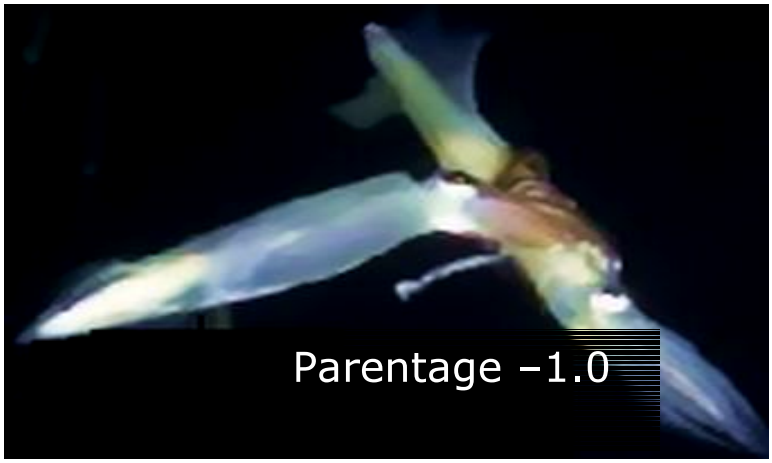
Parentage version 1.0

Users Guide



Ian Wilson
Department of Mathematical Sciences
University of Aberdeen
King's College,
Aberdeen AB23 3UE,
UK

Email:
I.Wilson@maths.abdn.ac.uk

Parentage Home Page:
http://www.maths.abdn.ac.uk/~ijw

Manual v0.01: Wednesday, 22 November 2000

COPYRIGHT NOTICE

(c) Copyright 2000 by Ian Wilson. Permission is granted to copy this document provided that no fee is charged for it and that this copyright notice is not removed.

Contents

Introduction

Background

The rationale for the development of this package was a problem in assessing the paternity of squid hatchings from egg strings (Emery *et al.*). We could be sure (up to laboratory error or contamination) that all eggs on a string had the same mother - although the mother's genotype was uncertain, further we had information on the allele frequencies from the breeding population. We were interested in, and did not know, the number of fathers, and the relationship between hatchlings on the egg strings (either half or full-sibs). This was a difficult inferential problem. This program was developed to draw inferences for this problem: namely shared maternity with an unknown number of fathers and no additional information except for allele frequencies in the breeding population. To improve statistical properties of the model – and so that the assumptions about the mother can be checked – we allowed more than one maternal genotype, which means that the model was more generally applicable than for the string of squid hatchlings. I also added options to allow some maternal or paternal genotypes to be entered, and for some relationships to be known, which expanded the possible problems that could be attacked.

Depending on the problem we may want to know the number of fathers, or the relatedness between individuals, or possibly information such as the relatedness of fathers or the mutation rate. As a consequence this package has been designed to be as general as possible. There are a number of scenarios where we want to know about parentage for a set of sampled individuals. The program is designed to be flexible so that it can work from making inferences about the genotype of the father of a set of full-siblings to making inferences about the number of mothers and fathers for a set of individuals when we have no information about fathers or mothers at all. The program will work with all situations with intermediate levels of information also.

Parentage is the program written to analyse the problem of inferring the number of parents and the relationships within samples. Early in its development I realised that the general modelling framework could be used to analyse a number of different problems with the same structure, so the program was developed to allow flexibility in its use, enabling us to include additional information in the form of breeding population, the genotypes of some or all of the potential parents, and even partial knowledge about maternity and paternity. The program is written in ANSI C, and, depending on the system on which you run it will either need to be compiled or will be distributed as an executable. This manual is written to try to help other users with the program, which through time constraints and philosophy (there is little point in writing software to produce graphics or edit genotype data when there are already a host of programs out there that can do the same thing) is command line driven. Subsequent versions may be in the form of libraries to add to either the R (free) or the S-plus (commercial) statistics packages that will allow you to use their extensive graphical capabilities directly.

The structure of this manual is: a brief introduction to the Bayesian Paradigm and Markov chain Monte Carlo methods; a description of its use; the format of input and output files; examples of its use on selected problems and a reference to all input options. The appendix contains a description of some R code for pre and post-processing of input and output from the program.

The Bayesian Paradigm

In Bayesian inference probability models are fitted to data and the results summarised as a posterior probability distribution of model parameters and unobserved random variables (Gelman *et al.* 1995). With complex problems it is generally not viable to calculate the posterior distribution directly, however, advances in computationally intensive statistical techniques, in particular Markov chain Monte-Carlo (MCMC), make it possible to sample from the distributions of interest for problems with complex dependencies. Inference can then be based on summary statistics from these samples. Under the Bayesian paradigm we treat everything in the model as a random variable, the number of fathers, the number of

mothers, the fathers for each individual hatching and the genotypes of both parents and the offspring – what would generally be regarded as data are then observed parameters (so that there is no uncertainty). This allows us to bring a variety of paternity assignment problems under the same framework.
Under the Bayesian (or direct probability) paradigm, inferences are made on the posterior probability distribution of variables of interest, conditional on observed data and prior models. This posterior density is proportional to the prior probability (before any data are observed, based on our knowledge about the problem) multiplied by the likelihood of observing the data under the model. Our data consist of the vectors **Y** and (possibly) **B**, the genotypes of the hatchlings and our sample from the breeding population. Vector **Y** consists of elements, $\left(y_l^{(i,1)}, y_l^{(i,2)}\right)$, the genotype of the *i*th individual at the *l*th locus where $i=1,\ldots,N$ and $l=1,\ldots L$, where N is the number of offspring sampled and L the number of loci. Vector **B** is similarly defined, with $i=1,\ldots N_b$, the sample size from the breeding population.

Also included in our model are the random vectors of mothers, **M**, and fathers, **F**. These vectors are defined similarly to **Y** and **B**, so that **M** consists of elements, $\left(m_l^{(i,1)}, m_l^{(i,2)}\right)$, with $i=1,\ldots,n_m$, the number of mothers. These may be observed, unobserved, or may be partially observed. If they are unobserved then they are updated in the Markov chain.

The sibling relationships within the sample are described by the parental vectors of fathers, $\mathbf{a_f}$ and mothers, $\mathbf{a_m}$. The n_f fathers are labelled from 1 to n_f, with element $a_f^{(i)}$ giving the father of individual *i*. Similarly, mothers are labelled from 1 to n_m and the mother of *j* is $a_m^{(j)}$. Again these may be observed on unobserved.

Inference from the model consists of drawing samples from the unobserved random variables and other parameters of the model. The framework for inference described here means that any combination of unobserved random variables may be put into the model.

Note that for unobserved mothers and fathers the labels in the random vectors of mothers and fathers are arbitrary (for example, we can swap the labels of males 1 and 2 with no change to the sibling relationships) and thus there are $n_f! \times n_m!$ different labellings that give the same set of sibling relationships.

In order to make inferences about the parentage of the offspring we require probability models for the observed offspring and breeding population genotypes, the parental genotypes and sibling relationships. These involve modelling the mutation process, the distribution of parental genotypes and patterns of sibling relationships in the sample. We model our problem more generally than is needed for the application here, as we allow for the possibility of more than one maternal genotype.

Modelling paternity and maternity share

The most basic assumption is that each male is equally likely to be the father of an individual in the sample, so that the joint probability of paternity vector and number of fathers is:

$$\Pr(\mathbf{a}_f, n_f) = \frac{\Pr(n_f) n_f!}{n_f^{\,n}}, \tag{1}$$

where $P(n_f)$ is the prior probability of n_f males and we have the factor $n_f!$, as each labelling of males is equally likely. This model may be over-simplistic, and we allow differential male success using two models: based on a multinomial-Dirichlet number of offspring, and on the Ewens' sampling formula (Ewens 1972), a distribution used to describe the distribution of alleles in population genetics for the infinite allele model.

We allow two models for the allele frequencies in the breeding population. The first is that we know the breeding population frequencies without error, so that the probability of sampling genotypes are simply the product of multinomial distributions. The second model uses the Dirichlet distribution to model the prior density of allele frequencies in the breeding population at each locus (Balding & Nichols 1995), so that the genotypes at a locus are multinomial-Dirichlet distributed.

The likelihood of the data, **Y**, conditional on M**, F**, $\mathbf{a_f}$, $\mathbf{a_m}$ and μ is calculated assuming simple Mendelian inheritance; ignoring mutation, the probabilities can be 0, ¼, ½ or 1. Furthermore, we assume that conditional on parent genotypes, offspring are independent, so that this likelihood may be multiplied over marker loci and individuals. We include the possibility of mutation or mis-scoring of marker alleles (for our models these are not discriminated, and we use mutation for both). We assume that mutations occur independently and at a constant rate of μ per gamete across all loci and at an equal rate for maternal and paternal gametes. Following a mutation from an allele *i*, the mutated allele is equally likely to be any other allele present with equal probability, including the original allele, *i*. This simplifies calculations so the probability that a gamete mutates from allele *i* to allele *j* is μ/K_l where K_l is the number of alleles at locus *l*. More realistic models are possible, however our focus of interest is not on the mutation process.

Statistical inference

We collectively denote **M, F** and other unobserved variables (such as the mutation rate μ) by θ. Our problem is to draw meaningful and correct inferences about $\mathbf{a_f}$ and $\mathbf{a_m}$ conditional on our observed data **Y** and **B**. From Bayes' rule we have

$$\Pr(\mathbf{a_f}, \mathbf{a_m}, \theta \mid \mathbf{Y}, \mathbf{B}) \propto \Pr(\mathbf{Y} \mid \mathbf{a_f}, \mathbf{a_m}, \theta, \mathbf{B}) \Pr(\mathbf{a_f}, \mathbf{a_m}, \theta \mid \mathbf{B}) \quad . \qquad (5)$$

The RHS of (5) can be expanded using the distributions described above to give

$$\Pr(Y \mid \mathbf{a_f}, \mathbf{a_m}, \mu, \mathbf{M}, \mathbf{F}) \Pr(\mathbf{M}, \mathbf{F} \mid \mathbf{B}) \Pr(\mathbf{a_m} \mid \alpha) \Pr(\mathbf{a_f} \mid \beta) \Pr(\mu) \Pr(\alpha) \Pr(\beta) .$$

This gives a complete specification of the model, given prior distributions for the mutation rate μ, and the Ewens' sampling formula parameters α and β.

We use a reversible jump MCMC approach to inference. A Markov chain with an equilibrium distribution proportional to the RHS of equation (5) is constructed in the space of all the unobserved variables (**M, F**, $\mathbf{a_f}$, $\mathbf{a_m}$, μ, α, and β). After a suitable burn-in period, samples are collected from the chain.

Markov-Chain Monte Carlo

There are so many possible combinations of parental genotypes and ancestries that it is impossible to exhaustively search them all. Instead we use a Monte-Carlo method to generate samples from the distribution of possible parents and ancestries proportional to their probability under the model. However, again the problem is too complicated with a complex dependency structure, so that we cannot simulate directly from the model. The solution is to define a Markov chain with a *stationary* distribution equal to the distribution of interest – the distribution of parents and ancestry conditional on observing the sample genotypes and perhaps some other background information.

We use a combination of Gibbs sampling, Metropolis-Hastings updates and Reversible Jump Markov chain Monte Carlo (Green 1995) to move around the space of possible combinations of parents.

Multiple Chains

The strong structure within the data may cause difficulties with mixing for the chains, particularly with large datasets (from experience more than 100 offspring and at least five loci). To deal with these difficulties the program can use multiple chains, of which only one has the correct target distribution, the others are hopefully faster mixing, and we have an extra step of swapping between chains. This technique is called metropolis-coupled MCMC (Brooks 1998). In our use of this the additional chains are sampled from distributions with posterior densities proportional to f^{α_i}, where f is the correct density, this give a flatter distribution and hence flatten the constraints.

Input

This program works from a prompt, under both Windows and unix, with the syntax

```
parentage (infile) (outfile) (seed)
```

where brackets indicate optional arguments. Parentage will also work by double clicking on the icon in Windows (if double clicking then the program will assume the default command line values – or a shortcut may be used, and then the command input may be altered by right clicking on the short-cut icon).

The infile (*default infile*) contains information about:

1. Data, in the form of file names and paths;
2. The probability model: a variety of options that can change the model that is run, and the priors that are needed for the model;
3. Program options such as the length of the MC run and how often we sample from it.

The outfile (default outfile) gives the name of the main output file for the program. It also acts as the root stem for the rest of the output files. More details are given in the Output section.

Inputting Data

The general input style for all genotype data is the same. Each genotype is input on a single line as pairs of alleles representing the genotype at each locus. The data can be input as positive integers, or letters, or a mixture of both. Missing data are input using –1 or ?. Comments can be put into the file by using the hash sign (#). Anything on a line after a hash sign will be ignored.

The main datafile is given by

```
datafile:  inputdata
```

An example of a datafile is in the distribution in Data/Neff.data, from Neff et al. (2000). The first few lines are:

```
#  data from Neff, …
88 98 217 227 118 128
88 98 217 227 118 128
88 98 217 227 118 128
88 98 217 227 118 152
88 98 217 247 118 128
88 98 217 247 118 152
```

These data are for 3 loci.

If you want the output as letters then setting `format: L`, will enable this. This is only safe for a datafile only comprising of letters.

Breeding Population Samples

If we have additional information on the alleles present in the breeding population from sampled genotypes then this information can be incorporated by using the line

```
freqfile: freqfilename
```

where `freqfilename` is the name (with or without path) of a file with containing the genotypes as in the datafile.

The further command

`freqmodel: <0 or 1>`
can also be used to set up the way that this additional information is used. Using
`freqmodel`: 0
then a dirichlet prior for the allele frequencies is used, whereas with
freqmodel: 1
the observed frequencies are taken to be the population frequencies. The program will exit with an error if freqmodel: 1 is used when there are alleles in the sample that are not present in the background frequencies. If no breeding population frequencies are available then a Dirichlet prior is used, with equal frequencies for all the alleles present in the sample.

Genotypes of Parents Known

When we know the genotypes of some or all of the potential parents then using

`fathersfile: fathersfilename`
or
`mothersfile: motherfilename`

enter them into the program. If these are the only potential fathers or mothers then fathersprior or mothersprior must be set to a constant value – otherwise the program may infer additional parents. If there are some data missing from these genotypes then the program will include these in the model as unobserved random variables, and perform Gibbs' sampling on the missing components. Using this method it is possible to put some ancestral relationships into the model without knowing the genotypes of the ancestor, by inputting the parentages that are known as described in the next section.

Parentage of Some Offspring Known

If we know either the father or mother of some offspring (and additionally we have already set the genotypes) then using either
`knownfathers: knownfatherfile`
or
`knownmothers: knownmotherfile`

We may input some, or all of the ancestral relationships.

These files must consist of a vector of the sample length as the number of samples, and consists of a list of integers which are either positive integers (less than or equal to the number of known fathers) or 0 when the parent is not known.

For example if we have a sample of size 10 and we have 3 known fathers then file knowfathersfile could consist of

1
2
1
3
3
1
1
0
0
0

This would indicate that the fathers of the first seven are known but that the mother of the last three at not known.

Input of Program Settings

```
thinning:
burn-in:
samples:
```

Thinning gives the number of MCMC updates (or attempted updates) are tried between sampling the chain. This is called the thinning interval in Brooks (1998). ***Burn-in*** gives how many samples to throw away (note that the actual number of MCMC steps is ***`burn-in`*** $\times$ ***`thinning`***). The total number of samples to take is given by samples.

```
chains:
```

If chains is set then metropolis coupled chains are used. The number of extra chains are determined by the number of values given in chains. For example to use two additional chains with posterior densities proportional to $f^{1/2}$, and $f^{1/3}$ use:

```
chains: 2 3
```

Input of Probability Model

Modelling Relative Parent Frequencies

Three models for how different parents share offspring are possible.

Incorporating Prior Information

Prior information for alpha is input into the program using a line in the input file, which looks like:

```
lambdaprior: prior
```

where prior can be a number, (or where more than one number is appropriate, a series of numbers in brackets), or a distribution. Hence valid prior declarations are:

```
lambdaprior: normal(10,1)
lambdaprior: 10
lambdaprior: constant(10)
lambdaprior: uniform
```

Note that lambdaprior is not used in the model, it is purely an example. The range of priors available is given in the table below. The only prior that is always set is `muprior`, the prior for the mutation rate, the other priors that are set determine the offspring model.

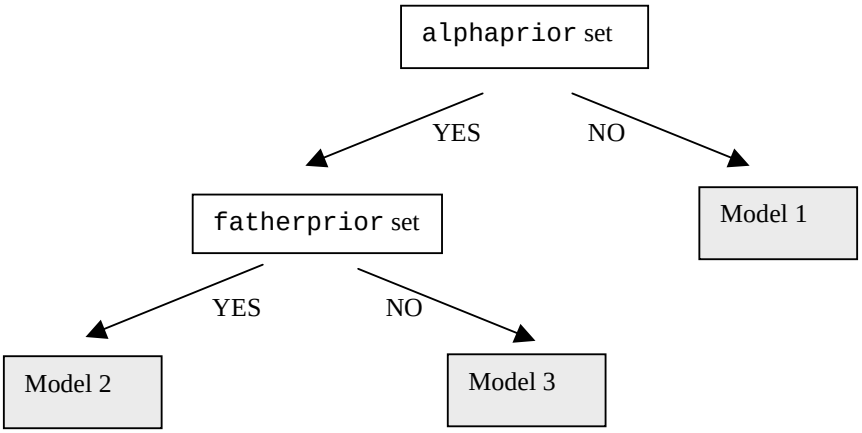
Other priors that may be set are for the number of fathers (`fatherprior`) and mothers (`motherprior`), and/or for the distribution of offspring between males (`alphaprior`) and females (`betaprior`).

The set of priors that are entered for `fatherprior` and `alphaprior` (the same is also true for `motherprior` and `betaprior`) determine the model used for the offspring numbers. The default values for both of these priors are null and you get one of:

Model 1: Each male is equally likely to be the father of any offspring,

Model 2: The male share is given by a Dirichlet distribution, with a prior on the number of fathers
Model 3: The number of fathers and male share are given by the Ewens' sampling formula.

The way this is decided is given in the following schematic:



If no prior is given for the number of fathers for model 1, then the prior on the number of fathers is set to be uniform between the upper and lower end of the range (default 1-10).

Possible Priors

Distribution	Use	Mean	Variance
X ~ gamma	`gamma(a,b)` note that if b=1 this is the exponential with mean a	a/b	a/b^2
X ~ uniform	`uniform(a,b)`	$(a+b)/2$	$(b-a)^2/12$
X ~ normal	`normal(`μ,σ^2`)`	μ	σ^2
X ~ lognormal	`lognormal(`μ,σ^2`)`	$\exp(\mu+\sigma^2/2)$	$\exp(2\mu+\sigma^2)(\exp(\sigma^2)-1)$
X ~ poisson	`poisson(`λ`)`	λ	λ
X ~ geometric	`geometric(`θ`)`	$1/\theta$	$(1-\theta)/\theta^2$
uniform constant	`uniform constant(10)` or `10`	*undefined*	*undefined*

For example if we think that a mutation rate of 0.0001 should be used we use

```
muprior:  0.0001,
```

whereas if we believe that a gamma with parameters 1 and 10000 is appropriate we use

```
muprior: gamma(1,10000),
```

The use of constant priors is discouraged for the number of fathers or mothers as this may not allow good mixing of the chain. A better approach is to use something like:

```
motherprior: normal(constval, 0.1)
```

This strongly penalises incorrect values on the "cold" chain, but allows values away from this on the "hotter" chains and hence may improve mixing.

Output

The name `outfile` from the command line gives the filename of the main output file, described below. It also forms the basis of other output files, constructed by postfixing to the stem `outfile` to get `outfile.something`, where there are a number of postfixes that give different aspects of the posterior distribution.

The Output File

The main output file gives simple summary measures of the data at each sampling point, such as the number of fathers, the number of mothers, the mutation rate, the number of mutations and the posterior density of the chain. The line below gives the maximum number of lines that may be out put, some of these may be missing. The top line of the output file gives the names of columns present out of those below.

```
nfathers  nmothers  nmuts  alpha  beta  mu  log_posterior  log_like  log_sample  log_prior
```

where

`Log_posterior` is the log of the posterior probability.

`log_like` is the log of the product of the probability of the data conditional on the parental genotypes for each individual

`log_sample` is the probability of sampling the maternal and paternal genotypes from the background allele frequencies

`log_prior` is the log of the prior probabilities.

Fathers File

The fathers file: `out.fathers` gives the paternal genotypes and the number of offspring for each of the fathers. each line has the format:

sample father_number #offspring $g_{11}\, g_{12}\, g_{21}\, g_{22}\, g_{31}\, g_{32}\, \ldots\, g_{l1}\, g_{l2}$

where the number of loci is l. An example output file is given below.

1 3 5 106 102 212 214 144 144
1 1 9 100 104 210 212 146 142
1 2 7 100 102 212 214 152 150
2 3 3 106 106 212 214 144 144
2 1 9 100 104 210 212 146 142
2 2 9 100 102 212 214 152 150
…

For this output the first two samples both have 3 fathers, the first with 5, 9 and 7 offspring, the second with 3 9 and 9 offspring. The genotypes of the fathers are given on the rest of the line. S-plus/R code for the analysis of fathers' files are described in the appendix.

If all the potential paternal genotypes are known then none are output, only columns 1-3. If there is only a single offspring, as in example 1, then only two columns, the first with the sample number and the second with the father's label will be output.

Mothers File

The mothers file is has name `outfile.mothers`, and has the same format as the fathers file.

Shared Paternity and Maternity

The output file:
`<outfile>.paternity c`onsist of $n \times n$ matrices with the number of times that individuals share the same father in the upper-trangle of the matrix and the number of times they share the same mother in the lower-triangle of the matrix. The diagonal of the matrix is just the number of runs as an individual always share paternity and maternity with itself. The appendix contains R functions that can postprocess and produce informative plots from this information.

Table 1: The file out.paternity, a matrix with shared paternity and maternity

1000	521	429	112	Here we have the output from an analysis of four
990	1000	312	59	individuals. We have 1000 samples and 521 times out of
921	950	1000	873	1000 individuals one and two share a father, these two
980	993	810	1000	individuals share a mother 990 times out of 1000.

Similarly individuals one and three shared fathers 429 times and mothers 921 times out of 1000.

Shared Parentage

Similar information is given in the output file
`<outfile>.parentage,`
which consists of the number of times that individuals are full siblings, and half siblings. This output file consists of an $n \times n$ matrix. The upper-triangle of the matrix is the number of times the individuals are full siblings, the lower triangle the number of times individuals are half-siblings.

Mutations File

The file
`<outfile>.muts`
gives information about mutations. This file consists of an $n \times l$ matrix. Each column of the matrix is for a locus and each row for an individual. The file gives the number of times (out of the total sample size) that at least one mutation is inferred for an individual at a locus, given the genotypes of the parents. A typical file will look like the table to the left, for a dataset with 3 loci.

```
0     0     0
0   324     0
0     0     1
0     0     1
```

Case Studies

In this section a selection of analyses will be performed on some case studies, which vary from being very simple tests of the method, as in the first two cases, to rather more involved. The input files, and data files are included in the directory `examples`.

Case Study 1

The files for this are in directory `examples\case1`. We have the genotype of a child. We know the mother's genotype. What is the relative probability that each of three males is the father of the child – including the possibility of mutation?

If we have potential fathers' genotypes:

```
1/1 2/2 1/2 1/1
1/2 2/3 1/1 1/3
1/3 2/1 1/3 1/2

And a mother

1/2 1/2 1/1 1/2

And the child has genotype

1/2 1/3 1/2 2/3
```

None of the potential fathers can be the father without at least a single mutation. In this case, as we only have three potential fathers the background allele frequencies can have no effect on which parents we choose. Assuming a k-allele mutation model with 3 alleles and ignoring all but the smallest powers of μ, the probabilities of the offspring for the three fathers given in the table below

	Locus 1	Locus 2	Locus 3	Locus 4	Overall
Father 1	½	$\mu/6$	½	$\mu/6$	$\mu^2/144$
Father 2	½	¼	$2\mu/3$	¼	$\mu/48$
Father 3	¼	$\mu/3$	$\mu/2$	$\mu/3$	$\mu^3/72$

We can get a Monte-Carlo estimate of the relative probabilities (and check the program for μ=0.005) using the input file in the test box:

```
datafile: child
fatherfile: fathers
motherfile: mother
motherprior: 1
fatherprior: 3
samples: 50000
burn-in: 10
thinning: 10
freqmodel: 1
muprior: 0.005
```

The data files are in the directory `examples\case1`.
The expected relative probabilities from the table above are: 0.0016639, 0.99832, and 1.6639×10^{-5}.
The output file `outfile.males` gives the index of the males (in this situation as all males are specified and there is only a single offspring). The observed proportions are: 0.00188, 0.99810, and 0.00002, which are not significantly different from the expected proportions. Note that the program uses higher powers of μ for its calculations.

Case Study 2 – Data with no variation

A very simple test of the program can be performed on data with no variability, setting the mutation rate to be very low. In this situation we should return the prior distributions for the number of fathers and mothers. We must set `freqmodel=1,` so that we do not allow for the possibility of more alleles in a breeding population.

The sub-directory Case2 in the examples directory contains 3 input files: infile1, infile2 and infile3.

```
datafile: testdata
burn-in: 100
thinning: 400
samples: 2000
muprior: 0.000001
freqmodel: 1
useloci: 1
fatherprior:
uniform(1,10)
motherrange: 1 20
motherprior:
poisson(2)
```

`infile1` is for model 1. A prior is set on the number of males, and the number of females. Each male is equally likely to be the father of any offspring, and each female equally likely to be the mother. The paternity and maternity for any individual are independent. Hence the number of offspring that the fathers have will be multinomially distributed with expected number $n \times 1/k$ if there are k fathers.

```
datafile: testdata
burn-in: 100
thinning: 400
samples: 2000
muprior: 0.000001
alphaprior:
gamma(1,2);
betaprior: gamma(1,2)
freqmodel: 1
useloci: 1
fatherprior:
uniform(1,10)
motherrange: 1 20
```

`infile2`: For model 2 – a prior for the number of fathers, and mothers, and a Dirichlet prior for the offspring share for both.

```
datafile testdata
burn-in: 100
thinning: 400
samples: 2000
muprior: 0.000001
alphaprior:
gamma(1,2);
betaprior: gamma(1,2)
freqmodel: 1
useloci: 1
fatherrange: 1 20
motherrange: 1 20
```

`infile3`: For model 3 – a prior for the Ewens' sampling formula parameter for the density of the number of parents and the offspring share.

The posterior numbers of fathers and mothers agreed with the priors for all three models.

Case Study 3 – Data from Neff (2000)

Table 2: Neff data from Neff et al (2000) table 5.

Lma102	Lma120	Lma87
88/98	217/227	118/128
88/98	217/227	118/128
88/98	217/227	118/128
88/98	217/227	118/152
88/98	217/247	118/128
88/98	217/247	118/152
88/98	217/247	118/152
88/98	217/247	118/152
88/98	217/247	118/152
88/98	217/247	118/152
88/98	227/227	118/128
88/98	227/227	118/152
88/98	227/227	118/152
88/98	227/227	118/152
88/98	227/231	128/152
88/98	227/247	118/128
88/98	227/247	118/128
88/98	227/247	118/128
88/98	227/247	118/128
88/98	227/247	118/152
88/98	227/247	118/152
88/102	217/227	118/128
88/102	217/227	118/128
88/102	217/227	118/128
88/102	217/227	118/152
88/102	217/227	118/152
88/102	217/247	118/128
88/102	217/247	118/128
88/102	217/247	118/128
88/102	217/247	118/128
88/102	217/247	118/128
88/102	217/247	118/152
88/102	227/227	118/128
88/102	227/227	118/128
88/102	227/227	118/152
88/102	227/227	118/152
88/102	227/227	118/152
88/102	227/247	118/128
88/102	227/247	118/152
88/102	227/247	118/152
88/102	227/247	118/152
98/98	211/231	118/128
98/98	227/231	128/128
98/98	231/245	118/128
98/102	211/231	128/128
98/102	211/231	128/152

mean 0.85, median 0.87
mean 0.86, median 0.87

Neff *et al.* (2000) present a method for determining the share of paternity for a number of models, and apply their techniques to a nest of Bluegill Sunfish (*Lemomis macrohirus)*. Here, as a case study, we apply our method to their data (for our method we need the population genotype frequencies, which were supplied by Dr B. Neff). The background frequencies were converted into genotypes.

The data consist of the genotype of the nest guarding male: 88/88, 227/247 118/118, and population frequencies, and the genotypes of 46 offspring from the nest. These data files are in the Data folder in the distribution. The input file for this analysis is:

```
datafile: ../../Data/neff.data
fatherfile: ../../Data/neff.father
freqfile:  ../../Data/neff.breeding
alphaprior: gamma(1,4)
betaprior: gamma(1,4)
burn-in: 200
samples: 2000
thinning: 400
```

and the file is in `Examples/Data`. We choose again to analyse this problem using the Ewens' sampling formula for the number of fathers and the relative share. Another possibility is to perform the analysis with a Poisson number of fathers and a Dirichlet prior for the share of offspring (use poisson(1), and alphaprior(1,1)). The results are similar for the proportion of offspring from the guarding male (shown in table). The main aim of inference here was the share of paternity for the guarding male, but we can also estimate the number of females in the nest, and the sibling relations for the offspring.

share of paternity – number that father 1 is the father to

number	<30	30-34	35-39	40	>=41
proportion	0.018	0.0415	0.117	0.804	0.0195
	0.005 5	0.0145	0.081	0.884	0.015

Number of Mothers	Number of Fathers				
	2	3	4	>=5	total
1	0.0125	0.0060	0.0005	0.000	0.019
2	0.2000	0.2525	0.1210	0.0415	0.615
3	0.0935	0.1105	0.0465	0.0125	0.263
4	0.0330	0.0305	0.0095	0.0050	0.078

>=5	0.0125	0.0085	0.0040	0.0000	0.025
total	0.3515	0.4080	0.1815	0.059	1

```
      2      3      4      5
1 0.0185 0.0060 0.0010 0.0005
2 0.4235 0.2110 0.0415 0.0100
3 0.1395 0.0705 0.0140 0.0025
4 0.0330 0.0150 0.0025 0.0005
5 0.0065 0.0025 0.0015 0.0000

   2      3      4      5
0.6210 0.3050 0.0605 0.0135
attr(,"class")
[1] "table"
> table(no[,2])/2000

    1      2      3      4      5
0.0260 0.6860 0.2265 0.0510 0.0105
attr(,"class")
[1] "table"
```

Case Study 4: Data from Kichler et al. 1999.

Input File Options

burn-in

Used	To give the number of warm-up steps to be taken before starting to sample from the chain. This is to allow the chain to reach an equilibrium, and to remove dependence on the initial state. Note that the actual number of iterations of the chain is ***burn-in*** $\times$ ***thinning***
Input	integer
default	100
restrictions	integer > 0
Example	burn-in: 1000 Discards 1000 steps before taking a sample
See Also	samples, thinning

chains

Used	Do you want metropolis-coupled chains? See the introduction for a discussion of when this may be useful
Input	list of real numbers; the tempering values for the additional chains
default	empty
Example file	/examples/subsetting/infile

datafile

Used	To give the filename for the data file with the offspring in. This file is ASCII with the genotypes given as integers, or characters with 2 columns for each locus
Input	name
default	datafile
restrictions	The file must be present in the directory – or in the path given
Example	`datafile: ../../data/off.data` reads a datafile off.data from the directory ../../data
Example file	/examples/subsetting/infile
See Also	freqfile, fatherfile, motherfile

fatherfile

Used	To give the filename for the data file with genotypes of potential known fathers – if you also make the number of fathers constant using fatherprior, then this gives the total pool of fathers for the sample. If this is not set then the set of potential fathers are sampled proportional to their probability.
Input	name
default	
restrictions	If set, the file must be present in the directory – or in the path given
Example	freqfile: ../../data/fathers.data reads a datafile off.data from the directory ../../data
Example file	/examples/subsetting/infile
See Also	datafile, fatherfile, motherfile

fatherprior

Used	To set the prior for the number of fathers. The list of available priors is set out in the input section.
Input	Distribution type with parameters
default	uniform - improper uniform prior
Example file	/examples/subsetting/infile
See Also	motherprior, muprior

fatherrange

Used	To set the range of possible fathers.
Input	pair of integers
default	1 10 – a range of 1 to 10 inclusive
Example file	/examples/subsetting/infile
See Also	fatherprior, motherrange

femalesfile

Used	Do you want to output a file of females (mothers) genotypes and maternity share
Input	0 (no) or 1 (yes)
default	1
See Also	malesfile

format

Used	To set the output type
Input	a letter, N for numbers, or L for letters
default	N
Example file	/examples/subsetting/infile
See Also	datafile

freqfile

Used	To give the filename for the data file with the genotypes of the sampled breeding population genotype frequencies. This file is ASCII with the genotypes given as integers with 2 columns for each locus
Input	name
default	
restrictions	If set, the file must be present in the directory – or in the path given
Example	freqfile: ../../data/back.data reads a datafile off.data from the directory ../../data
Example file	/examples/subsetting/infile
See Also	datafile, fatherfile, motherfile

freqmodel

Used	To determine the modelling used for sampling parents from background frequencies. The models can be simple multinomial sampling if the background frequencies contain all the alleles that are seen in the data, or multinomial-Dirichlet, if there are alleles in the data that are not in the background frequencies. If background frequencies are not provided then a multinomial-Dirichlet model should be used.
Input	either 0 (Dirichlet) or 1 (multinomial)
default	0
restrictions	0 or 1
Example	Useloc: 1 2 3 Uses the first 3 loci in a paternity analysis
Example file	/examples/subsetting/infile
See Also	usesamp

initialfile

Used	Do you want to start the chain from a state that is given in initialfile
Input	0 (no) or 1 (yes)
default	0
Example file	/examples/subsetting/infile
See Also	chains

knownfathers

Used	Do you know which father is associated with offspring ?
Input	0 (no) or 1 (yes)
default	0
Example file	/examples/subsetting/infile
See Also	knownmothers

knownmothers

Used	Do you know which mother is associated with offspring ?
Input	0 (no) or 1 (yes)
default	0
Example file	/examples/subsetting/infile
See Also	knownfathers

malesfile

Used	Do you want to output a file of males genotypes and paternity share
Input	0 (no) or 1 (yes)
default	1
See Also	femalesfile

motherfile

Used	To give the filename for the data file with genotypes of potential known mothers – if you also make the number of mothers constant using motherprior, then this gives the total pool of mothers for the sample. If this is not set then the set of potential mothers are sampled proportional to their probabilitity.
Input	name
default	
restrictions	If set, the file must be present in the directory – or in the path given
Example	freqfile: ../../data/mothers.data reads a datafile off.data from the directory ../../data
Example file	/examples/subsetting/infile
See Also	datafile, freqfile, fatherfile

motherprior

Used	To set the prior for the number of fathers. The list of available priors is set out in the input section.
Input	distribution type with parameters
default	
restrictions	prior should be restriced to positive values - although if this is not set then the actual prior used is conditioned on positive values.
Example	uniform - improper uniform prior
Example file	/examples/subsetting/infile
See Also	motherprior, muprior

motherrange

Used	To set the range of possible mothers
Input	pair of integers
default	1 10 – a range of 1 to 10 inclusive
Example file	/examples/subsetting/infile
See Also	motherprior, motherrange

muprior

Used	The prior for the mutation rate
Input	prior
default	gamma(2,2000)
restrictions	
Examples	muprior: 0.001 muprior: gamma(4,1000)
See Also	alphaprior, betaprior

samples

Used	To give the number of samples to be taken from the posterior chain
Input	integer
default	1000
restrictions	integer > 0
Example	samples: 1000 takes 1000 samples from the posterior chain
See Also	burn-in, thinning

seed

Used	To give the number of samples to be taken from the posterior chain
Input	integer
default	1
restrictions	integer > 0
Example	seed: 1000 takes 1000 samples from the posterior chain
See Also	burn-in, thinning

thinning

Used	To determine the thinning of the chain. If this is set to one, a single Metropolis-Hastings' update or Gibbs' sampling step is attempted per sample from the chain. as the updates are a random scan of the possible steps a thinning value of at least 10 is recommended.
Input	integer
default	100
restrictions	must be greater than or equal to 1
Example	`thinning` Collects output after every 10^{th} Metropolis' or Gibbs' step.
See Also	samples, burn-in

useloc

Used	To determine which of the loci are to be used in the analysis
Input	List of integers
default	Empty (NULL) – use all data
restrictions	The values must lie between 1 and the number of loci, with no repeats
Example	Useloc: 1 2 3 Uses the first 3 loci in a paternity analysis
Example file	/examples/subsetting/infile
See Also	usesamp

usesamp

Used	To determine which of the samples are to be used in a paternity analysis
Input	List of integers
default	Empty (NULL) – use all samples
restrictions	The values must lie between 1 and the number of samples, with no repeats
Example	Useloc: 1 2 3 Uses the first 3 loci in a paternity analysis
Example file	/examples/subsetting/infile
See Also	useloc

Acknowledgements

I would like to thank ….
Bryan Neff, for kindly providing the data for a reanalysis of the problem in his paper.

References

Brooks S (1998) *The Statistician*
Emery A, Wilson I.J. (2000) ….
Green (1995)
Kichler *et al* (1999) *Mol. Ecol*
Neff (2000) *Mol. Ecol*
Wilson & Balding (2000) *Genetics*

Appendix 1: R/S-Plus Code for pre- and post-processing

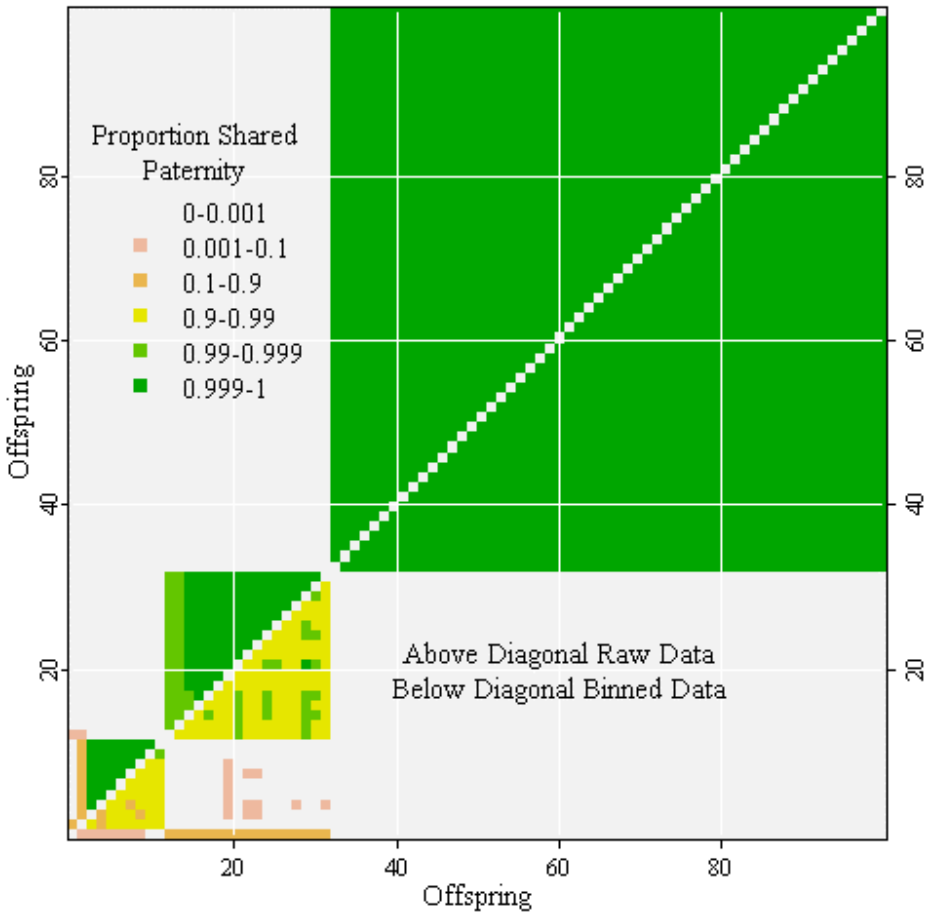


Figure 1: Graphical output from the post-processing functions from R version 1.1 for windows. This figure illustrates the shared paternity for a full dataset (above diagonal) and partial data (below diagonal) for the data given in Emery *et al.* (2000).

The descriptions of output files may have convinced you that the volume of output required to draw inferences from a Bayesian MCMC analysis can be overwhelming. The post-processing of output can be a time consuming process; and without care can be very difficult. The functions within the file *parentage.R* included with the distribution make most of these jobs much more simple. Furthermore, graphical examination of the data can help to guide statistical analyses, and pre-processing functions can make sure that the data output to *Parentage* is of the correct format, and graphical representations of the data can be produced

Functions for pre-processing

Before describing all the pre-processing functions I have included an example R pre-processing session and a post-processing session.

```
> source("parentage.R")  # read the parentage functions”
> string1_readstring("../data/string1.data","../data/breeding.data")
> plot(string1, main=”String 1”)  # plot the data with title String 1
warning
```

```
locus 2 allele 90 is not in background
locus 2 allele 126 is not in background
locus 3 allele 108 is not in background
locus 3 allele 131 is not in background
locus 5 allele 263 is not in background
locus 5 allele 292 is not in background
```

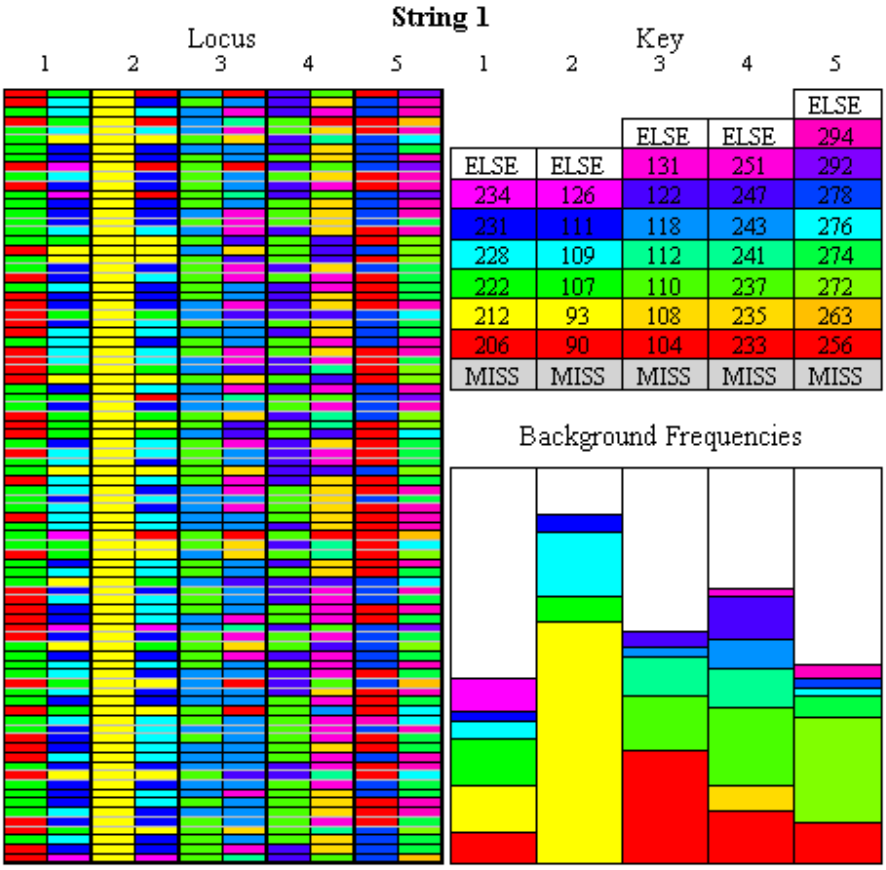


Figure 2: Schematic of String 1 and breeding population information. This figure produced from the pre-processing session in this appendix.

Also included are R-files containing functions to simulate data from particular models. When combined with the *parentage* program these can help to design parentage surveys experiments.