population size \times mutation rate per site per generation and effective population size \times migration rate per generation in population 1 and 2, respectively. The estimation process uses an expansion of the coalescent theory (Kingman 1984a,b) which includes migration (Hudson 1990, Nath and Griffiths 1993, Notohara 1994). A likelihood estimate of the parameters \mathcal{P} using genealogies \mathcal{G} with data \mathcal{D} would be

$$L(\mathcal{P}) = \sum_{\mathcal{G}} \text{Prob}(\mathcal{D} | \mathcal{G}) \text{Prob}(\mathcal{G} | \mathcal{P}).$$

This is the sum over the joint probability of the data given a genealogy (this is the conventional likelihood in a phylogenetic tree) and the probability of the coalescent. Unfortunately, this sum has an infinite number of summands; we have to sum over all genealogies and all possible branch length. We can solve this problem by using a Markov chain Monte Carlo approach with importance sampling due to Metropolis (1954) and Hastings (1974). For an introduction see Hammersley and Handscomb (1964) or Chib and Greenberg (1995), and see Kuhner et al. (1995) for its application to the coalescence theory). We bias the search path through all trees towards trees with higher likelihoods (Fig. 2) and have then to correct for this. The likelihood formula changes to

$$\frac{L(\mathcal{P})}{L(\mathcal{P}_0)} = \frac{1}{m} \sum_i^m \frac{\text{Prob}(D | g_i) \text{Prob}(g_i | \mathcal{P})}{\text{Prob}(D | g_i) \text{Prob}(g_i | \mathcal{P}_0)}.$$

This is very reasonable, because summands with low probabilities will almost not contribute to the final likelihood. For more information on the base model, you should read Beerli and Felsenstein (1999) and Kuhner et al. (1995). The approximation of the likelihood using a ratio makes it difficult to compare different runs of the program, if the program reports a likelihood then this is actually a ratio of likelihoods and since we recalculate the parameters for each chain, the values for \mathcal{P}_0 are different between runs, and therefore it is impossible to compare them. An escape of this problem is to run the program using the full model (e.g. $n \times n$ parameters and use the likelihood ratio test for specific scenarios.

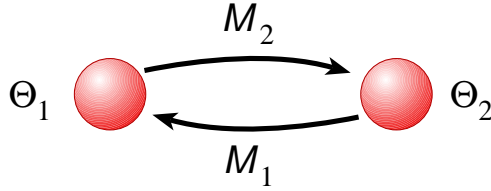


Figure 1.1: Populations exchanging migrants with rate m_i per generations and with size N_e . The parameters are scaled by mutation rate μ which is with sequence data per site per generation. The estimated parameters are therefore: Θ_i which is $4N_e^{(i)}\mu$ and \mathcal{M}_i which is m_i/μ , the migration estimate is more commonly expressed as $4Nm$ which is just $\Theta\mathcal{M}$.

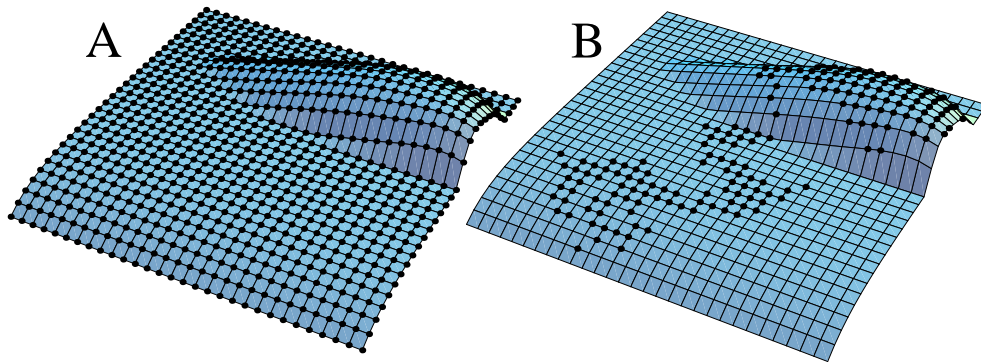


Figure 1.2: (A) On an imaginary, infinite likelihood surface we would need to sample every possible genealogy and sum all these values which is not possible, but trees with low probability will not contribute much to the final likelihood, (B) by biasing towards better trees we can sample effectively from those trees with high contribution to the final likelihood and can approximate the likelihood.

1.2 Performance criteria

There is a difficulty with these kind of samplers that we do not know how long we have to run the sampler to get “accurate” estimates. Despite the huge literature about measures when to stop sampling, there is still no good criteria available. Two kind of errors can produce problems: (1) programming errors, (2) the sampler was not run long enough. Several ways exist to investigate these two sources of problems, one can check if

- 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 $\text{Prob}(G|\mathcal{P}_0)\text{Prob}(D|G)$, the one we sample from. [checks **(1)**]
- simulation studies show that we can recover parameters and population structure that was used to create the data. [checks **(1,2)**]
- comparison with other programs produce similar results. I compared `migrate` with `genetree` (Bahlo and Griffiths 1999) and with `fluctuate` (Kuhner et al. 1998). The comparison with

`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 `migratedeliver` the same estimates, and approximative confidence intervals, although `genetree` is very slow compared to `migrate` for that specific data set (Beerli and Felsenstein, in prep.). 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 $\Theta = 0.0119$. [checks (1,2)]

- 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 $\text{Prob}(D|G)$, Figure 1.3). A plot of a sequence of $\text{Prob}(\mathcal{P}|G_i)\text{Prob}(D|G_i)$ 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 1.4), 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)]

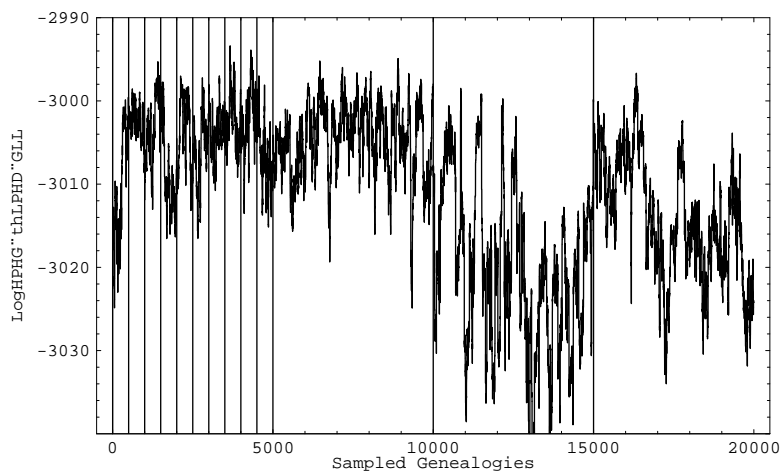


Figure 1.3: Data likelihood $\text{Prob}(D|G)$ for all sampled genealogies: A sample run of migration estimation using 2 populations, the very long vertical lines mark chain boundaries (10 short and 3 long chains). Totally, 10 short chains \times 500 sampled genealogies + 3 long chains \times sampled 5000 genealogies were sampled out of total 400,000. The values for not recorded trees are not shown.

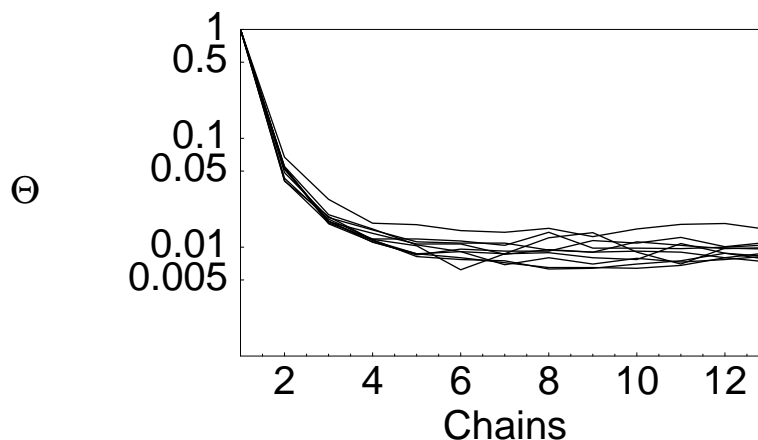


Figure 1.4: Convergence to the true parameter region. Ten runs were started from a $\Theta = 1.0$. The data was generated using a $\Theta = 0.01$. Totally, 10 short chains \times 500 sampled genealogies + 3 long chains \times sampled 5000 genealogies were sampled out of total 400,000.

1.3 Data models

1.3.1 Infinite allele model

This assumes that every mutation will result in a new allele, there is no back mutation (Fig. 1.5). 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 of each locus into account. These mean-parameters can be found by either assuming that the mutation rate has no variation (as all, at least those I know, other programs do) or uses a Γ distributed mutation rate with shape parameter α which is in this case

$$\frac{1}{(\text{coefficient of variation})^2}.$$

1.3.2 Microsatellite model

Ladder model

The ladder model was invented by Ohta and Kimura (1973, 1978) for electrophoretic markers, but was not as good as expected in describing real electrophoretic alleles. For microsatellites this model seems much more appropriate (e.g. Valdes et al. 1993, but see Di Rienzo 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. 1.6). Summarizing over loci can be done either by assuming the mutation rate is Gamma distributed or constant. This assumes, of course, independence between loci.

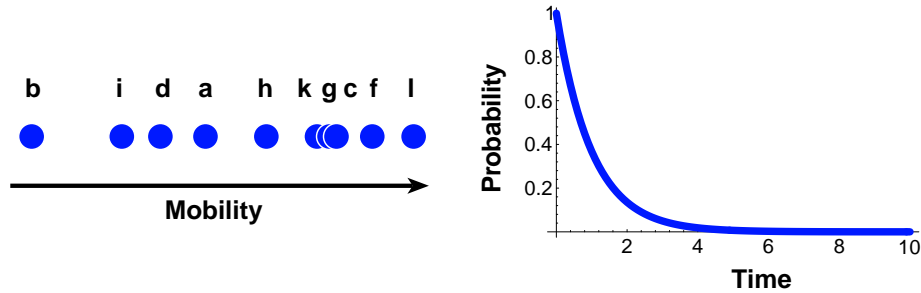


Figure 1.5: Left: Mobility of electrophoretic marker in an electric field. the alleles a,b,c,.. describe a possible sequence of mutation, their mobility is not correlated with the mutational history. Right: The probability that a given allele is not mutating during some time, this is a simple exponential relationship.

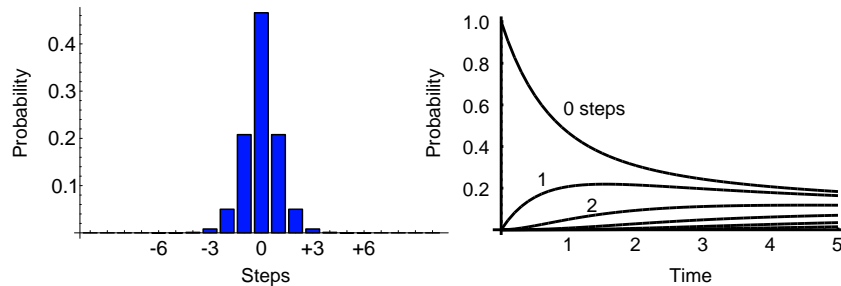


Figure 1.6: Left: Number of repeat changes of a microsatellite marker. The probability to have a slippage of only one repeat is higher than the slippage of more than one repeat, in a given time, here time=0.1. Right: The probability that a change of 0,1,2,.. steps is occurring during some time.

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 work still in progress (Felsenstein and Beerli, in prep.).

1.3.3 Sequence model

Migrate implements the sequence model of Felsenstein (1984) available in `dnaml` (PHYLIP 4.0, Felsenstein 1997)(Fig. 1.7). The transition probabilities were published by Kishino and Hasegawa (1989). *Migrate* does not allow for recombinations and therefore is only well suited for mitochondrial sequences or other non-recombining DNA stretches. Summarizing over "loci" assumes in addition that the loci are unlinked. The mutation rate among loci may be either constant or following a Gamma distribution.

Like `dnaml`, *Migrate* also allows for different evolutionary rates, mutation categories and autocorrelation, although any use of these additional features can slow down the program to a crawl, but this may change in the future as computers double their speed roughly every 2 years.

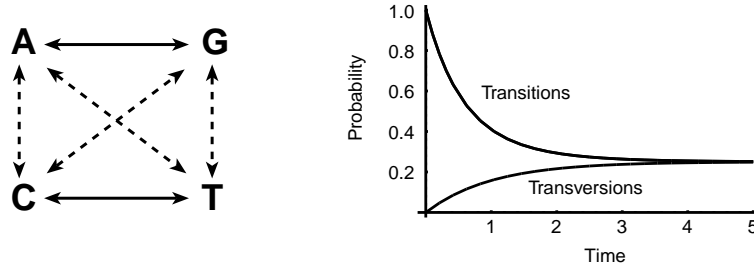


Figure 1.7: Left: Sequence mutation model. Transitions are shown in black lines, transversion are shown with dotted lines. Right: The probability that a transition or transversion is occurring during some time. The shown graph uses equal base frequencies, but the used model does not need this restriction.

1.3.4 Single nucleotide polymorphism data (SNP)

We use a rather restrictive ascertainment models for SNPs (Kuhner *et al.* 2000). Currently there are two versions implemented. If you want to use the SNP options, please contact me before you run large scale analyses.

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.
2. SNP were developed from a panel population of which we know the number of individuals, and that the markers developed were variable, but we do not know the actual nucleotides for the individuals [Not fully tested].

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.

If you want to assume that each SNP is unlinked then you need to code each SNP like a sequence data locus with one nucleotide (see the examples for sequences), I have run successfully 50 SNP loci on a laptop with 40 MB of RAM. But there may be better ways to run loci consisting of only one site.

Chapter 2

Program usage

If you want to know how to install or compile the program goto the sections at the end of this manual.

This manual is in a transition phase until the two-population program `migrate` and the n -population `migrate-n` are merged. The merger will happen once the n -population program is formally described in a peer-review journal. Options only available with the 2-population version are marked with **(2-POP)**.

2.1 Data file specifications

The data needs to be in a certain form; for us, the following format was most convenient. Eventually we will include the NEXUS format (which is used in MacClade and Paup), but currently the NEXUS format is not able to keep all the data format LAMARC allows.

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.

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 blanks, the token for the individual name `Ind1 02 @` is legal. The most common data file for enzyme electrophoretic data or microsatellite data would look like this (examples follow):

```
<Number of populations> <number of loci> {delimiter between alleles} [project title 0-79]
<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>
<Individuum 1 10-10> <data>
```

```
<Individuum 2 10-10> <data>
....
```

The delimiter is needed for microsatellite data and the project title is optional. The **data** will be described in the following sections. The individual name has to be by default 10 characters (same as in PHYLIP), but can be changed to an other constant in the parmfile, even to a length of 0. 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> <title for population 0-79>
<Individuum 1 10-10> <data locus 1>
<Individuum 2 10-10> <data locus 1>
....
<Individuum 1 10-10> <data locus 2>
<Individuum 2 10-10> <data locus 2>
....
<Number of individuals> <title for population 0-79>
<Individuum 1 10-10> <data locus 1>
<Individuum 2 10-10> <data locus 1>
....
<Individuum 1 10-10> <data locus 2>
<Individuum 2 10-10> <data locus 2>
....
```

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> <title for population 0-79>
<Individuum 1 10-10> <data locus 1 part 1>
<Individuum 2 10-10> <data locus 1 part 1>
....
<data ind1 locus 1 part 2>
<data ind2 locus 1 part 2>
....
<Individuum 1 10-10> <data locus 2>
<Individuum 2 10-10> <data locus 2>
....
<data ind1 locus 2 part 2>
<data ind2 locus 2 part 2>
....
etc.
```

The input for SNPs is the same as for sequence data.

2.1.1 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.

Enzyme electrophoretic 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.

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
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
PB16843 ee bb ab bb aa aa aa cc bb cc aa
PB16844 ee bb bb bb ab aa aa cc bb cc aa
```

2.1.2 Microsatellite data

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
-----
ACCTATAGCACACACACAAATGCGA
```

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

Example:

```
2 3 . Rana lessonae: Seeruecken versus Tal
2 Riedtli near G\"undelhart-H\"orhausen
0 42.45 37.31 18.18
0 42.45 37.33 18.16
4 Tal near Steckborn
1 43.46 33.37 18.18
1 44.46 33.35 19.18
1 44.46 35.? 18.18
1 43.42 35.31 20.18
```

2.1.3 Sequence 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 GENE BANK 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 (which is how they are treated). The characters constitute the IUPAC (IUB) nucleic acid code plus some slight extensions. They enable input of nucleic acid sequences taking full account of any ambiguities in the sequence.

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

Example with 2 population with **2 loci**, the sequences are NOT interleaved:

```

2 2 Make believe data set using simulated data (2 loci)
50 46
3  hinders wiesli
eis      ACACCCAACACGCCCGCGGACAGGGGCTCGAGGGATCACTGACTGGCAC
zwo      ACACAAAACACGCCCGCGGACAGGGGCTCGAGGGGTCACTGAGTGGCAC
drue     ATACCAGCACGCCCGCGGACAGGGGCTCGAGGGAGCACTGAGTGGAAC
eis      ACGCGGCGCGGAACGAAGACCAAATCTTCTTGATCCCCAAGTGTC
zwo      ACGCGGCGCGGAACGAAGACCAAATCTTCTTGATCCCCAAGTGTC
drue     ACGCGGCGCGGAACGAAGACCAAATCTTCTTGATCCCCAAGTGTC
2  vorders wiesli
vier     CAGCGCGGTATCGCCCATGTGGTTCGGCCAAAGAATGGTAGAGCGGAG
fuef     CAGCGCGAGTCTCGCCCATGGGGTTAGGCCAAATAATGTTAGAGCGGCA
vier     TCGACTAGATCTGCAGCACATACGAGGGTCATGCGTCCCAGATGTG
fuefLoc2 TCGACTAGATATGCAGCAAATACGAGGGGCATGCGTCCCAGATGTG

```

Same example with 2 population with **2 loci**, but the sequences are now interleaved:

```

2 2 Make believe data set using simulated data (2 loci, interleaved)
50 46
3   hinders wiesli
eis   ACACCCAACACGGCCCGCGGACA
zwo   ACACAAAACACGGCCCGCGGACA
drue  ATACCCAGCACGGCCCGCGGACA
      GGGGCTCGAGGGATCACTGACTGGCAC
      GGGGCTCGAGGGGTCACTGAGTGGCAC
      GGGGCTCGAGGGAGCACTGAGTGGAAC
eis   ACGCGGCGCGGAACGAAGACCA
zwo   ACGCGGCGCGAGAACGAAGACCA
drue  ACGCGGCGCGAGAACGAAGACCA
      AATCTTCTTGATCCCCAAGTGTC
      AATCTTCTTGATCCCCAAGTGTC
      AATCTTCTTGATCCCCAAGTGTC
2   vorders wiesli
vier  CAGCGCGCGTATCGCCCCATGTGGTTCGGCCAAAGAATG
fuef  CAGCGCGAGTCTCGCCCCATGGGGTTAGGCCAAATAATG
      GTAGAGCGGAG
      TTAGAGCGGCA
      TCGACTAGATCTG CAGCACATAC
      TCGACTAGATATG CAGCAAATAC
      GAGGGTCATGCGTCCCAGATGTG
      GAGGGGCATGCGTCCCAGATGTG

```

2.2 Additional files

2.2.1 Overview

I tried to make it simple and redundant, so that there are more than one way to set up things. There are several special file names, some of them can be changed others not:

Filename	Type	Description	Needed?
infile	Input	holds you data	necessary
parmfile	Input	holds options	optional
seedfile	Input	holds a random number seed	optional
catfile	Input	holds categories for mutation rate variation	optional
weightfile	Input	holds weights for each site	optional
outfile	Output	will be created and replace any file with the same name in the same directory	necessary
treefile	Output	holds genealogies, this file will be created and will replace any file with the same name in the same directory	optional
mathfile	Output	holds plot coordinates for the use in a mathematica notebook, this file will be created and will replace any file with the same name in the same directory	optional
sumfile	Output	holds the summary statistic of the sampled genealogies for further analysis, this file will be created and will replace any file with the same name in the same directory	optional
logfile	Output	logs the progress information that is displayed onto the screen into a file	optional, this file will be created and will repl

2.2.2 Necessary input file

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.

2.2.3 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**.

seedfile holds a random number seed, this is just present for compatibility with PHYLIP, the random number seed can be set in various ways either in the menu or in the parmfile.

catfile hold the categories, for each locus you must give the number of categories, and the value of each category and then a string of category assignments for each site. You can use the # as a commentary character.

```
# Example catfile for two loci with 40 and 30 bp each
#
2 1 10 111111111111111111112222222222222222222
3 1 3 9 111111111122223333333333222222
```


weightfile, for each site and locus you need to give a weight, acceptable weights are integers from 0 - 9 and letters A-Z, A is the weight 10, B 11 and so on, in total there are 35 possible different weights possible. **You need a weight string for each locus.**

```
# Example weightfile for two loci with 40 and 30 bp each
#
1101101101101101101101101101101101101101101101101
33F33F22F22F22F22F22FHFFFFFFHHH
```

2.2.4 Output file

outfile 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.

2.2.5 Optional output files

treefile holds all, only those of the last chain or the best tree(s). The likelihood of each tree is given ($\text{Prob}(\mathcal{D} | \mathcal{G})$) in a comment. The program writes trees with migrations using the Newick format with extensions from the Nexus format, unfortunately I do not know yet a program who can print them nicely. Writing trees to a treefile adds some burden to the program and it will run slower, especially with the option BEST.

mathfile holds the raw likelihood surface data, if this was requested in the options. The name mathfile is the default, but can be changed in the menu or parmfile (see appendix).

sumfile holds the summaries of all genealogies, if this was requested in the parmfile or menu. The name sumfile is the default. His option allows to reanalyze a previous run for likelihood ratio test or profiles.

2.3 How to run

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

```
migrate-n for 1 to n populations
```

Either by double clicking its icon (see on the title page) or for UNIX typing its name in a 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.doc**. 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.

2.4 Menu and Options

You can change the options in the menu (Fig. 2.1) 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: t/t ratio for sequences.

```

=====
MIGRATION RATE AND POPULATION SIZE ESTIMATION
using Markov Chain Monte Carlo simulation
=====
Version 0.9.6
Program started at Thu Jul 13 22:11:58 2000

Settings for this run:
D      Data type
      (currently set: microsatellite model)
I      Input/Output formats
P      Start values for the Parameters
S      Search strategy
W      Write a parmfile
Q      Quit the program

Are the settings correct?
(Type Y or the letter for one to change)

```

Figure 2.1: Top menu of *Migrate*

Menu options can also be changed in the `parmfile`. All possible options are shown in the `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`. Although, I do not recommend being terse, because the `parmfile` is rather hard to read.

2.4.1 Data type

If you chose D in the main menu then you see

```

DATATYPE AND DATA SPECIFIC OPTIONS

1      change Datatype (currently: DNA sequence model)
2      Transition/transversion ratio: 2.0000
3      Use empirical base frequencies? Yes
4      One category of sites? One category
5      One region of substitution rates? Yes
7      Sites weighted? No
8      Input sequences interleaved? No, sequential
0      Start genealogy is estimated using a UPGMA topology

```

```

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

```

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 | Panel-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 is used (this is **much** faster than exact model, but is not a good approximation if population sizes Θ_i are small (say below 10)).

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), 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.

Panel-SNP: the data likelihood is corrected for using a panel of SNP sites, that were polymorphic. The panel has to be population 1.

Genealogies: Reads the `sumfile` (see INPUT/OUTPUT section) of a previous run, with this options the genealogy sampling step will not be done and the genealogies provided in the `sumfile` are analyzed. This datatype makes it easy to rerun the program for different likelihood ratio test or different settings for the profile likelihood printouts.

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.genetics.washington.edu/phylip.html`)

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.

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.

interleaved=<Yes | No >

If your data is interleaved you need to specify this here, the default is **interleaved=No**.

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.

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          111111111222222222
5 0.1 2 5 23 3 1111122223333445555
```

rates=< n : r1 r2 r3 ..rn>

by specifying rates a hidden Markov model or rates is used for the sequences (Felsenstein and Churchill 1995), 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 (**prob-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.

autocorrelation=<Yes:value | No>

if you assume that 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.

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
111111111222222222
111112222AAAA445XXXX5
```

distfile=<Yes | No>

You can supply a distance file for each locus (using PHYLIP syntax). Each individual must have its own name. This option appears in the menu when you choose

0 Start genealogy is estimated using a UPGMA topology

The distance file is then used to create an UPGMA tree with a minimal number of migration events. For large trees this is options help to get better starting trees than the automatic tree generation which uses a rather unsophisticated distance method (differences).

usertree=<Yes | No>

If you specify **Yes** you need a file **intree**. In this file you have starting trees for each locus, **BUT** these trees need to have migration events in them, currently only *Migrate* can write trees with migration events on it, if you inspect such a file you can see, how such an intree file is organized and could insert migration events by hand. If you need this options, please contact me at beerli@genetics.washington.edu

randomtree=<Yes | No>

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.

sequence-error=number

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).

fast-likelihood=<Yes | No>

With very large data sets the common scheme to keep conditional likelihood values in the tree breaks down and a scaling factor is needed to get correct results. If you specify “Yes” the scaling factor is used. This comes with a penalty: the program runs about 20-40% slower!

Microsatellite data

If the **datatype=Microsatellite** is used, the following options have some meaning, please remark that if you use the Brownian motion option these restriction do not apply.

micro-max=value

specifies the maximal allowed number of repeats, this **MUST** be higher than your actual maximal repeat number in your dataset, if it is too high there is a penalty only on allocating to much space and perhaps in slight runtime degradation (the empty space has to be copied), but if it is too small your results will be **wrong!** The default is set to **micro-max=200**.

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 1.6 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**

Electrophoretic data

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

Nucleotide polymorphism

Similar to **sequence data**.

2.4.2 Input/Output formats

This group of options specifies input file names and various output file options. Also, titles for the analysis can be specified. In addition, one can tailor the information the program is presenting

during the execution. Some of the options in this manual are currently not implemented in the two population program (*migrate-0.4*, Beerli and Felsenstein 1999), the n-population version which will eventually replace the two-population version will contain all the mentioned options.

```

INPUT FORMATS
-----
1      Datafile name is infile
2      Use automatic seed for randomisation?  Yes
3      Title of the analysis is <no title given>

OUTPUT FORMATS
-----
5      Print indications of progress of run?  Verbose
6      Print the data?                          No
7      Outputfile name is outfile
8      Plot likelihood surface?                  No
9      Profile-likelihood? Yes, tables and summary
      [Percentiles using exact Bisection method]
10     Likelihood-Ratio tests?                   No
11     Print genealogies?                        None
12     Plot coordinates are saved in mathfile
13     Summary of genealogies will not be saved
14     Save logging information? No

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

```

Figure 2.2: Input/Output menu of *Migrate*

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 **NOT** 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. If 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 (best numbers are divisible by 4 + 1) The default is **random-**

seed=Auto. I personally use always **random-seed=Own:seedvalue**. But then you need to change this for different run, otherwise the sequence of random numbers is always the same.

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**.

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**

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-data=<Yes|No>

Print the data in the outfile. defaults is **print-data=No**.

print-fst=<Yes|No>

Print a table of an F_{ST} estimate for comparison (Beerli and Felsenstein 1999, Beerli 1998) [not recommended].

plot=<No | Yes>[:<Outfile|Both>[:<std|log>:{mig-axis-start,mig-axis-end,theta-axis-start,theta-axis-end}<:printpos<M | Nm>>]]

if **plot=No** then no plot of the parameter space is shown in the outfile, if **Yes** then you can specify whether you want to have the accurate numbers in a separate file (**mathfile**) using *printpos* “pixel” in each direction, or only the ASCII-graphics plot in the outfile. The last option (M or N) let you define whether you want the plot in $\mathcal{M} \times \Theta$ or (default) $4Nm \times \Theta$. Default is **plot=Yes:Outfile**, Example of a more complicated statement: **plot=Yes:Both:std:0,10,0,0.025:100N**

After a run **mathfile** will contain the following

2-pop **locus1**=($(x_{11},x_{12},\dots,x_{1n}), (x_{21},\dots,x_{2n}), \dots, (x_{n1},\dots,x_{nn})$); **locus2**=... the combination of all estimates is the last locus = locus(n+1) the syntax of this file is so that you can import it directly into Mathematica by using **<<mathfile** (see in the example directory of this distribution for more material on this issue). The default is **plot=Yes** which is equivalent to **plot=Yes:Both**.

n-pop The **mathfile** will print only all summed up emmigration and immigration from/into a population, and the format changed to printing only raw numbers: there are *printpos* \times *printpos* cells for each plot (default for *printpos* is 36), so for 2 loci and 3 populations you get a total of 1552 numbers, you can read these into mathematica using **rows=cols=36;**

```

pop=3;
data=ReadList[‘‘mathfile’’,Table[Table[Table[Table[Number, {cols}], {rows}],
{2}],{pop}]; loci=Length[data]; (* now you can do something like the following
after having filled in the xstart, xend etc, looking it up in the outfile
*) ListContourPlot[data[[1,1]]-Max[data[[1,1]],
MeshRange->xstart,xend,ystart,yend,
Contours->0,-1,-2,-3,-4]

```

profile=<No|Yes<:<Fast|Percentile |Discrete|Quick >><:M | Nm >

Print profile likelihood. See section **Likelihood ratio tests and profile likelihood**. Default is **profile=Yes:Percentile:N**.

No: No profile likelihoods are evaluated.

Yes, All: Evaluate profile likelihoods and print tables for each parameter and also a summary table with the approximative percentiles for each variable.

Percentile evaluates the profiles at the percentiles (0.005, 0.025, 0.05, 0.25, 0.50, 0.75, 0.95, 0.975, 0.995). This will need a LOT of time because it has to find the percentiles by evaluating a full maximization for n-1 parameters each

Quick [means quick and dirty] Evaluates the profiled parameter assuming that the parameters (Θ_I and \mathcal{M}_{ji}) are uncorrelated. This is equal to fixing all parameter at the maximum likelihood and evaluate the likelihood for the profiled parameters. This is very fast and often rather close to the *Percentile* option.

Fast A mixture of *Quick* and *Percentile*. This is the default. The percentiles are found using *Quick* and then one final full maximization of all other parameters is done.

Discrete Evaluate the profile likelihood at specific points which are ML-estimate \times (0.02, 0.10, 0.20, 0.5, 1, 2, 5, 10, 50).

M or N The profiles are evaluated using Θ and \mathcal{M} , with the option N (the default) the migration values are printed as $\Theta\mathcal{M}$ (for most data this is $4Nm$, but for mtDNA this could mean it is Nm). With **M** the $\mathcal{M} = m/\mu$ are printed instead of the $4Nm$.

l-ratio=<None | <Mean|Loci>:testparam>

Likelihood ratio tests. See section **Likelihood ratio tests and profile likelihood**. Default is **l-ratio=None**.

print-trees=<All | None | Last | Best>

print genealogies into **treefile**. Remember these trees contain migration events, although I followed the NEXUS rules (Maddison 1998) and the migration events are in comment brackets, I do not know of any program being able to read this kind of trees. I would like to hear from you if you know any other program who can read and display such a tree.

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 **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 give not very much information, except if you are more interested in the best tree than in the best parameter estimate.

Default is **print-trees=None**

mathfile=filename

the plotcoordinates are directed into this file. If you use this option, do NOT use spaces or “/” or on Macs “:” The default is obviously **mathfile=mathfile**.

sumfile=<No | Yes | Yes:filename >

Intermediate results of the genealogy sampling process are save into a file named **sumfile** or into the file for that you specify the filename. You can use this **sumfile** to rerun the program for further analysis, e.g. calculating likelihood ratios or profile likelihoods, see **datatype=Genealogy**.

2.4.3 Start values for the Parameters

```
START VALUES FOR PARAMETERS
-----
1      Use a simple estimate of theta as start?
        Estimate with FST (Fw/Fb) measure
2      Use a simple estimate of migration rate as start?
        Estimate with FST (Fw/Fb) measure
3      Mutation rate is constant? Yes

FST-CALCULATION (for start value)
-----
4      Variable Theta, M symmetric

MIGRATION MODEL
-----
5      Model is set to Full migration matrix model

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

Figure 2.3: ‘Start value for the parameter’ menu of *Migrate*

theta=<Fst | Own:value1,value2>

With **Fst** the programs tries to use an F_{ST} based measure (Maynard Smith 1970, Nei and Feldman 1972) for the estimation of Θ_1 and Θ_2 which are the $4 \times$ effective population size \times mutation rate for each population. **Own: value1, value2** defines arbitrary start values. The default is **theta=Own:1.0,1.0**, which is inappropriate for sequence data where values around 0.01 are more common.

migration=<Fst|Own:value1,value2> (2-POP)

With **Fst** the programs tries to use an F_{ST} based measure (Maynard Smith 1970, Nei and Feldman 1972, Beerli 1998, Beerli and Felsenstein 1999) for the estimation of m_1/mu and m_2/mu . The values for **Own** are given in terms of $4N_e m$ which is $4 \times$ effective population size \times migration rate per generation. The default is **migration=FST**

migration=<Fst|Own:Migration matrix >

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 - }
```

mutation=<Gamma | NoGamma>

If there are more than one locus the program summarizes over all loci. The **Gamma** flag allows for the variation of the mutation rate of each locus according to a Gamma distribution with shape parameter α (alpha) (which is the inverse of the square of the coefficient of variation (CV) of the mutation rate, $CV = \text{standard deviation} / \text{mean}$). This is computationally daunting mostly for numerical reasons: the program is maximizing a product of integrals over all possible mutation rates for each locus likelihood. With **Nogamma** the summarizing step is simply finding the best parameters by maximizing the sum of the log-likelihoods of each locus. The default is **mutation=Nogamma**

2.4.4 F_{ST} calculation

Migrate is using the F_{ST} calculation only to generate starting values for the MCMC runs, when you did not want to give your guess-values for the parameters. With two population and one locus we can only calculate 3 quantities from the data for F_{ST} : the homozygosity within each population and between them. Therefore we only can estimate 3 parameters, either both populations have the same size and different migration rates or the sizes can be different, but the migration rates are the same.

fst-type=<Theta | Migration >

fst-type=Theta

Θ for each population is variable, and the migration rate is fixed.

fst-type=Migration

Migration rate for each population is variable, and Θ is fixed. If the number of populations in the program is bigger than 2 only the option **fst-type=Theta** is available. All pairwise Theta estimates are averaged.

2.4.5 Migration model

If you do not specify anything the joint maximum likelihood estimate of all $n \times n$ parameters are found.

custom-migration=< *NONE* | *migration – matrix* >

The migration matrix contains the migration rates 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
- c: constant value (together with `migration=OWN..` or `theta=OWN..`) [does not work yet]
- *: no restriction

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
m mmm}

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):

```
custom-migration={*000**000**0*000}
```

2.4.6 Search strategy

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

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.

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**.

SEARCH STRATEGY

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 [Burn-in]	10000
8	Combine chains for estimates?	No
9	Heating:	No

Obscure options (consult the documentation on these)		
10	Sample at least a fraction of new genealogies?	No
11	Epsilon of parameter likelihood	100.00
12	Use Gelman's convergence criterium?	No

Are the settings correct?

(Type Y to go back to the main menu or the letter for an entry to change)

Figure 2.4: 'Search strategy' menu of *Migrate*

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 about 200. 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? (**short-chains=value**)

we run most of the time 2 long chains. The first equilibrates and the last is the one we use to estimate the parameters. Default is **long-chains=2**.

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 about 2000. 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 (many) 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. (1995) and is based on 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;number:{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 and an acceptance-rejection step swap between chains so that the the “cold” chains will sample from peaks on the genealogy probability surface proportional to their probability. This scheme is known as MC³ (MCMCMC: Markov coupled Markov chain Monte Carlo), it is based on the work of Geyer and Thompson (1991) and uses for four 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 right that a single chain path could do. You can set the temperatures yourself. There is no default. 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. 1.2 1.5 3.0}**. 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 threadlibrary (often called with slight variations but containg word parts such as pthread, thread, linuxthread), then you can compile the program using “make thread” this could improve speed considerably. **number** try swap every number trees. I normally use 1, but this needs more exploration.

2.4.7 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 unfortunat 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 parameter 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

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

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 has to be smaller than 1.2 for **all** parameters. This can take a very long time.

2.4.8 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 individual, 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**

allelenlength=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 **allelenlength=6**

2.5 Likelihood ratio tests and profile likelihood

2.5.1 Likelihood ratio test

The parameter estimation is done with a maximum likelihood method, this gives the opportunity to easily test different hypotheses against others, when the hypotheses are hierarchical (e.g. Casella and Berger 1996). For example, we wish to test that the migration rates are the same in a two population model with 4 parameters:

$$H_0 : \mathcal{M}_{21} \neq \mathcal{M}_{12} \quad \Theta_1 = \hat{\Theta}_1, \Theta_2 = \hat{\Theta}_2, \quad (2.1)$$

$$H_1 : \mathcal{M}_{21} = \mathcal{M}_{12} \quad \Theta_1 = \hat{\Theta}_1, \Theta_2 = \hat{\Theta}_2, \quad (2.2)$$

and then can test using the test statistics

$$-2 \log \left(\frac{L(\Theta_x)}{L(\hat{\Theta})} \right) \leq \chi_{df, \alpha}^2 \quad (2.3)$$

In the example the degrees of freedom would be two: we are changing two parameters. We need to run `migrate` with the full model: all parameter can vary independently. We get parameter estimates $\hat{\Theta}_1$, $\hat{\Theta}_2$, $\hat{\mathcal{M}}_{21}$, and $\hat{\mathcal{M}}_{12}$. We compare this maximum likelihood with the likelihood when we restrict the migration rate to be the same for example the mean of both estimates. The ratio between these two likelihoods is in the limit (if there is a huge amount of data) χ^2 distributed (Formula 2.3, Figure 2.5). If the probability is smaller than 0.05 we would reject the Null-hypothesis and accept the alternative, saying that the values are not equal. If you have mtDNA data this methods is theoretically not applicable, because you cannot increase the data beyond the full sequence of the mitochondrion, but I am pretty sure that for most situations the test will be still appropriate. There is a problem due to the implementation of the program that we can not allow that parameters go to 0.0. A parameter of 0.0 has a 0.0 probability. Tests against 0.0 need a halved significance level, because we truncate at 0.0, and therefore are testing only one-sided (...cit...).

Do not forget that these likelihoods are only approximations. Comparison with exact likelihoods for genealogies with 3 tips and no migration show that the MCMC curves are exactly the same as the “exact” curves. When the program is not run long enough the MCMC curves tend to be wider than the “exact” curves and have their maximum biased towards the parameter value at which we run the chains. We expect when there are many sampled individuals that it is likely that you run the program not long enough and therefore will get wrong confidence interval estimates and will stick too close to the start parameters. (Figure 2.6). You can check for this by running the program several times from very different start values. Just looking at the point estimates, is probably not enough, you need to inspect the profile likelihoods too. Most of the time it seems that real single locus data is not very great for the estimation of migration rates and the “confidence” intervals are huge.

For the `parmf` there is an option `l-ratio` which you can use to define a hypothesis against the program run (Null-hypothesis). You can repeat the statement for testing more than one hypothesis, but you may need to correct your significance level for multiple tests. The syntax is:

l-ratio:<Means|Loci> <:param1,param2,param3,...paramn*n>

Means over all loci

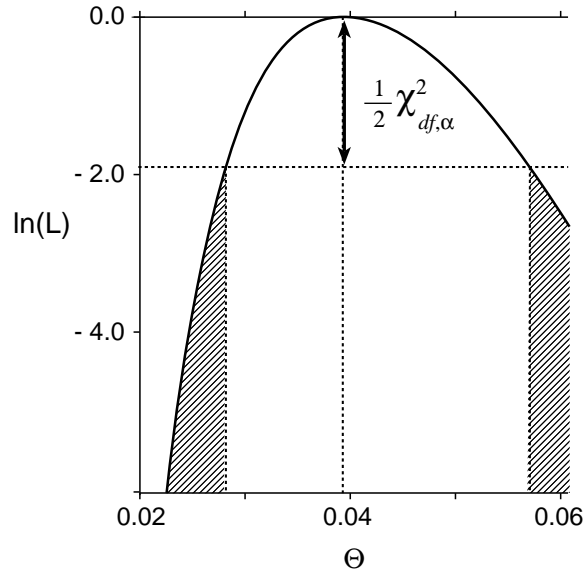


Figure 2.5: Likelihood ratio test: dashed areas are outside of the 95% confidence limit. Θ is $4N_e\mu$; $df = 1$, $\alpha = 0.05$

loci for each locus, this may not be valid for sequences, the likelihood ratio test assumes convergence if the sample size goes to infinity, but with a finite sites model and one locus this can not be achieved, so the the χ^2 statistic may not be appropriate.

The syntax for each **param1, param2,...** is rather complicated: **param1 = <* | x | m | value>**

* the value is the same as the one from the estimate (= H_0)

x the value will be maximized.

m the value is the mean of the parameters, either Θ or $4N_e m$.

value is any arbitrary value you want to test against the H_0 .

Examples for two populations for the parmfile entries:

l-ratio=Means:0.01,0.011,1.0,1.1;

l-ratio=Means:*,*,m,m;

l-ratio=Means:x,m,*,0;

The parameters are ordered according to the following rule:

$\Theta_1, \Theta_2, \dots, \Theta_n, 4N_e^{(1)}m_{2,1}, 4N_e^{(1)}m_{3,1}, \dots, 4N_e^{(1)}m_{n,1}, 4N_e^{(2)}m_{1,2}, 4N_e^{(2)}m_{3,2}, \dots, 4N_e^{(2)}m_{n,2}, \dots, 4N_e^{(n)}m_{(n-1),n}$

Although you specify $4Nm$ the program evaluates \mathcal{M} for the test and prints $4Nm = \Theta\mathcal{M}$. This seems more accurate, then the parameters Θ and \mathcal{M} are uncorrelated.

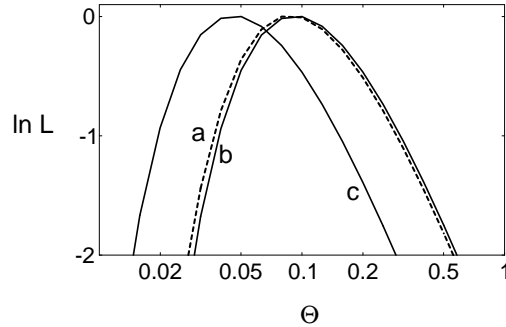


Figure 2.6: Log likelihood curves from (a) the exact likelihood calculation for a genealogy with 3 samples, (b) an MCMC based estimator with only one (1) sampled genealogy with start value $\Theta_0 =$ Watterson estimate, (c) with one acceptance using a $\Theta_0 = 0.00001$. The data are 3 sequences each 1000 bp long and generated with a $\Theta = 0.1$, running the program some 1000 genealogies delivers a likelihood curve indistinguishable from the exact likelihood curve.

Example with 3 populations based on the following migration matrix:

$$\begin{array}{ccc} - & 2 & 1 \\ 1.8 & - & 1 \\ 0.5 & 0.6 & - \end{array}$$

results in the string

l-ratio=Loci:*,*,*,2,1,1.8,1,0.5,0.6;

Do not forget the semicolon, the current program is picky and needs it

2.5.2 Profile likelihood

Parameter estimation in high dimensions causes serious problems in the presentation of results: for 2 population we have 4 parameters, with 8 population 64, etc. One would like to show the high dimensional surface but we are crudely limited to 3 and perhaps can understand graphs up to five. Showing one parameter at a time only shows us a transection through the solution space, but is perhaps the best we can do. By using profile likelihoods we can trace a parameter and also see how the other parameter change at given values for our profile parameter. Instead of finding the parameters at the maximum likelihood, we fix the profile parameter at some arbitrary value and then maximize the other parameters at that profile likelihood. This constructs a path through the solution space, which we can use to construct approximate confidence limits using the likelihood ratio test criteria (Fig 2.7) with a degree of freedom of 1 (well, this is true in “asymptopia” but may produce very tight confidence intervals (see Beerli and Felsenstein 2000). Several advanced statistic textbooks discuss the use of likelihood ratio and the related profile likelihoods (e.g. Casella 1996), but I like the compact, and in my opinion, very readable, short text of Meeker and Escobar (1995).

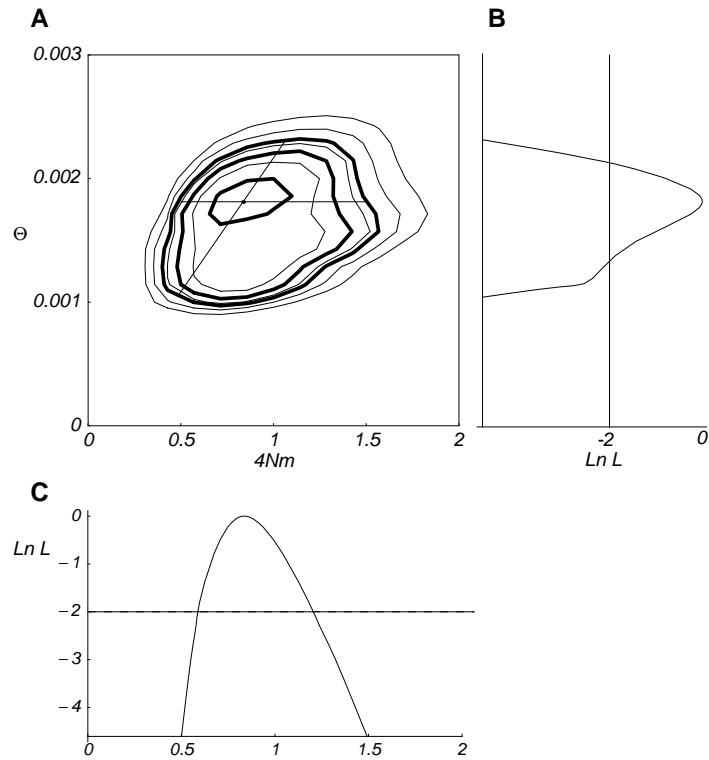


Figure 2.7: Profile likelihood, for a series of values of a parameter, the other parameter are maximized and the likelihood given that parameter is highest along the straight lines in A. (A) Contour plots for a run with two variables, the thick lines are the 50%, 95%, and 99% confidence contours. (B) is the profile likelihood curve for Θ and (C) is the profile likelihood curve for $4Nm$ (based on \mathcal{M}). The 95% confidence range for B and C are for values with log likelihood values above -2.

Chapter 3

Accuracy of results

3.1 Monitoring progress

The program will show additional information if the **progress** flag is set (**progress=Yes** is the default). You can even see more with **progress=verbose**. With **logfile=filename** all progress is also directed into this logfile, the default name is **logfile**. The progress is report similar to the following screen dump fragment for each chain and each locus. I added a line number which is not part of the output (Y means standard progress report, V are the additional lines in verbose mode).

```
01Y 11:49:01  Start conditions: theta={811.90959,0.03487}, M={140.99436,0.00000},
02Y          Start-tree-log(L)=-93.678120
03Y 11:49:01  Equilibrate tree (first 200 trees are not used)
04Y 11:49:03  Long chain 1: lnL=0.21525 ,
05Y          theta={0.04026,0.05527}, M={83.96647,45.78351}
06V          Sampled tree-log(L)={-98.760356 .. -93.035062}, best in group =-93.019453
07V          log(P(g|Param)) -20 to -18 -16 -14 -12 -10 -8 -6 -4 -2 0 All
08V          Counts          0 0 0 0 0 0 0 0 0 144 56 200
09V          Maximization steps needed: 134
10V          Coalescent nodes: 0 1 2 3
11V          population 0: * - - -
12V          population 1: - - - *
13Y          Acceptance-ratio = 1095/2000 (0.547500)
.....
14Y 11:49:09  Final parameter estimation over all loci
15Y
16Y          <paste in correct part>
17Y
18Y 11:49:09  Program finished
```

The values reported should give some hints how the program progresses through the sample space. The tree likelihoods (line 06V) should go steadily up until a peak in the likelihood surface has been reached. It can go down through a valley of bad values and either recover on the same peak or another one. If this process runs long enough it is guaranteed that it will find the global maximum. But the program is not searching the tree-likelihood maximum, it searches through the space defined by $\text{Prob}(\mathcal{D} | \mathcal{G})\text{Prob}(\mathcal{G} | \mathcal{P})$ and its maximum is not necessarily at the highest tree likelihood. The “histogram” (07V, 08V) of the $\text{Prob}(\mathcal{G} | \mathcal{P})$ reflects this. The histogram is scaled so that the best value is 0. If most of the values are in the topmost class the estimate is probably in good accordance with the trees, otherwise the process should run longer. Of course if all genealogies

are in the topmost class one could wonder if the process is sampling different trees at all, but this can be checked with the acceptance ratio. If the Acceptance ratio (13Y) drops below 10% consider to run the program with ten time longer chains just to sample enough different genealogies, so that the parameter estimates are not governed by a few genealogies only.

If the single locus maximization step needs more than 200 iterations (09V), please send a report, then it should find most of the time the maximum in fewer than 50 iterations.

If you have chosen to discard the first few trees using **burn-in=value**, you will see line (3Y).

3.2 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.

3.3 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 and 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 then if 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.
- If the results do not change much, perhaps you can stop. Otherwise increase the length of the chains, increasing the increment (e.g. **short-inc=100** and **long-inc** does not increase memory usage, but run-time. You can also increase the number of sampled genealogies (**short-sample** or **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 than 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, I am thinking of implementing a parallel processing machinery (based on MPI) that would distribute the loci onto different machines.

3.4 How to avoid conflicts with other computer users

The run time **and the memory usage** of `migrate` is highly dependent on the number of populations, the length of the chains, and the number of loci. It is common that a single locus data set can run for many hours even very fast machines, resulting in runs of many days for multilocus data sets. For some users this can produce a problem, either the system administrator or other users gets mad about you consuming “all” resources, this is mostly CPU and for large data sets also memory.

For UNIX systems the immediate, but perhaps wrong, answer to this people is that these demanding programs are one of the reasons to use these fast computers; a run of `migrate` does normally not compromise any editing, mail reading, word processing on shared machines. To free a terminal you can put `migrate` into background and log out.

1. Run `migrate-n`
2. Change the menu as you think is appropriate.
3. In the main menu use **(W)rite a parmfile**.
4. Kill the program (Control-c) or use **(Q)uit**.
5. Edit the `parmfile` and change the entry `menu=YES` to `menu=NO` and any other option you want to change. If you intend to run the program several times you should change for each run the `random-seed=OWN:somenumber`.
6. Rerun the program with

```
nohup (nice migrate-n > migrate.log ; date | \ mail -s 'migrate finished' youremailaddr  
&
```

the `nohup` allows you to logout without stopping the program, additionally potential output is logged into `nohup.out`. The `nice` causes to program to run slower when other users are using the machine “unniced”. On servers the nicing often happens automatically after some time or they have a specific batch system, ask you system administrator what’s best for a long run.
7. logout or do something else, you will get mail when `migrate` has finished, if you are curious and want to know when approximately it will finish peek into the file `migrate.log`, but do not save it.

For Windows and especially Macintosh systems the program is unfortunately not a so good citizen and is disturbing other programs. To run long `migrate` on these machines the best way is to run this on a private machines, where you have the control.

Chapter 4

Presentation of results

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

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*
	plot of the likelihood surface into mathfile	o
α -histogram	Table of shape values versus $\log(\text{likelihood})$, α is varying whereas the other parameters are held constant at the maximum of the surface.	o
Profiles	Profile likelihood tables	o*
Percentiles	Percentiles table, summary of profile tables	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) and Hudson et al. (1989). In the Appendix you can find a sample outfile with some comments.

4.0.1 Walk through an outfile

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

Title and Options

```
=====
Example for sequence data
=====
MIGRATION RATE AND POPULATION SIZE ESTIMATION
```

using Markov Chain Monte Carlo simulation
 =====
 Version 0.7

Program started at Sun May 22 23:40:38 1998
 finished at Mon May 23 00:25:32 1998

Options in use:

Datatype: DNA sequence data
 Random number seed (with internal timer) 674365543
 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
 Gamma-distributed mutation rate is not used
 Markov chain parameters:
 Short chains (short-chains): 10
 Trees sampled (short-inc*samples): 10000
 Trees recorded (short-sample): 500
 Long chains (long-chains): 3
 Trees sampled (long-inc*samples): 100000
 Trees recorded (long-sample): 5000
 Number of discard trees per chain: 200
 Print options:
 Data file: infile
 Output file: outfile
 Print data: No
 Print genealogies: No
 Plot data: Yes, to outfile and mathfile
 Profile likelihood: Yes, tables and summary

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: 1

Population	Individuals
1 population_number_0	25
2 population_number_1	21
Total of all populations	46

Empirical Base Frequencies

Locus	Nucleotide				Transition/ Transversion ratio
	A	C	G	T(U)	

```
-----
1      0.2461  0.2450  0.2497  0.2591      0.60000
```

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.

Parameter estimates

```
=====
MCMC estimates
=====
Population [x] Loc. Log(L)   Theta   4Nm
                [4Ne mu]   1,x     2,x
-----
1: population    1  2.88  0.04567 ----- 4.03909
2: population    1  2.88  0.02857 7.80435 -----
```

Comments:
 There were 10 short chains (500 used trees out of sampled 10000)
 and 3 long chains (5000 used trees out of sampled 100000)

This is the main output of the program. For each population there is a list of all loci and the estimates and if there are more than one locus, there is also an estimate over all loci. The $\ln(L)$ is the maximum log likelihood. This value is a ratio $\ln(L) = \ln(L(\mathcal{P})/L(\mathcal{P}_0))$. The parameter \mathcal{P}_0 are different between different runs of the program and therefore you cannot simply compare between different runs.

The column marked Theta (Θ) gives the population sizes for each population and each locus, of course the number of individuals in that population N_e is for all loci the same, and the variance you see is (a) the variance of the sampler, (b) stochastic variance due to the coalescence process, (c) variance of the mutation rate. The migration parameter $4Nm$ is to read the following way: in population 1, the **2,x** means that the immigration from population two into one is $4N_1m_{21} = 4.039$. in population 2 the **1,x** means that the immigration from population one into two is $4N_2m_{12} = 7.804$. If the program is also allowing for variable mutation rate (you don't want to use that with one locus), then you will get also an estimate for the shape parameter alpha (α) for the distribution of the mutation rates.

F_{ST} table

This will not be shown as a default, anymore. It is merely used as a starting value for the Maximum likelihood estimates. The table are similar to the table of the MCMC estimates.

Likelihood surface plots

```
Log-Likelihood surfaces for each of the 2 populations
-----
```

Legend:

X = Maximum likelihood
 * = in approximative 50% confidence limit
 + = in approximative 95% confidence limit
 - = in approximative 99% confidence limit

Locus 1

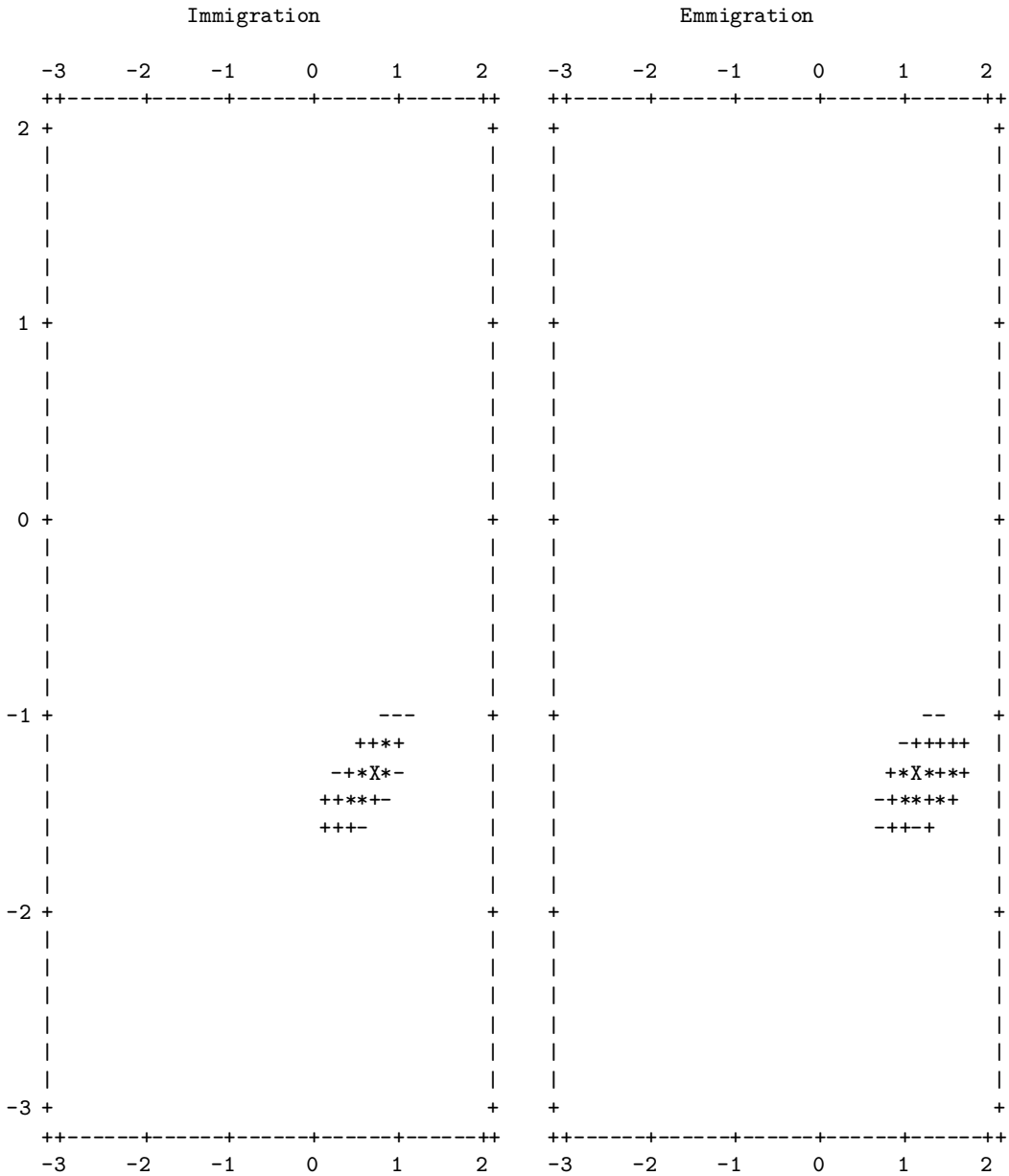
x-axis= 4Nm [effective population size * migration rate],
 y-axis = Theta,
 units = log10

Maximum log likelihood on plot

Population 1: population_number_0

Immigration: 4Nm=5.179470, Theta=0.051795, log likelihood=2.678661

Emmigration: 4Nm=13.895000, Theta=0.051795, log likelihood=2.749731



For each population and each locus there will be a summary contour plot for all immigrations and all emigrations. These plots give some information about the confidence you should have in the estimates. Keep in mind that even with two populations there are 4 parameters and the likelihood. A plot is a kind of diagonal through this high dimensional space (in this example: 5 dimensions);

Profile likelihoods

Profile likelihood for parameter Theta_1
 Parameters are evaluated at percentiles
 using cubic splines of profiled parameter
 (faster, but not so exact).

Per.	Ln(L)	Theta_1	*Theta_1*	Theta_2	M_21	M_12
0.01	-3.645	0.0223	0.0223	0.0297	81.9303	293.8230
0.05	-2.065	0.0240	0.0240	0.0297	81.9441	294.2779
0.10	-1.329	0.0250	0.0250	0.0297	81.9766	294.5011
0.25	-0.284	0.0266	0.0266	0.0296	82.0709	294.7953
0.50	2.878*	0.0457	0.0457	0.0286	88.4385	273.2104
0.75	0.324	0.0789	0.0789	0.0279	96.2738	252.5011
0.90	-1.065	0.0900	0.0900	0.0277	97.4555	251.1683
0.95	-1.910	0.0966	0.0966	0.0277	98.0544	250.5631
0.99	-3.884	0.1119	0.1119	0.0276	99.2213	249.4557

- = not possible to evaluate, most likely value either 0.0 or Infinity
 in the parameter direction, the likelihood surface is so flat
 that the calculation of the percentile(s) failed.

The profile likelihood table give you some idea how the parameters vary when we hold one constant. In the default setting the program tries to find the parameter values that are at percentiles. How this is done for Θ_1 : (1) calculate the likelihood value for a few values smaller and bigger than the ML-estimate. (2) calculate a spline function. (3) find the Θ_1 that is at the percentile x using the splines. (4) recalculate the likelihood and maximize the other parameter again using the full formula. In the example, Θ_1 varies almost independently from the others, looking more closely it seems that Θ_2 slightly shrinks while Θ_1 grows.

Summary of profile likelihood tables

=====
 Summary of profile likelihood percentiles of all parameters
 =====

Parameter	Lower percentiles				
	0.01	0.05	0.10	0.25	0.50
Theta_1	0.02228	0.02399	0.02497	0.02664	0.04567
Theta_2	0.00946	0.01188	0.01331	0.01567	0.02857
M_21	30.53718	36.49126	39.97529	46.64759	88.43845
M_12	114.08445	132.49441	143.22648	163.32323	273.21045

Parameter	Upper percentiles				
-----------	-------------------	--	--	--	--

	0.50	0.75	0.90	0.95	0.99
Theta_1	0.04567	0.07889	0.09003	0.09660	0.11190
Theta_2	0.02857	0.05709	0.07586	0.09833	0.15052
M_21	88.43845	201.06595	215.26333	225.18048	245.85767
M_12	273.21045	805.85153	896.08361	957.07762	1083.66503

- = not possible to evaluate, most likely value either 0.0 or Infinity
in the parameter direction, the likelihood surface is so flat
that the percentiles cannot be calculated.

This summarizes only the likelihood and profile parameter column in the profile likelihood tables and can be used to give some idea about the confidence you should have into the estimates. Θ_1 has a approximative 90%-confidence interval from 0.02399 to 0.09660 with a best estimate of 0.04567. (the data was simulated with a $\Theta_1 = 0.05$, for further “true” values see the README in the example directory.

Chapter 5

Frequently asked questions, errors and warnings, and troubleshooting

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

5.1 Questions

5.1.1 General

1. The program crashes! Your program has a bug!
2. The program crashes with large but not with small data sets, what is wrong?

5.1.2 About the datafile

1. How can I code haploid data for *Migrate*?
2. Can I use haplotype frequencies as input?
3. Can I use gene frequencies as input?

5.1.3 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?

5.1.4 About reading the outfile

1. I have haploid data, what is Θ ?
2. I have mtDNA sequence data what is Θ ?
3. How should I interpret each of the $4Nm$ estimates for pair i,j ?

4. Why are the Likelihood values different between runs?
5. Why do I have positive numbers in the Ln(L) column?
6. I have problems to understand what are the Null-hypothesis and the alternative hypothesis in the likelihood ratio test section.

5.2 Answers

5.2.1 General

1. **The program crashes! Your program has a bug!**
 Sure, this program most likely 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.
2. **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.

5.2.2 Data file related

1. **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.

2. **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. I principle it would be easy to add a “frequency” input mode, but currently I have not time to do that. But keep asking for it, if this is so important to you.
3. **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 Perl¹, Mathematica², or C³.

5.2.3 About option 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).

¹freely available for Windows, Mac, and UNIX, check <http://www.perl.org>

²nice, but not free software

³freely available for almost all systems see <http://www.gnu.org> [Free Software Foundation]

2. How long does it run?

With `progress=Yes` 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: No. **Long Answer 1:** If you use the `heating` option and your machine is a symmetric multiprocessor machine and you compiled with `make thread` 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:** No, but you can do it yourself manually. See the file `HOWTO-PARALLEL`. I am currently working on an implementation of migrate that will use the message passing interface (MPI), and this would spread the loci over processors, but this update is not yet coded.

5.2.4 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. How should I interpret each of the $4Nm$ estimates for pair i,j ? as $4 \times N_i \times (4 \times N_e \times \text{migration rate from } j \text{ to } i)$? Then, can I take 17 times (for 18 demes) this number as $4 \times N_i \times (\text{migration rate into } i)$

Yes, the overall immigration rate into i is

$$\text{total immigration rate into population } i = \sum_{j \neq i} 4N_e^{(i)}m_{ji}$$

but I personally tend to report

$$\mathcal{M}_{ji} = m_{ji}/\mu = 4N_e^{(i)}m_{ji}/\Theta_i \text{ or } \mathcal{M}_{.i} = \sum_{j \neq i} \mathcal{M}_{ji}$$

4. Why are the likelihoods between runs different?

The likelihoods are really ratios

$$\frac{L(\mathcal{P})}{L(\mathcal{P}_0)} = \frac{1}{m} \sum_i^m \frac{\text{Prob}(D | g_i) \text{Prob}(g_i | \mathcal{P})}{\text{Prob}(D | g_i) \text{Prob}(g_i | \mathcal{P}_0)}$$

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.

5. 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

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

its MCMC derivation is

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

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.

6. **I have problems to understand what are the Null-hypothesis and the alternative 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 $4N_2m_{12} = 3$. Now you want to test if 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** $4Nm$, but that you still need to specify your values to check in terms of $4Nm$. 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 it in several ways:

- l-ratio=Means:m, m, m, m [easiest]
- l-ratio=Means:0.0265, 0.0265, 0.34, 0.67

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 average. Because the input for the likelihood ratio test is in terms of $4Nm$ and not \mathcal{M} .

5.3 Errors and warnings displayed by migrate

The errors are in no particular ordering, but I will move more important ones to the beginning of their sections.

5.3.1 Errors

The program aborts when it encounters one of the following conditions. Of course there are certainly conditions I have not thought of.

SEVERE ERROR: Most often your `infile` contains a problem (e.g. number of sites does not match the number actual sites given, number of individuals does not match). If you fail to correct the problem. please contact me.

ERROR: Datatype is wrong, please use a valid data type!

ERROR: the program will crash anyway, so I stop now

You probably specified a wrong letter for the data type in the `parmfile`

ERROR: Wrong datatype, only the types a, m, s, n ERROR: (electrophoretic alleles,

ERROR: microsatellite data,

ERROR: sequence data,

ERROR: SNP polymorphism) are allowed.

You probably specified a wrong letter for the data type in the menu

ERROR: The parmfile contains an error on line XX

There was a wrong entry or even more likely wrong values in the `parmfile` on line `xx`.

ERROR: Inconsistency between your Menu/Parmfile and your datafile

Most likely your `parmfile` assumes there are `n` subpopulations and you assume `m` subpopulations. Problems with the migration matrix are likely.

ERROR: There is a conflict between your menu/parmfile

ERROR: and your datafile: number of populations are not the same

Most likely your `parmfile` assumes there are `n` subpopulations and you assume `m` subpopulations.

ERROR: cannot find seedfile

You specified that the random number is in `seedfile`, but the file is not present in the directory `migrate` is running.

ERROR: Failure to read seed method, should be

ERROR: seed=auto or seed=seedfile or seed=own:value

ERROR: where value is a positive integer

Either seed specification in `seedfile` or `parmfile` is wrong.

ERROR: Failure to read start theta method, should be

ERROR: theta=FST or theta=Own:x.x

ERROR: or theta=Own:{x.x, x.x , x.x,}

ERROR: migration=Own:migration value

the start parameters are not correctly specified.

ERROR: Failure to read start migration method

the start parameters are not correctly specified.

ERROR: Custom migration matrix was completely set to zero?!

the custom migration matrix was not correctly specified.

5.3.2 Warnings

WARNING: migration limit (xx) exceeded: yy

WARNING: results may be underestimating migration rates

WARNING: for this chain

If this happens only a few times in short chains, don't worry. If it happens in the last chain or very often, then your migration estimates will be most likely underestimated, but the migration rates between these populations will be very high, anyway. It means that there is an upper limit of possible migration events on the genealogies, and this is set as a default to `number_of_populations × 1000`.

WARNING: Migration forced

WARNING: results may overestimate migration rates

WARNING: for this chain

Migration rate is essentially 0.0, the program proposes sometimes a migration event even so the probabilities would force a coalescence, this heuristic helps to escape the fatal attraction to 0.0. If $4Nm$ is smaller than 0.1 the program will propose randomly every tenth event a migration event. This genealogy has then still to be accepted. Hitting this boundary can produce an upwards bias, but it should be only be recognizable when your populations are barely connected, if at all.

WARNING: This does look like sequence data

WARNING: I just read a number of sites=0

WARNING: If you use the wrong data type, the program will abort

Check your datatype!

WARNING: _____

WARNING: Target branch problems with time=xx

WARNING: _____

If you encounter this, abort the program, and try to find the error in the `infile`, but if the data prints correctly, please contact me. Probably I should declare this a severe error and abort.

WARNING: proposed and new likelihood differ: xx != yy

WARNING: abort the program and try to find the errors

WARNING: there could be a wrong datatype, or infile

WARNING: to check the data you can print it (see menu)

If you have problems to resolve this error (check for errors in `infile`), please contact me and try to

give as much information as you can (including your dataset).

WARNING: Inappropriate entry in parmfile: *keyword* ignored

The *keyword* of a parmfile entry was wrong, often misspelled.

WARNING: You forgot to add your guess value:

WARNING: Theta=Own:pop1,pop2, ...

WARNING: or Theta=Own:guess_pop (same value for all)

You probably specified Theta=Own and forgot to say what values.

WARNING: You forgot to add your guess value, use either:

WARNING: migration=FST

WARNING: or migration=Own:{guess_4Nm} (same value for all)

WARNING: or migration=Own:{ - 4Nm21 4Nm31 ... 4Nm12 - 4Nm23 ... }

You probably specified migration=Own and forgot to say what values. See the parmfile section, about how to give the migration values.

5.4 Troubleshooting

If you think you have found a bug please report this to

beerli@genetics.washington.edu. I would like to know every warning you see while you compile the program, if you send me bug-reports please include your hardware and system specifications, your `infile`, your `parmfile` (if any), and a “printout” of the warnings or errors.

BUT, mostly, the problem is that the data in the `infile` is in a wrong format: you can expect the program to crash when you try to use the `datatype=Allelic` and your `infile` contains sequence data. I am trying to reduce the number of strange error messages, but this has lower priority than adding new features/improving code.

Please, before you report a bug, compare your `infile` with the examples.

Chapter 6

Program distribution and installation

6.1 Program availability

Migrate can be fetched from our www-site (<http://evolution.genetics.washington.edu/lamarc.html>) and is free for non commercial use. Currently we have the following packages available

migrate.tar.gz	Source
migdoc-0.9.pdf	Documentation [this document you are reading]
migrate-0.9.src.pm.sea.hqx	Source for Powermac (Metrowerks)
migrate-0.9.powermac.sea.hqx	Powermac binaries
migrate-0.9.bsdintel.tar.gz	Dec Alpha DUNIX binaries
migrate-0.9.macosx.tar.gz	Mac OS X server binaries
migrate-0.9.linux.tar.gz	LINUX binaries
migrate-0.9.solaris.tar.gz	SUN Solaris binaries
migrate-0.9.win.exe	WindowsNT/9*/2000 self extracting archive

6.2 Installation

6.2.1 Binaries

On UNIX system unpack with `tar xvfz migrate.[system].tar.gz` or `gunzip -c migrate0.9.[system].tar.gz | tar xf -`. This builds a directory `migrate-0.9` 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.

6.2.2 Source

The program is known (at least to me) to compile on every UNIX machine that has a decent ANSI compatible compiler. I compiled it myself successfully on the following UNIX machines: Linux INTEL (Redhat 5, 6), Linux PPC (Yellowdog Champion 1.2), NeXT Cube (NEXTSTEP 3.3), SGI

Origin (SGI ?), G3 (Mac OS X beta), Compaq Alpha 500au (Tru64 v5.0), SUN (Solaris 2.6). And on the following non-UNIX machines: PPC (Mac OS 8, 9), INTEL (Windows 95, 98, 2000).

UNIX

1. **gunzip -c migrate0.9.tar.gz | tar xf -** or **tar xzf migrate0.9.tar.gz** this creates a directory "migrate-0.9" with "src", "contribution", and "examples" in it.
2. **cd migrate-0.9**
3. **./configure**
(this script 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 appropriate 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) 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)

Powermac (MAC OS 8 or 9)

The source code for the Powermac is the same as the general source code but it is packaged with a minimal graphical interface file and a Metrowerks Codewarrior project, which should make it very easy to compile (if you have a very recent Metrowerks compiler).

1. Unpack (it is a self extracting archive).
2. Open the **migrate.mcp** file and use the submenu **Make** (I compiled with Metrowerks CodeWarrior Pro 5)

Chapter 7

Miscellaneous

7.1 Wish list

- Send me a reprint if you used **Migrate** for your publication [people obviously do not read that far in this documentation, because did not happen yet]
- Cite the documentation and our paper, see below.
- Report problems to `beerli@genetics.washington.edu`
- Suggestions (if you need these improvements very soon, add a check so that I can hire a programmer to implement all those 😊)

7.2 How to give credit

Please cite:

Beerli, P. 1997-2000. MIGRATE: documentation and program, part of LAMARC. Version 0.9. Revised November 26, 2000. Distributed over the Internet, <http://evolution.genetics.washington.edu/lamarc.html> [Downloaded: ...date....]

Beerli, P., and J. Felsenstein. 1999. Maximum likelihood estimation of migration rates and population numbers of two populations using a coalescent approach. *Genetics* 152(2): 763-773.

Beerli, P. 1998. Estimation of migration rates and population sizes in geographically structured populations. In: *Advances in molecular ecology* (Ed. G. Carvalho). NATO-ASI workshop series. IOS Press, Amsterdam. Pp. 39-53.

7.3 Copyright

(c) Copyright 1996-2000 by Peter Beerli and Joseph Felsenstein, Seattle. Permission is granted to copy this document and the program *Migrate-n* and *Migrate* provided that no fee is charged for it and that this copyright notice is not removed.

7.4 Acknowledgement

This project is and was supported by grants from National Science Foundation (USA) BIR 9527687 and National Health Institutes (USA) GM51929 and HG01989 all to Joseph Felsenstein and a fellowship of the Swiss National Science Foundation to Peter Beerli (1994-1996). I thank Mary K. Kuhner and Jon Yamato for help during debugging and many discussion.

And also all people who thought it worth to report errors and foggyness in menu and explanation: Mats Bjorklund, Allen Rodrigo, Carol Reeb, Byron Adams, Tony Metcalf, Toby Hay, Peter Galbusera, Scott Edwards, Reinaldo Brito, Tonya Bitner, Vicki Friesen, Cliff Cunningham, Natalie Bulgin, Keith Crandall, Eric Simandle, Martin Damus, Ron Goldthwaite. [List is unordered and certainly incomplete].

Chapter 8

Literature

- Beerli, P.** 1998. Estimation of migration rates and population sizes in geographically structured populations. In *Advances in molecular ecology* (Ed. G. Carvalho). NATO-ASI workshop series. IOS Press, Amsterdam. Pp. 39-53.
- Beerli, P., and J. Felsenstein.** 1999. Maximum likelihood estimation of migration rates and population numbers of two populations using a coalescent approach. *Genetics* 152(2): 763-773.
- Casella, G., and R. L. Berger** 1990. *Statistical inference*. Duxbury Press, Belmont, California.
- Chib, S., and E. Greenberg.** 1995 Understanding the Metropolis-Hastings algorithm. *American Statistician* 49: 327-335.
- Di Rienzo, A., A. C. Peterson, J. C. Garza, A. M. Valdez, M. Slatkin, and N. B. Freimer** 1994. Mutational processes of simple sequence repeat loci in human populations. *Proc. Natl. Acad. Sci. USA* 91: 3166-3170.
- Geyer, C.** 1994. Estimating Normalizing Constants and Reweighting Mixtures in Markov Chain Monte Carlo. Technical report, University of Minnesota Nr. 568 R(4).
- Geyer, C. and E. A. Thompson.** 1994. Annealing Markov chain Monte Carlo with Applications to Ancestral Inference. Technical report, University of Minnesota Nr. 589 R(1).
- Felsenstein, J.** 1993. PHYLIP 3.5: Phylogeny Inference Programs. Pro gram package and documentation distributed by the author. Department of Genetic s, University of Washington, Seattle.
- Felsenstein, J. and G. A. Churchill.** 1996. A hidden markov chain approach to variation among sites in rate of evolution. *Genetics* .
- Hammersley, J. M. and D. C. Handscomb.** 1964. *Monte Carlo methods*. Methuen, London.
- Hudson, R. R.** 1990. Gene genealogies and the coalescent process. *Oxford Surveys in Evolutionary Biology* 7: 1-44.

- Kimura, M. and T. Ohta. 1978.** Stepwise mutation model and distribution of allelic frequencies in a finite population. *Proc. Natl. Acad. Sci.* **75**: 2868-2872 .
- Kingman, J. F. C. 1982a.** On the genealogy of large populations. pp. 27-43 in *Essays in Statistical Science*, ed. J. Gani and E. J. Hannan. London: Applied Probability Trust.
- Kingman, J. F. C. 1982b.** The coalescent. *Stochastic Processes and their Applications* **13**: 235-248.
- Kishino, H. and M. Hasegawa. 1989.** Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea. *Journal of Molecular Evolution* **29**: 170-179.
- Kuhner, M. K., J. Yamato, and J. Felsenstein. 1995.** Estimation effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. *Genetics* **140**:1421-1430 .
- Kuhner, M. K., P. Beerli, Jon Yamato, and Joseph Felsenstein. 2000.** Usefulness of single nucleotide polymorphism (SNP) data for estimating population parameters. *Genetics* 156(1).
- Maynard Smith, J. 1970.** Population size, polymorphism, and the rate of non-Darwinian evolution. *American Naturalist* **104**: 231-237
- Meeker, Q., and L. A. Escobar. 1995** Teaching about approximate confidence regions based on Maximum Likelihood estimation. *American Statistician* 49: 48-53.
- Nath, H., B., and R. C. Griffiths. 1993.** The coalescent in two colonies with symmetric migration. *Journal of Mathematical Biology* **31**: 841-851.
- Notohara, M. 1990.** The coalescent and the genealogical process in geographically structured population. *Journal of Mathematical Biology* **29**: 59-75.
- Ohta T. and M. Kimura. 1973.** A model of mutation appropriate to estimate the number of electrophoretically detectable alleles in a finite population. *Genetical Research* **22**: 201-204.
- Slatkin, M. 1995.** A measure of population subdivision based on microsatellite allele frequencies. *Genetics* **139**: 457-462.
- 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.
- Valdes, A. M., M. Slatkin, and N. B. Freimer. 1993.** Allele frequencies at microsatellite loci: the stepwise mutation model revisited. *Genetics* **133**: 737-749.

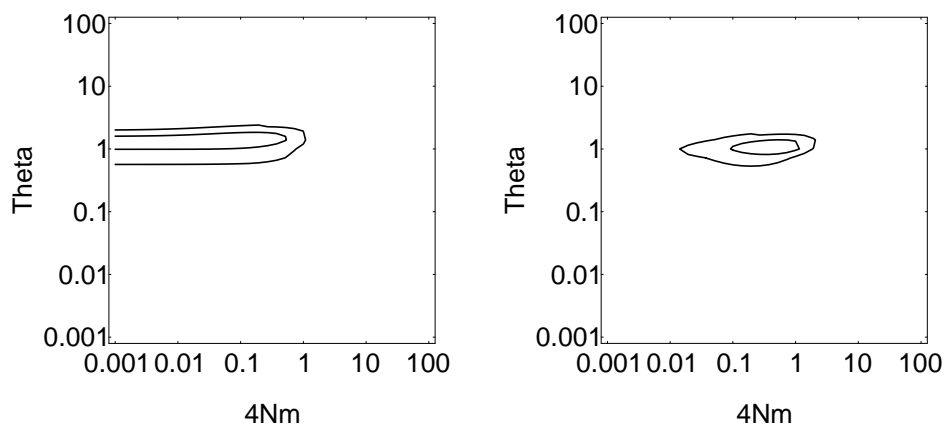
Chapter 9

Appendix

9.1 Mathematica plot package

If you have access to the program `Mathematica`, you can open the `lamarc.example.ma` in the `example` directory. With it you can create nicer likelihood surface plots than the ones you see in the outfile.

Example:



9.2 History and persistent problems

[people] in brackets helped to find bugs/problems.

November 26 2000 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 run around 20-40

October 18 2000 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 2000 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 2000 MIGRATE-N 0.9.8 Bug fix: The reanalysis of a sumfile with one locus failed in the profile likelihood calculation. Addition of safeguards for machines (Dec Alphas) that return for $\exp(-1000) = \text{NaN}$ 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_120 chains.

July 28 2000 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 2000 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 2000 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/:1wrong/old: 0.5The 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[mutation=Gamma is still broken, sigh]

May 30 2000 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 2000 MIGRATE-N 0.9.3 embarrassed 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 2000 MIGRATE-N 0.9.2 inconsistency in likelihood calculation with replication fixed.

April 21 2000 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 2000 MIGRATE-N 0.9 First introduction of estimation of parameters over multiple chains or multiple runs. The strict two-population version Migrate-0.4 is removed from this distribution, although it will be still available separately. Multiple chain/runs with the combination of gamma deviated mutation rate does not work yet. Heating scheme is broken. Registered users: 430

December 10 1999 MIGRATE-N 0.8.5 Change of defaults: plot=FALSE, moved eventloop() in plot routine for Macintosh.

December 2 1999 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 1999 MIGRATE-N 0.8.3 Revision of heating scheme [note it is still broken].

November 5 1999 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 Bittner], Profile Quantile table, verbose Progress reporting) Registered users: 372

September 7 1999 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 1999 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 1999 MIGRATE-N 0.7.1 Division by 0 bug fixed in fst-calculation, this seems to bother only DEC Alphas.

May 19, 1999 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.

- Feb 14, 1999 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's with out gcc.
- Oct 11, 1998 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 contains now more about 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, 1998 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 correctness 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, 1998 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, 1998 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, 1998** 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, 1998** 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.
- March 18, 1998: MIGRATE 0.4
Update of the manual, but still not complete. 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: 103

August 20, 1997: 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 no 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, 1997: 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 much longer chains than the proposed default. Try ten times longer "long" chains. Or use the option "moving-steps". Registered users:38

May 12, 1997: 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, 1997: MIGRATE 0.2a

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, 1997: 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