Probability assessments of family relations using the program 'pater'

Petter F. Mostad   Thore Egeland

Norwegian Computing Center
P.O.Box 114, Blindern
N-0314 Oslo, Norway
February 1996

version 2.0

Preface

This document is a manual for **pater**, version 2.0, a program for probability assessments of family relations. The program has been developed, tested and documented by Petter Mostad and Thore Egeland of NCC (Norwegian Computing Center) in close cooperation with Bjørnar Olaisen, Bente Mevåg and Margurethe Stenersen of Institute of Forensic Medicine, Oslo.

The main difference compared to previous versions is the implementation of an algorithm of the Elston-Stewart type, which allows more complex cases to be treated.

Contents

1 Probability calculations for DNA-data 1

1.1 Introduction 1

1.2 Pedigrees 2

1.3 Probability calculations 2

1.4 Odds calculations 3

1.5 Cutsets 4

1.5.1 Exact treatment of cutsets 6

2 Manual for ‘pater’ 8

2.1 Commands 8

2.1.1 Command for case identification 9

2.1.2 Commands for entering the pedigree 10

2.1.3 Commands for entering allele data 11

2.1.4 Commands for computations and files 13

2.1.5 Commands for information and help 15

2.2 File input and output 16

2.2.1 Advanced use of the read command 17

3 Examples 18

3.1 A paternity case 18

3.1.1 The pater report 19

3.2 Mutations 20

3.3 The case of the two sisters 20

3.3.1 Analytical calculations 21

3.3.2 Using pater 22

3.4 The case of the missing father 24

3.5 Testing pater in more complex cases 26

3.5.1 Example based on similar pedigrees 28

References **28**

A The algorithm **30**

Chapter 1

Probability calculations for DNA-data

This is the written documentation accompanying the program `pater`. Most people will probably not read it from start to finish, but instead look up the information they need when they need it. The program should be largely self-explanatory for persons who are familiar with the kind of problems it can solve. Nevertheless, it may be a good idea to read through this first chapter, to learn about `pater`'s way of handling pedigrees and odds calculations. This chapter also explains exactly what `pater` calculates, and what assumptions are made in these calculations.

Chapter 2 gives a detailed account of how to use `pater` in practice, listing and explaining all the possible interactive commands. The use of input from and output to files is also explained. In Chapter 3, a number of examples using `pater` are included. They should be a help when learning to use the program, but they are also included to verify the correctness of the program. Appendix A is included to describe the algorithm employed by the program.

1.1 Introduction

Developments during the last few years have made it increasingly feasible to use measurements of DNA to determine the true family relationships between people. The `pater` program is expected to be used primarily in cases where DNA measurements are available. However, the algorithm and its implementation apply equally well to systems used prior to the breakthrough of DNA in forensic applications. `pater` was developed mainly motivated by identification cases, as opposed to standard paternity cases. To be more specific, our main ambition has been to provide a convenient tool to calculate paternity indices in complex pedigrees. Obviously, `pater` deals with standard cases if the user so desires.

`pater` makes some simplifying assumptions, discussed below, but is quite flexible in the kinds of data it can handle. It tries to answer the following two related questions:

- Given a pedigree, what is the probability of making certain DNA measurements for persons in it?
- Given two alternative pedigrees, what is the odds ratio between them, given certain DNA measurements?

Currently, `pater` bases its results on allele frequencies, pedigrees and a simple mutation model. Later versions are expected to include *kinship*.

The present document is a manual. How it is used depends to a large extent on the user. A new user may choose to skip directly to the examples. Several example files appear on the diskette so that retyping may be avoided. A more experienced user might choose to go into more detail. Methods and theory are generally not covered. The reader may need to read elsewhere. Some references are Essen-Möller (1938), Morris, Sanda & Glassberg (1989), Gjertson, Mickey, Hopfield, Takenouchi & Terasaki (1988), Berry (1991), Evett (1991), Devlin, Risch & Roeder (1992), Mayr & Rossi (1993) Roeder (1994), Balding & Nichols (1996), Enquist (1994), Olaisen, Stenersen & Gedde-Dahl (1994), Stenersen, Hoff-Olsen & Olaisen (1994).

1.2 Pedigrees

Before we can go on, we must discuss how to formalize the concept of a pedigree, or family tree. Genetically speaking, all pedigrees are built up from a single kind of family relation: That between a parent and a child. For example, a family with two parents A and B and two children C and D may be described by saying that A is the parent of C, A is the parent of D, B is the parent of C, and B is the parent of D. In a more complicated example, suppose we want to express that X is the uncle of Y. We can do this by saying that X is the child of X1 and X2, X1 and X2 have the child X3, and X3 has the child Y.

When inputing a pedigree to `pater`, one can only use the parent/child relation. Thus one must add the extra persons to make this possible. To add the uncle X and his niece Y into `pater`, one must also add the persons X1, X1 and X3, and the relations described above. `Pater` users are expected to draw pedigrees routinely.

Note that, although `pater` requires every person to be specified as male or female, this is only for book-keeping purposes, and does not affect the probability calculations. Thus, one does not need to know whether X3 is male or female in reality; either specification will give the same result.

1.3 Probability calculations

We return to the first question mentioned in the introduction. Assume we are given a pedigree and DNA observations of some persons in it. That is, for some allele systems, we assume we have measured the allele pairs of some persons in these systems. We then want to calculate the probability of measuring exactly these allele pairs, compared to any other set of allele pairs we could have measured. We make the assumption that the DNA observations we are given are the true values; thus we exclude the effects of measuring errors, errors in the testing procedure, and so on.

We also make the important assumption that the allele systems are independent, that is, having a particular allele pair in one system has no influence on the probability of having another pair in another system. This justifies multiplying the probabilities of the different systems to arrive at the overall probability.

Each person will have two alleles in each system, one paternal and one maternal allele. `pater` assumes that there is a certain number of possible allele types in each system, and that each type has a certain frequency in the general population. (The program will require as input the frequencies of the allele types that appear in the observations.) With this assumption, there are a finite number of possibilities for specifying the alleles (for a single system) for all the persons in the pedigree. Each such specification, or "allele configuration", has a probability which may be calculated as described below. To find the probability of the given DNA observations, we can simply take the sum of the probabilities of the configurations that are compatible with the observations.

To find the probability of a configuration, one must take the product of the probabilities of each allele, given the remaining relevant alleles. The alleles may be divided into two groups. Those alleles which do not have the relevant parent present in the pedigree are given the same probability as the general population frequency. Thus, that one person in a family has a certain allele does not influence the probability that others in the family have the same allele (unless one is a direct descendant of the other). So the effects of kinship are not considered in this version of the program. If an allele has the relevant parent present in the pedigree, its probability is a function of the alleles of that parent. There is a 50% chance of inheriting either of the alleles from the parent. Given that a certain allele is inherited, it may have mutated on the way from the parent to the child. The chance that a mutation occurs is given as input to the program; it may vary from system to system. Given that a mutation occurs, it is assumed that there is an equal probability of ending up with any of the other allele types in the system. The total number of allele types in a system is also input into the program. This number must always be at least as large as the actual number of alleles registered in the system. We see that the fact that some allele types are in some sense closer than others is disregarded[1]. Using all these assumptions, one can calculate the probability of interest.

The above gives a nice conceptual description of the probability computation. However, the actual computations are done in a different and more efficient manner, see the Appendix.

1.4 Odds calculations

Calculating the probability of the observations, given the pedigree, is the basic, default action of the program `pater`. However, one is usually interested in a different question: There are two alternative pedigrees, and one wants to

[1]`pater` may well be extended to include a more sophisticated mutation model, as well as kinship.

calculate the odds ratio between them, given the DNA observations. The odds ratio may in fact be computed as the following fraction

$$\frac{\pi(\text{data} \mid \text{one pedigree})}{\pi(\text{data} \mid \text{another pedigree})}.$$

Here $\pi(\text{data} \mid \text{pedigree})$ denotes the probability of the observed data given the pedigree. Thus the program uses the computations of the previous section to compute the numerator and denominator, and then divides the two numbers.

Lets take an ordinary paternity case as example. The most direct approach is to say that there are three people involved: The mother, her child, and a man who may be the father. Then there are two possible pedigrees: Only the mother is a parent, and alternatively both the mother and the man are parents. However, `pater` requires that we think in a slightly different way: We must define four persons: The mother, her child, the father and the man from which DNA data has been taken. The question for which to calculate the odds is then whether the two men are in fact one and the same person.

It may seem unnecessary to force users of the program to think in this way, but it has definite advantages. Consider the following example: A corpse is found, and one wants to determine whether this is in fact the missing father of two sons (who have the same mother). To use `pater` on this problem, one must define a total of 5 persons: The two sons, their mother, their father, and the corpse. The question for which to calculate the odds is then whether the corpse and the father are identical. Assume the program had not forced the user to define a separate father. Then, when calculating the probability that the corpse is not the missing father, one would loose the information that the sons have the *same* father (whether or not that father is the corpse).

It is possible to imagine odds problems that cannot be rephrased into a question of whether two persons are identical. However, such problems are probably quite rare. In these cases, one must calculate the probability for the data given each pedigree separately, and then divide the numbers manually. See Section 2.2.1 for some ideas on how this may be done efficiently.

1.5 Cutsets

Pater uses an algorithm that efficiently discards configurations of genotypes with zero probability. This means that computations in allele systems with a mutation rate of zero will generally be quite fast. However, a positive mutation rate means that all genotype configurations have positive probability. In a system with n allele types appearing in the data and m persons in the pedigree, the computation time will be proportional to $(n+1)^{2m}$, which may be unacceptably long, unless we use *cutsets*.

Using cutsets increases the efficiency of `pater`, whether the mutation rate is zero or not, and whether one is computing odds ratios or probabilities. The results are of course not changed. Choosing the cutsets that give the lowest computation time may be difficult, but luckily it is generally sufficient to find reasonably good cutsets. A bad choice of cutsets can never give longer computation time than using no cutsets at all.

So what is a cutset? It is one or more persons that separate different parts of the pedigree. For example, consider the following case (which, by the way, does not require cutsets for sufficiently efficient computations):

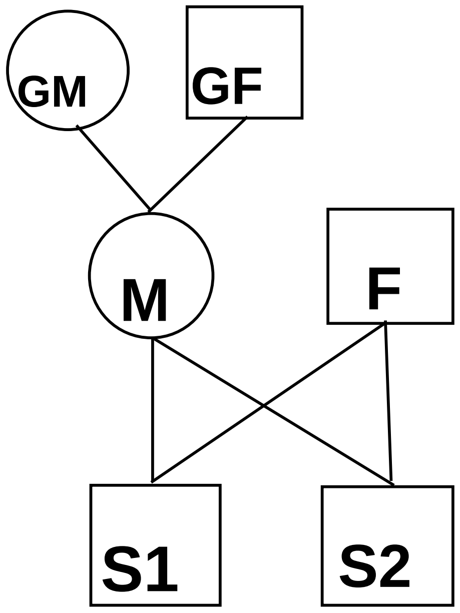

Figure 1.1: Example of a pedigree.

Here, the mother "M" separates her parents "GF" and "GM" from her sons "S1" and "S2" and their father "F". This makes the mother a good cutset. Let us try to be more precise about what it means that the mother for example separates "GM" from "F": We mean that any path in the pedigree from "GM" to "F" must include the mother "M". In a path, we are only allowed to move from parent to child, or from child to parent. As we are only using the parent/child relation, we have drawn Figure 1.1 using this relation instead of the standard way.

In the above example, "M" separates the pedigree into two "components": One consisting of the grandparents, and one consisting of the sons and their father. We can go on to subdivide the components by using further cutsets. The grandparents cannot really be separated any further, but "M" may be separated from "F" by the cutset consisting of "S1" and "S2". Note that a cutset consisting of only one of the sons does not separate "M" from "F", because then there exists a path between them via their other son. So in our example a possible list of cutsets is one consisting of the mother, and one consisting of her two sons. Cutsets are input into the program in the execute command (see Section 2.1.4). So we would write

```
execute M; S1, S2
```

Let us look at a more complex example involving an odds ratio computation. (This example does require cutsets, or at least something beyond the algorithm of `pater` 1.0). Consider the pedigree of Figure 1.2. The file `uncle.txt` provided with the installation diskette contains an example with this pedigree.

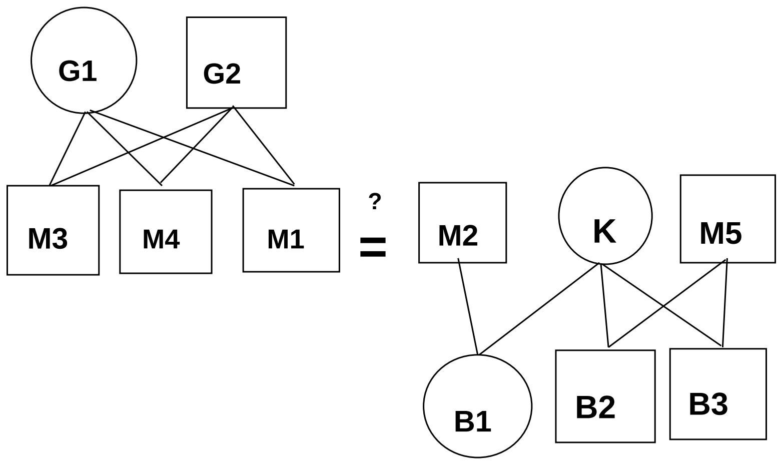

Figure 1.2: Example of a pedigree.

Again we use a non-standard way of drawing the pedigree to emphasize that only parent/child relations matter to us. When specifying cutsets for an odds ratio computation, we should primarily be concerned with the case where the two "odds persons" are identified into one. A good cutset to start with is often at this "combined" person. In our case the first cutset would be M1 (or M2). The family of M1 may then be separated with the cutset G1, G2. The family of M2 may be separated into pieces by using two cutsets: B1, and then K. The entire execute command could then be

```
execute M1; G1, G2; B1; K
```

The same list of cutsets is used both for the computations where M1 and M2 are identical, and when they are not. Note how this works perfectly well in our case: The cutset M1 is not useful in the non-collapsed case, and is simply discarded by the program. Other choices of cutsets are possible in this example. Remember that the goal is just to find a list of cutsets reducing the computation time to an acceptable level. Note that cutsets with too many persons may make your computer run out of memory.

1.5.1 Exact treatment of cutsets

Below is a more precise and more technical description of how the program reads in and handles cutsets:

Consider first the case when the program computes the probability of the observed data, given the pedigree. Cutsets are checked according to the following requirements:

- No person may appear in two different cutsets.
- A cutset may not contain persons from different components of the pedigree, as defined by the previous cutsets. This means that for any two persons in the cutset, there must be a path in the pedigree from one to

the other not including members of previous cutsets.

A line of output on the screen is produced for each cutset, describing whether it is "OK" or "removed". If a cutset includes all the persons in a component (as defined by the previous cutsets) it cannot help the calculations, and it is discarded.

When an odds ratio is calculated, the program constructs two different sets of cutsets: One for the case when the "odds persons" are different, and one for the case when they are identical. The output to the screen will concern the last of these. The first structure is constructed exactly as above. The second is constructed as above after identifying the persons in the odds command as a single person, with one exception: The identified persons may appear in several cutsets, provided all ut the first of these have only one member. Only the first of these cutsets is kept; the others are discarded.

We should mention that there exist algorithms for automatic generation of cutsets. Later versions of `pater` may include such algorithms.

Chapter 2

Manual for ‘pater’

The program `pater` is run interactively by typing in commands. Each line entered is a command (with the exception of the `information` command). The commands may be used to enter data into the program (see Sections 2.1.2 and 2.1.3), and to produce output (see Section 2.1.4). One may also enter data into the program by means of text files (see Sections 2.1.4 and 2.2).

Individual data items may be entered, and also removed, in any order. At any point, the output is based on the current contents of data. One may always use the command `list` to see what the current data structure is (see Section 2.1.5).

The data may be divided into data about family relations (see Section 2.1.2) and data about alleles (see Section 2.1.3). Each person, each system of alleles, and each allele inside a system must be given a name by the user. These names can be real names, dummy names like X, Y, or Z, or even numbers 1,2,3,.... The only requirement is that names do not contain the following characters:

? = , | ;

Note that names may in fact contain spaces; “John Jones” is a valid name.

2.1 Commands

Below is a full list of the possible commands in the program. The special command words may in fact be abbreviated to any length, including one letter: For example, one may write `male Ole` or indeed `m Ole`, instead of `males Ole`. The exception is the command `clean`, which must always be written in full. In the subsections below, the underlining signifies the possible abbreviation of the command.

It is not necessary to memorize all the commands; they can always be listed with the command `help`:

Example:

```
pater> help
The following commands are available:
information  Specify general information
females      Specify names of females
males        Specify names of males
parents      Specify family relations
children     Specify family relations
odds         Specify the odds question
system       Add an allele system
alleles      Add alleles to an allele system
data         Add DNA measurements
execute      Compute the odds or probabilities
write        Write data to file
read         Read data from file
quit         End the program
clean        Remove all data
list         List contents of data structure
help         Use "help COMMANDNAME" for info on each command
pater>
```

2.1.1 Command for case identification

Background information about the computation is entered using the command

`information`

When this command is entered, the program will respond with

`pater information>`

The user may then enter a number of lines with information about this particular application, e.g., case title, case number, case description, program user and so on. (Note that the current date is automatically added when `pater` writes data to files, and need not be added by the user). The input is terminated by two blank lines. The information entered is printed out when using `write` and `list` commands. It may be removed with the command `- information`. This must be done before new `information` can be read from file (see 2.1.4) or entered with a new `information` command.

Example:

```
pater> i
Input case information. End with two blank lines:
pater information> **** The infamous royalty case ****
pater information> Case number 666-1421
pater information> User of program: John Doe
pater information>
pater information>
New information added.
pater> list
**** The infamous royalty case ****
Case number 666-1421
User of program: John Doe
pater> - i
Information removed.
pater> list
pater>
```

It is not necessary to use the `information` command to do computations.

2.1.2 Commands for entering the pedigree

See Section 1.2 for general information on how pedigrees are stored by the program. All persons must be defined with one of the commands below before they can be used in family relations or DNA data commands:

males < name1 >, < name2 >, < name3 > ...

females < name1 >, < name2 >, < name3 > ...

The word `males` may be followed by zero or more names to represent males, separated by commas. Similarly for females. Remember that names may contain spaces, so "John Paul" is a single name, and not two. Persons may be removed by using the same commands as above, preceded by "-".

After persons have been defined, one may add family relations between them with the commands[1]

parents < parent1 > (, < parent2 >) ; < child1 >, < child2 >, < child3 > ...

children < child1 >, < child2 >, < child3 > ...; < parent1 > (, < parent2 >)

In either of the two commands above, one may list one or two parents, and any number of children. The data is added to any previous family relations for these persons. Each combination between parent and child is added separately, and relations that cannot be added result in error messages instead. For example, a person cannot have two fathers or mothers, and a person cannot be his or her own ancestor. Preceding the commands above with "-" remove the relations instead of adding them.

One may use the `odds` command below to instruct the program to compute the odds ratio between two different pedigrees (see Section 1.4):

odds < name1 > = < name2 >

After this command is entered, a computation (with the `execute` command) will compute the odds that "name1" is in fact identical to "name2". If no `odds` command is entered, the program will instead compute the probability of the observed data given the family structure. The odds question may be removed by the command

- odds

Example:

[1]Strictly speaking, one of the commands `parents` and `children` would be sufficient. Both are, however, included for user convenience.

```
pater> m John Doe, father
The male "John Doe" added to the data.
The male "father" added to the data.
pater> f mother, daughter
The female "mother" added to the data.
The female "daughter" added to the data.
pater> p father, mother; daughter
Added to data: "father" is the father of "daughter".
Added to data: "mother" is the mother of "daughter".
pater> o John Doe = father
The odds question "John Doe" = "father" has been added to the data.
pater> list
******************************************************************************
FAMILY STRUCTURE
******************************************************************************

females  mother, daughter
males    John Doe, father

Family relations:
+--------------------+------------+
|  parents           |  children  |
+--------------------+------------+
|  mother,  father   |  daughter  |
+--------------------+------------+

Question for odds computation: is John Doe = father?
pater> - m father
Removing the male "father" from the data.
pater> list
******************************************************************************
FAMILY STRUCTURE
******************************************************************************

females  mother, daughter
males    John Doe

Family relations:
+-----------+------------+
|  parents  |  children  |
+-----------+------------+
|  mother   |  daughter  |
+-----------+------------+
pater>
```

2.1.3 Commands for entering allele data

Each allele belongs to an allele system. Each allele system must be defined with the following command before it can be used:

system < systemname > ; < mutationrate > < number of possible alleles >

This command adds a new allele system to the data. Its mutation rate must be given; it may of course be 0 to exclude mutations. One may also give the number of “possible alleles” (see Section 1.3). If no argument is given, the default number is assumed to be 100. The number of alleles specified with the `allele` command below must be less than or equal to the given number of “possible alleles”. The mutation rate and the number of possible alleles in a system may be changed by using the `system` command again. An allele system may be removed with the command

- system < systemname >

As soon as an allele system has been defined, we may define alleles in it with the command

allele < sys.n. > ; < all.name1 >, < freq.1 > ; < all.name2 >, < freq.2 > ...

We may have one or more terms, separated by semicolons, with an allele name and its general population frequency (see Section 1.3). The information is added to that already present. Alleles may be removed with a command

- allele < systemname > ; < all.name1 > ; < all.name2 > ...

Alleles that have been defined with the `allele` command may be used in the command below, where the actual DNA observations of different persons are input:

data < sys.n. > ; < pers.1 >, < a1 >, < a2 > ; < pers.2 >, < a3 >, < a4 > ...

We may have one or more terms, separated by semicolons, of persons and alleles. In each term, a person is specified, together with his or her observed alleles in this system. (If only one allele type is observed for a person, the allele must be listed twice). We may remove DNA observations from the data for a given system and for a number of persons with the command

- data < sys.n > ; < pers.1 > ; < pers.2 > ...

Example:

```
pater> m person1, person2, person3
The male "person1" added to the data.
The male "person2" added to the data.
The male "person3" added to the data.
pater> s sys1; 0
The allele system "sys1" with mutation rate 0
and 100 possible alleles added to the data.
pater> a sys1; allele1, 0.1; allele2, 0.2; allele3, 0.2
The allele "allele1" with probability 0.1 added to the system "sys1".
The allele "allele2" with probability 0.2 added to the system "sys1".
The allele "allele3" with probability 0.2 added to the system "sys1".
pater> d sys1; person1, allele1, allele1; person2, allele1, allele2
Alleles "allele1" and "allele1" for "person1" has been added to the data.
Alleles "allele1" and "allele2" for "person2" has been added to the data.
pater> s sys2; 0.02 50
The allele system "sys2" with mutation rate 0.02
and 50 possible alleles added to the data.
pater> list
*******************************************************************************
FAMILY STRUCTURE
*******************************************************************************

females
males    person1, person2, person3

No family relations registered.
*******************************************************************************
ALLELE SYSTEM sys1
*******************************************************************************
```

```
Mutation probability: 0

General population frequencies of alleles:
+-----------+-------------+
|  allele   |  frequency  |
+-----------+-------------+
|  allele1  |  0.1        |
|  allele2  |  0.2        |
|  allele3  |  0.2        |
+-----------+-------------+

Observed alleles in this system:
+-----------+---------------------+
|  person   |  observed alleles   |
+-----------+---------------------+
|  person1  |  allele1,  allele1  |
|  person2  |  allele1,  allele2  |
+-----------+---------------------+
*******************************************************************************
ALLELE SYSTEM sys2
*******************************************************************************

Mutation probability: 0.02, number of possible alleles: 50

No alleles registered.

No observations of alleles registered.
pater>
```

2.1.4 Commands for computations and files

To execute probability computations, use the command

execute

If an odds question has been entered, its odds ratio is computed. Otherwise, the probability of the observed data, given the family structure, is computed. The above computations may take a long time to complete, especially if some of the allele systems have non-zero mutation rates. To speed the calculations, one may use cutsets in the execute command, see Section 1.5. As described there, members of each cutset are separated by commas, while different cutsets are separated by semicolons:

execute < cut1-p1 >, ..., < cut1-pn >; < cut2-p1 >, ..., < cut2-pm > ...

We may write all or parts of the current data structure to a file with the command

write < filename > (< options >)

The filename should be a name which is recognized by your computer as a text file. It may be a good idea for it to end with ".pater", ".p" or ".txt", for example. If no options are given, all data is written to the file. To write parts of the data, use the same options as with the `list` command. The file produced will have the same format as the listing produced by the corresponding `list` command, except that it will contain a heading, which among other things will contain the date when the file was made. See Section 3.1.1 for an example.

To read data back in again, use the command

read < filename >

Data is read from the file "filename" and *added* to the current data. Thus, for example, information about a single allele system may be saved on a file and used in several cases. The `read` command can read any output produced by the `write` command, and can in addition read interactive commands (see Section 2.2.1).

To remove all current data from the data structure, use the command

clean

Note that, for safety reasons, `clean` must always be written with all five letters.

To quit the program, use

quit

If data has been changed after the last `write` statement, the user will be asked if she really wants to quit. This is to avoid that data will be lost by mistake.

Example:

```
pater> read oldcase.pater
pater> sys system 1; 0.03 50
The system system 1 now has mutation rate 0.03
and 50  possible alleles.
pater> e
******************************************************************************
RESULTS
******************************************************************************

Results for each system:
+----------------+----------+
|  allele system  |  result   |
+----------------+----------+
|  system 1       |  4.97333  |
|  system 2       |  2.5      |
|  system 3       |  16.6667  |
+----------------+----------+

The total odds that man = father: 207.222
pater> w report.pater
pater> clean
All information removed.
pater> read uncle.txt
pater> execute M1; G1, G2; B1; K
Cutset OK: M2
Cutset OK: G1, G2
Cutset OK: B1
Cutset OK: K
Finished computations for system g3.
Finished computations for system ms31.
Finished computations for system ynh.
Finished computations for system ms43.
Finished computations for system cmm.
Finished computations for system ms1.
Finished computations for system b67.
Finished computations for system ms205.
*******************************************************************************
RESULTS
*******************************************************************************

Results for each system:
+----------------+------------+
|  allele system  |  result     |
+----------------+------------+
|  g3             |  0.266095   |
|  ms31           |  0.246615   |
|  ynh            |  0.486544   |
|  ms43           |  0.488778   |
|  cmm            |  0.254383   |
|  ms1            |  0.0553425  |
|  b67            |  0.24968    |
|  ms205          |  0.0218408  |
+----------------+------------+

The total odds that M2 = M1: 1.19809e-06
pater> q
```

2.1.5 Commands for information and help

To get information about the current data structure, use

list (< options >)

Without any options, all data will be listed on the screen. However, a number of options are available so that parts of the data can easily be focused on:

list family

will list only the pedigree.

list males

will list only the males.

list females

will list only the females.

list systems

will list all data about allele systems.

list system < systemname >

will list all data concerning the allele system “systemname”.

list frequencies

will list all frequency data of all allele systems.

list frequencies < systemname >

will list all frequency data in the allele system “systemname”.

list information

will list the information added with the `information` command.

There are two forms of the general help command:

help

and

?

They will both result in a listing of all available commands in the program. To get information about a specific command, use

help < command >

or

< command >?

2.2 File input and output

The commands `write` and `read` should be very useful, and may be used in many different ways. First of all, the `write` command can be used to make a report of the probability computations in a specific case. It should be possible to read and understand this output without any knowledge of the program `pater` itself. Secondly, one may use the `write` command to save a particular data structure before terminating the program. One can then continue to work on it by reading it in again later. Thirdly, one may avoid typing many of the interactive commands by instead editing on a report from a similar case, and then reading in the changed data.

The `read` function has been made fairly flexible, so that many ways of editing the report should be understood by the program. One can change, add, and remove names of persons, allele systems and alleles, and change the numbers that appear. The forms appearing in the report may also be edited: Lines may be added or removed, as long as the '|' character is used similarly to the other lines. The area just below the heading will contain the `information` added to the program, and may also be edited. The file will be read from top to bottom, and all lines that are not understood result in error messages. If adding the data as described in the file is illegal for some reason, that will also result in error messages.

It may also be quite useful to save parts of a data structure in a file, and read it in from that file later, when needed in another computation. For example, an allele system may have a long list of possible alleles and corresponding general population frequencies. When that has been read into the program, it may be saved to a file with a command

write savefile.pater frequencies < systemname >

(recall that the `write` command has the same options as the `list` command).

When the same system is used in other computations, one may just write

read savefile.pater

to enter the data, instead of typing it in all over again.

In the files that are read by the `read` command, one may want to add lines with comments and explanations, in ordinary language. This can be done by starting the lines with any of the characters

% # \ / $

Such lines are ignored by the read function, and do not result in error messages.

2.2.1 Advanced use of the read command

In addition to reading edited reports as described above, the `read` command may be used to read files containing interactive commands as those described in Section 2.1. All interactive commands (except the `information` command) may appear in such files, and are interpreted exactly as if they had been written directly on the terminal. If a sequence of commands will be used frequently, it should be written in a file named for example "commands.pater". One can then just write `read commands.pater` to execute these commands.

Assume, for example, that one wants to calculate the odds ratio between two pedigrees that are fairly different. The commands that must be used to transform one pedigree into the other could be collected in a file "trans.pater", while the commands to return to the first tree is collected in "back.pater". One may then easily go back and forth between the two cases using `read trans.pater` and `read back.pater` to compute odds ratios with different DNA data.

Note that the interactive commands may appear inside a report file written by the `write` command. Thus, to modify the data in a report, on can add interactive commands to it, instead of editing its contents. However, if the commands are added inside the "information section" of the file, i.e., just below the heading, they simply become part of the `information` text. If the commands are added in the "RESULTS" section, they will be ignored. And interactive commands may not appear inside the tables listed in the file.

As an example, including the command `- info` at the top of a report file will ensure that any `information` included in the report (e.g. case number and such) will replace the `information` already present in the data structure. If one instead adds the `clean` command at the top of the report, all previous data will be removed from the program before the new data is added. Thus the data in the file will *replace* the old data. As a third example, using the command `quit` in the file will cause the program to stop reading from the file at that point.

Note also that even the `read` command may appear in a file read by the `read` command. Thus one may have nested command files, to any depth. One can imagine this to be useful for example in the following case: Information about several allele systems is spread around on several files. To collect it together in one file, on need only write a file with a list of `read` commands reading the other files.

Chapter 3

Examples

We conclude this manual with a number of examples. They have two purposes: Documenting how to use `pater` in different situations, and verifying that the program functions correctly.

3.1 A paternity case

Consider one system in a paternity case where we have data for mother (M), son (S) and alleged father (AF) and disregard mutations. Assuming the alleged father is in fact the true father, the probability of the data may be written (using standard notation; some explanation is offered shortly)

$$P(M, S, AF \mid \text{father}) = P(S \mid M, AF)P(M)P(AF) \tag{3.1}$$

whereas if there is no family relation between AF and S the probability becomes

$$P(M, S, AF \mid \text{random man}) = P(S \mid M)P(M)P(AF). \tag{3.2}$$

Equations (3.1) and (3.2) combine to give

$$\text{odds father} = \frac{P(S \mid M, AF)}{P(S \mid M)} \tag{3.3}$$

The numerator is the *conditional probability for the son's allele measurements given those of his father and mother.* In the denominator only data for the mother is given. Some algebra is required to show that

$$P(S \mid M) = \frac{1}{2}\left(P(A) + P(B)\right).$$

The odds *may or may not depend on both alleles originating from AF.* If AF is $\{A, B\}, \{A, A\}$ or $\{B, B\}$, the odds becomes

$$\frac{1}{P(A) + P(B)}$$

whereas if AF is $\{A, z\}$ or $\{B, z\}$, where z differs from A and B, the odds equals

$$\frac{1}{2(P(A) + P(B))}.$$

This is in agreement with results from `pater`. In this case some information on both alleles of AF is required; but the exact z frequency is irrelevant. Section 3.4 provides a case where the frequency of all alleles matter.

3.1.1 The pater report

A pater report (also available from the file `patcas.txt`) follows describing one case based on the discussion of the previous section.

```
*******************************************************************************
DNA PROBABILITY COMPUTATIONS REPORT
Written by the program pater, version 2.0
Date: Wed Mar  6  1996
*******************************************************************************
*******************************************************************************
RESULTS
*******************************************************************************

Results for each system:
+----------------+-----------+
|  allele system  |  result   |
+----------------+-----------+
|  S1             |  13.3333  |
+----------------+-----------+

The total odds that a.father = father: 13.3333
*******************************************************************************
FAMILY STRUCTURE
*******************************************************************************

females  mother
males    son, a.father, father

Family relations:
+------------------+------------+
|  parents          |  children  |
+------------------+------------+
|  mother,  father  |  son       |
+------------------+------------+

Question for odds computation: is a.father = father?
*******************************************************************************
ALLELE SYSTEM S1
*******************************************************************************

Mutation probability: 0

General population frequencies of alleles:
+----------+-------------+
|  allele  |  frequency  |
+----------+-------------+
|  a       |  0.025      |
|  b       |  0.05       |
|  c       |  0.05       |
+----------+-------------+

Observed alleles in this system:
+------------+--------------------+
|  person    |  observed alleles  |
+------------+--------------------+
|  mother    |  a,  b             |
|  son       |  a,  b             |
|  a.father  |  a,  b             |
+------------+--------------------+

The odds that a.father = father: 13.3333
******************************************************************************
```

3.2 Mutations

Lets look at a very simple case where we can test out the mutation features of `pater`. We assume there is an alleged father with alleles A and B in some system, and a son with alleles C,C, see Figure 3.1.

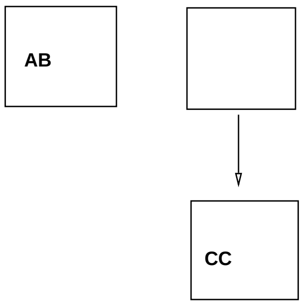

Figure 3.1: Illustration of model for mutation.

Obviously, with a mutation rate of 0, the odds that the alleged father is the real father is 0. With a positive mutation rate M however, we get a more complex computation. If the alleged father *is* the real father, then we get a probability of observing the given data of

$$P(A)P(B)P(C)M\frac{1}{n-1},$$

where n is the total number of possible alleles in the system. If he is *not*, then `pater` computes the probability of the data by considering whether the allele inherited by the child from the real father was C or non-C. The probability of the data becomes

$$P(A)P(B)P(C)\left[P(C)(1-M)+(1-P(C))M\frac{1}{n-1}\right].$$

The odds now becomes

$$\frac{M\frac{1}{n-1}}{P(C)(1-M)+(1-P(C))M\frac{1}{n-1}}=\frac{M}{P(C)(1-M)(n-1)+(1-P(C))M}.$$

If we look at an example where $P(A)=0.1$, $P(B)=0.2$, $P(C)=0.3$, $M=0.02$ and $n=50$, then we get odds of 0.00138696. This result is also obtained with `pater`; see the example file `mut.txt` provided on the diskette.

3.3 The case of the two sisters

The possible family structures are shown Figure 3.2. We would like to determine the odds for two sisters having a common father. We do this first analytically and then using `pater`. Towards the end we consider the odds for the fathers of the girls being brothers vs. the girls having the same father.

3.3.1 Analytical calculations

Mutations are disregarded throughout this example.

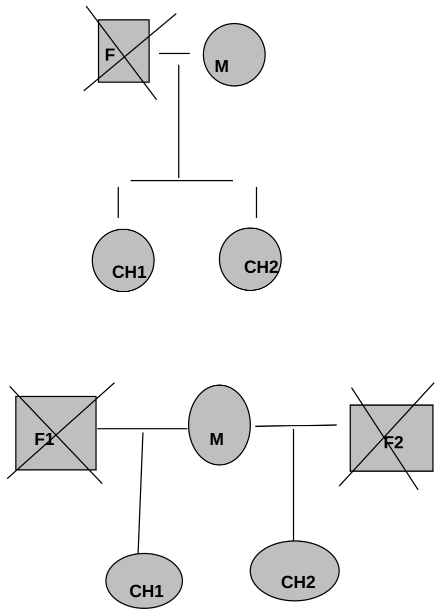

Figure 3.2: The objective is to calculate Essen–Möllers to determine which of two family structures is the most likely.

For 2 of the systems, referred to as S1 and S2, the picture is as follows:

M	CH1	CH2
(A1,A2)	(A2,A3)	(A2,A3) (or (A1,A3))

For simplicity we assume that the frequency of the paternal allele, $P(A_3)$, coincides for systems S1 and S2. The odds for these loci may be shown to equal

$$\frac{\frac{1}{2}(1+P(A_3))}{P(A_3)}. \tag{3.4}$$

For the remaining three systems, called S3, S4 and S5, we have

M	CH1	CH2
(B2,B3)	(B3,B4)	(B1,B3).

For these loci the odds become $\frac{1}{2}$ independently of allele frequencies.

The odds in favor of the girls being full sisters equal

$$\frac{1}{32}\left(\frac{(1+P(A_3))}{P(A_3)}\right)^2. \tag{3.5}$$

As $P(A_3)$ increases from 0 to $\frac{\sqrt{32}+1}{31} \approx 0.215$ the odds decrease from *infinity* to 1. Moreover, as $P(A_3)$ decreases from 0.215 to 1 the odds decrease from 1 to $\frac{1}{8}$. The odds in favor of the girls being sisters is depicted for $P(A_3)$ ranging from 0.02 to 0.1 in Figure 3.3.

The odds for the fathers of the girls being brothers (see Figure 3.4) vs. identical equal, (assuming all allele frequencies to be 0.05) 1.0. *Intuitively*, one would expect the last odds to be smaller than 1 provided $P(A_3) < 0.05$ and greater than 1 whenever $P(A_3) > 0.05$. The above numbers as well as the previous calculations, may be checked by considering the **pater** output provided in the next section.

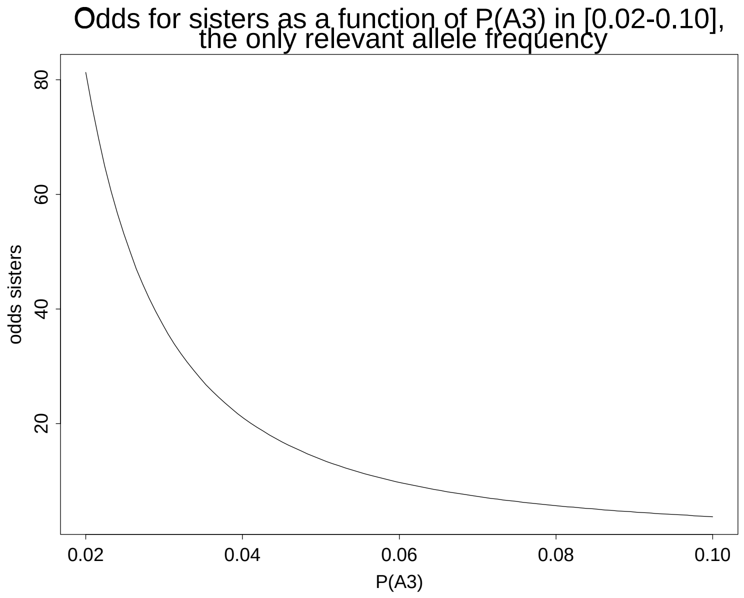

Figure 3.3: The odds in favor of the girls being sisters is shown for $P(A_3)$ ranging from 0.02 to 0.1.

3.3.2 Using pater

Consider first the problem summarized by Figure 3.2. Applying **pater** to the problem we need to calculate the probabilities for the two pedigrees and then manually divide the numbers. It is convenient to define the last pedigree by modifying the first. Some time may also be solved by editing files rather than doing the input manually. The **pater** files describing all required input for the pedigrees in Figure 3.2 are **sis.txt** (same father) and **hafsis.txt**; all allele frequencies are assumed to equal 0.05. Assuming the existence of these files (which the user should be able to load from the diskette), the **pater** session runs:

```
pater> read sis.txt
pater> exec
*******************************************************************************
RESULTS
*******************************************************************************
```

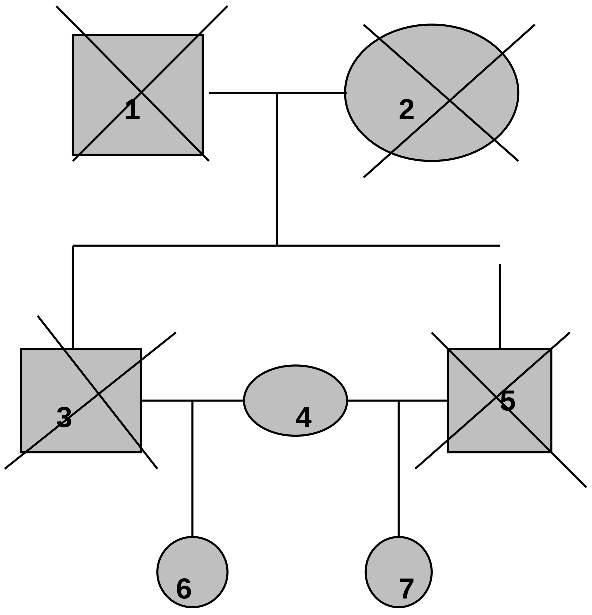

Figure 3.4: The pedigree if the fathers of the girls are brothers.

```
Results for each system:
+-----------------+---------------+
|  allele system  |  result       |
+-----------------+---------------+
|  S1             |  3.28125e-05  |
|  S2             |  3.28125e-05  |
|  S3             |  1.5625e-06   |
|  S4             |  1.5625e-06   |
|  S5             |  1.5625e-06   |
+-----------------+---------------+

The total probability of the data given the family structure: 4.10713e-27
pater> clean
All information removed.
pater> read hafsis.txt
pater> exec
******************************************************************************
RESULTS
******************************************************************************

Results for each system:
+-----------------+-------------+
|  allele system  |  result     |
+-----------------+-------------+
|  S1             |  3.125e-06  |
|  S2             |  3.125e-06  |
|  S3             |  3.125e-06  |
|  S4             |  3.125e-06  |
|  S5             |  3.125e-06  |
+-----------------+-------------+

The total probability of the data given the family structure: 2.98023e-28
```

The odds in favor of the girls having the same father equals

$$\frac{4.10713e-27}{2.98023e-28} = 13.8$$

coinciding with the analytical result provided Equation (3.5):

$$\frac{1}{32}\left(\frac{(1+0.05)}{0.05}\right)^2 = 13.8.$$

Investigating whether the father of the girls are brothers is approached similarly.

3.4 The case of the missing father

Recall the following example: A corpse is found, and one wants to determine whether this is in fact the missing father of two brothers. The pedigree is shown in Figure 3.5.

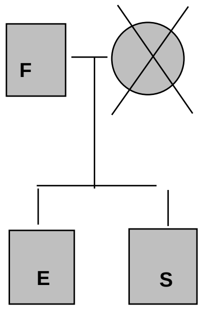

Figure 3.5: The pedigree shows that data are available for two brothers and their possible father.

The data for all systems follows:

```
*******************************************************************************
ALLELE SYSTEM hla
*******************************************************************************

Mutation probability: 0

General population frequencies of alleles:
+----------+------------+
|  allele  |  frequency |
+----------+------------+
|  a1      |  0.125     |
|  b1      |  0.237     |
+----------+------------+

Observed alleles in this system:
+----------+-------------------+
|  person  |  observed alleles |
+----------+-------------------+
|  F       |  a1,  b1          |
|  S       |  a1,  b1          |
|  E       |  a1,  b1          |
+----------+--------------------+
******************************************************************************
ALLELE SYSTEM humf
******************************************************************************

Mutation probability: 0

General population frequencies of alleles:
+----------+-------------+
|  allele  |  frequency  |
+----------+-------------+
|  a2      |  0.298      |
|  b2      |  0.197      |
+----------+-------------+

Observed alleles in this system:
+----------+--------------------+
|  person  |  observed alleles  |
+----------+--------------------+
|  E       |  a2,  a2           |
|  S       |  a2,  b2           |
|  F       |  a2,  b2           |
+----------+--------------------+
******************************************************************************
ALLELE SYSTEM humact
******************************************************************************

Mutation probability: 0

General population frequencies of alleles:
+----------+-------------+
|  allele  |  frequency  |
+----------+-------------+
|  a3      |  0.063      |
|  b3      |  0.01       |
|  c3      |  0.072      |
+----------+-------------+

Observed alleles in this system:
+----------+--------------------+
|  person  |  observed alleles  |
+----------+--------------------+
|  E       |  a3,  c3           |
|  S       |  b3,  c3           |
|  F       |  a3,  b3           |
+----------+--------------------+
******************************************************************************
```

In this case, it is possible to derive analytically the odds in favor of F being the father of E and S; i.e., that the pedigree shown in Figure 3.5 is the correct. The odds for the three systems may be written adhering to the noation of the above `pater` output (details are provided in Egeland & Mostad (1995):

$$\frac{(P(a1)+P(b1))\,(1+P(a1)+P(b1))}{4P(a1)P(b1)(1+P(a1)+P(b1)+2P(a1)P(b1))} = 2.9275 \qquad (HLADQA1)$$

$$\frac{1+P(a2)+P(b2)}{4P(a2)P(b2)[(1+P(a2))]} = 4.905. \qquad (HUMFES)$$

$$\frac{1+P(c3)}{4P(a3)P(b3)[1+2P(c3)]} = 371.85. \qquad (HUMACTBP2)$$

The results agree with those obtained from `pater`:

```
pater> exec
*******************************************************************************
RESULTS
*******************************************************************************

Results for each system:
+----------------+-----------+
|  allele system  |  result   |
+----------------+-----------+
|  hla            |  2.9275   |
|  humf           |  4.90483  |
|  humact         |  371.85   |
+----------------+-----------+
The total odds that F = alternativ.F: 5339.36
```

The reader may want to experiment by for instance varying input parameters. The complete output file from `pater`, `pcr1.txt` may be loaded from the installation diskette and so there is no need for retyping.

3.5 Testing pater in more complex cases

In the examples above, we have found that `pater` gives the same results as we get from analytical computations. But the examples were fairly simple, and we would like to test `pater` in some more complex cases also. The problem is that then, analytical computations often become intractable, which is precisely why `pater` was invented in the first place.

There are however several classes of examples where analytical computations are easy and these may be used to check the results of pater. For example, one may test pater on pedigrees where the persons with observed data are in fact unrelated, but have common children. In another set of examples, we compare the computations of `pater` for pedigrees that are similar in such a way that we can prove that the correct results are identical. Let us digress to develop some theoretical results. Below, we use the same assumptions as are used in `pater`; in particular, we disregard the effects of kinship.

Lemma 1:

Assume we are given a pedigree with a person A with no parents in the tree, and a person B having A as a parent. Then making the assumption that B inherits A's *maternal* (or *paternal*) allele does not change the probability of a given data set.

Proof:

This may be regarded as obvious, but let us nonetheless include a short proof: We have

$$\begin{aligned} P(\text{data}) &= \frac{1}{2}P(\text{data} \mid B \text{ inherits } A\text{'s maternal allele}) + \\ &\quad \frac{1}{2}P(\text{data} \mid B \text{ inherits } A\text{'s paternal allele}). \end{aligned}$$

But

$$\begin{aligned}
& P(\text{data} \mid B \text{ inherits } A\text{'s maternal allele}) \\
= \;& P(\text{data} \mid B \text{ inherits } A\text{'s paternal allele}),
\end{aligned}$$

because A has no parents in the tree, and neither its children nor any data measurements can separate between its maternal and paternal alleles. The lemma follows.

Lemma 2:

Assume we are given a pedigree with persons A and B, where A has no parents in the tree, and B has only A as a parent. Assume also we have a mutation rate of 0. Then the probability of a given data set is the same as with the family tree where B is instead the parent of A.

Proof:

The probability of a given data set may be written as the sum of the probabilities of all the different allele "constellations" compatible with the family relations and with the data. By an allele constellation, we mean an assignment of allele types to the maternal and paternal alleles of all the persons in the pedigree. Lets write m_A, p_A, m_B and p_B for the maternal and paternal alleles of A and B, respectively, in such a constellation. We may assume, for simplicity, that both A and B are female. By Lemma 1, we may assume that B inherits the maternal allele of A. Writing $I(m_A = m_B)$ for the function which is 1 when $m_A = m_B$, and 0 otherwise, we get

$$\begin{aligned}
P(\text{constellation}) &= P(m_A)P(p_A)I(m_A = m_B)P(p_B) \\
&\quad \times P(\text{rest of constellation} \mid m_A, p_A, m_B, p_B) \\
&= I(m_A = m_B)P(p_A)P(m_B)P(p_B) \\
&\quad \times P(\text{rest of constellation} \mid m_A, p_A, m_B, p_B)
\end{aligned}$$

Here, when we write $P(m_A)$ for example, we simply mean the general population frequency of the allele m_A. But the last line is exactly the probability of the constellation in the situation where B is the mother of A and we assume A inherits B's maternal allele. By Lemma 1, we may remove the assumption that A inherits B's maternal allele. Summing over all constellations compatible with the family relations and the data, we get that the probability of the given data set is independent of whether A is the parent of B, or vice versa.

Definition:

We say that two pedigrees are *similar* if one can be transformed into the other by repeatedly interchanging who is parent and who is child for pairs of persons who have no other parents in the tree.

Proposition

The probability for a given data set is equal for similar pedigrees, when disregarding mutations.

Proof:

This follows from Lemma2.

One can now construct a number of pairwise similar complex pedigrees. Testing with `pater`, we find that we get the same results for similar trees, as expected (under the assumption that the mutation rate is 0).

3.5.1 Example based on similar pedigrees

Consider the pedigree shown in Figure 3.6. V.1 is (a,b); the corpse, assumed to be II.2, is (a,c). By arguments similar to the ones developed above, the body found is equally likely to be I.1 as II.2 provided mutations are disregarded. This is confirmed by the `pater` algorithm.

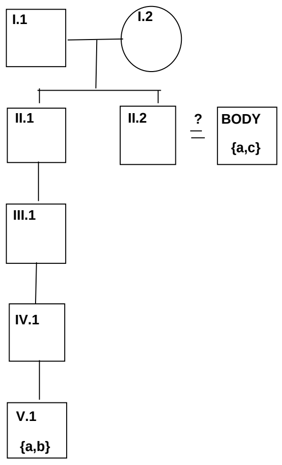

Figure 3.6: A body is found, suspected to be II.2. Only data from V.1 is available.

Performance In the above case there is a total of $14^4 = 38416$ constellations. An odds calculation like the above takes 20 seconds on a dec 5000 station. If mutations are included all configurations contribute and the execution time increases to 1180 seconds when cutsets are not used. (Using cutsets reduces the computation time dramatically). As a rough estimate, `pater` only considers 1.7% of the total number of constellations. If data is available also for person II.1, III.1 and IV.1, say (a,d), (a,e) and (a,f), respectively, the total number of constellations equals $14^7 = 105 * 10^6$. However, `pater` calculates the odds instantaneously in this case since more constellations may be avoided.

Bibliography

Balding, D. J. & Nichols, R. A. (1996), 'A method for quantifying differentiation between populations at multi–allelic loci and its implications for investigating identity and paternity', *To appear in Genetica.*

Berry, D. A. (1991), 'Inference using DNA profiling in forensic identification and paternity cases', *Statistical Science* **6**, 175–205.

Devlin, B., Risch, N. & Roeder, K. (1992), 'Forensic Inference From DNA Fingerprints', *JASA* pp. 337–349.

Egeland, T. & Mostad, P. (1995), ' Essen–Möller's indeks og PCR'. Manuskript.

Enquist, E. (1994), 'Statistisk utvärdering av DNA-analyser i samband med faderskapsundersökningar'. Manuskript.

Essen-Möller, E. (1938), 'Die Beweiskraft der Ähnlichkeit im Vaterschaftsnachweis.Theoretische Grundlagen', *Mitt. Anthropol. Ges (Wien)* **68**, 9–53.

Evett, I. (1991), Interpretation: a personal odyssey, *in* C. G. G. Aitken & D. A. Stoney, eds, 'The Use of Statistics in Forensic Science', Ellis Horwood Ltd., Chichester, pp. 9–22.

Gjertson, D. W., Mickey, M. R., Hopfield, J., Takenouchi, T. & Terasaki, P. I. (1988), 'Calculation of probability of paternity using DNA sequences', *Am. J. Hum. Genet* **43**, 860–869.

Mayr, S. M. & Rossi, U. (1993), 'Basic methods and criteria for paternity investigation'. Education course following the 15th international Congress of ISFH. Contributors include (in order of appearance) P. J. Lincoln, W.R. Mayr, G. Geserick, G. G. De Lange, K. M. Sullivan, A. Walton, C. Kimpton, G. Tully, P. Gill, V. L. Pascali, S. Rand, M. P. Baur, A. Piazza, U. Rossi.

Morris, J., Sanda, A. & Glassberg, J. (1989), 'Biostatistical Evaluation of Evidence from Continuous Allele Frequency Distribution Deoxyribonucleid Acid (DNA) Probes in Reference to Disputed Paternity and Identity', *Journal of Forensic Sciences* pp. 1311–1317.

Olaisen, B., Stenersen, M. & Gedde-Dahl, T. (1994), 'Bioassay of kinship using VNTR alleles'.

Roeder, K. (1994), 'DNA Fingerprinting: A review of the Controversy', *Statistical Science* pp. 222–278.

Stenersen, M., Hoff-Olsen, P. & Olaisen, B. (1994), Paternity investigation – experiences using single locus probes, *in* B. Olaisen & B. Teige, eds, 'XII Nordiske Møte i Rettsmedisin'.

Appendix A

The algorithm

Consider a pedigree consisting of persons $X_1, \ldots, X_n$, and focus on a single allele system. Let $t_1, \ldots, t_k$ be a list of the alleles in this system appearing as observations from the pedigree, and let t_0 denote all alleles different from these. Specifying all alleles for all persons in the pedigree, we obtain what we will refer to as a constellation. Disregarding for a moment family relations, we realize that there is a total of $(2n)^{k+1}$ constellations: Each of the $2n$ alleles $a_{11}, a_{12}, \ldots, a_{n1}, a_{n2}$ (a_{i1} denotes the paternal allele of person i while a_{i2} is the maternal) may be of either of $k+1$ types $t_0, t_1, \ldots, t_k$. Assuming knowledge of allele frequencies of t_i for $i = 1, \ldots, k$, the frequency of the rest allele becomes $P(t_0) = 1 - P(t_1) - \ldots - P(t_k)$. Including family relations but disregarding mutations (the modification required to take care of mutations is attended to in the next subsection), we may compute the probability $P(a_{11}, a_{12}, a_{21}, a_{22}, \ldots, a_{n1}, a_{n2})$ of a configuration by writing

$$\begin{aligned}
& P(a_{11}, a_{12}, a_{21}, a_{22}, \ldots, a_{n1}, a_{n2}) \\
= \; & P(a_{11}) \cdot P(a_{12}) \qquad\qquad (A.1) \\
& \cdot P(a_{21} \mid a_{11}, a_{12}) \cdot P(a_{22} \mid a_{11}, a_{12}) \\
& \vdots \\
& \cdot P(a_{n1} \mid a_{11}, a_{12}, \ldots, a_{n-1,1}, a_{n-1,2}) \cdot P(a_{n2} \mid a_{11}, a_{12}, \ldots, a_{n-1,1}, a_{n-1,2}).
\end{aligned}$$

If we order $X_1, \ldots, X_n$ chronologically implying that parents of X_i have indices smaller than i, the conditional probabilities are easy to compute. The conditional probability of an a_{ij} whose relevant parent is not among $X_1, \ldots, X_n$ coincides with the general allele population frequency $P(a_{ij})$. If, however, X_k is the appropriate parent (then we know that $k < i$) then

$$P(a_{ij} \mid a_{11}, a_{12}, \ldots, a_{i-1,1}, a_{i-1,2}) = \begin{cases} 1 & \text{if } a_{ij} = a_{k1} \text{ and } a_{ij} = a_{k2}, \\ \frac{1}{2} & \text{if } a_{ij} = a_{k1} \text{ and } a_{ij} \neq a_{k2}, \\ \frac{1}{2} & \text{if } a_{ij} \neq a_{k1} \text{ and } a_{ij} = a_{k2}, \\ 0 & \text{if } a_{ij} \neq a_{k1} \text{ and } a_{ij} \neq a_{k2}. \end{cases}$$

We may now compute the probability of the data given the pedigree by summing all constellations compatible with data:

$$P(\text{data} \mid \text{pedigree}) = \sum_{\substack{\text{constellations} \\ \text{compatible with data}}} P(a_{11}, a_{12}, a_{21}, a_{22}, \ldots, a_{n1}, a_{n2}).$$

The outlined approach id extremely inefficient and so more efficient algorithms are called for. In particular, we would like to detect and discard zero probability constellations as early as possible. Intuitively, it is also reasonable to attend to persons assigned observations as early as possible. We may do so by rearranging the factors appearing in (A.1). Define functions

$$p(a_{11}, a_{12}, \ldots, a_{i-1,1}, a_{i-1,2}, a_{i1}, a_{i2}) = f \cdot b_1 \cdot b_2 \cdot \ldots \cdot b_s, \tag{A.2}$$

where

$$f = \begin{cases} 1 & \text{if } X_i \text{ has both parents among } X_1, \ldots, X_n, \\ P(a_{i1}) & \text{if only mother is among } X_1, \ldots, X_n, \\ P(a_{i2}) & \text{if only father is among } X_1, \ldots, X_n, \\ P(a_{i1})P(a_{i2}) & \text{otherwise.} \end{cases} \tag{A.3}$$

There is one factor b_j for each parent/child pair appearing in the pedigree, where either the parent or the child is X_i, and where the related person X_k has index $k < i$. We set

$$b_j = \begin{cases} 1 & \text{if both parent alleles are identical} \\ & \text{to the allele the child has inherited from this parent,} \\ \frac{1}{2} & \text{if one parent allele coincides with the child's,} \\ 0 & \text{otherwise.} \end{cases} \tag{A.4}$$

The definitions of f and b_j in (A.3) and (A.4) are illustrated in Figure A.1.

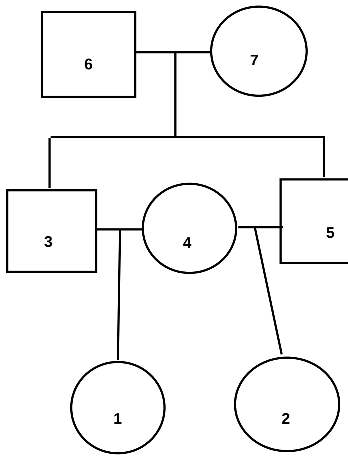

Figure A.1: Assume the persons are indexed as shown in the figure, and assume $a_{11} = A1$, $a_{12} = A2$, $a_{21} = A3$, $a_{22} = A4$, $a_{31} = A1$, $a_{32} = A3$. Then $p(a_{11}, a_{12}, a_{21}, a_{22}, a_{31}, a_{32}) = 1 \cdot \frac{1}{2} \cdot \frac{1}{2}$.

It is now possible to show that the product in the equation below contains exactly the same factors as the product in (A.1), and so

$$\begin{aligned} & P(a_{11}, a_{12}, a_{21}, a_{22}, \ldots, a_{n1}, a_{n2}) \\ = \; & p(a_{11}, a_{12}) \\ & p(a_{11}, a_{12}, a_{21}, a_{22}) \\ & \vdots \\ & p(a_{11}, a_{12}, \ldots, a_{n1}, a_{n2}) \end{aligned}$$

Sorting $X_1, \ldots, X_n$ to achieve that persons with measured alleles are attended to first, we get

$$\begin{aligned}
& P(\text{data} \mid \text{pedigree}) \\
= & \sum_{\substack{\text{constellations} \\ \text{complying with data}}} P(a_{11}, a_{12}, \ldots, a_{n1}, a_{n2}) \\
= & \sum_{\substack{\text{constellations} \\ \text{complying with data}}} p(a_{11}, a_{12}) \cdot \ldots \cdot p(\ldots, a_{n1}, a_{n2}) \\
= & \sum_{\substack{(a_{11}, a_{12}) \\ \text{complying with data}}} p(a_{11}, a_{12}) \\
& \cdot \left[\sum_{\substack{(a_{21}, a_{22}) \\ \text{complying with data}}} p(a_{11}, a_{12}, a_{21}, a_{22}) \right. \\
& \left. \cdot \left[\sum_{\substack{(a_{31}, a_{32}) \\ \text{complying with data}}} p(a_{11}, a_{12}, a_{21}, a_{22}, a_{31}, a_{32}) \cdot [\ldots] \right] \right].
\end{aligned}$$

Rephrased as above, the algorithm is well suited for *recursive* implementation. At each level, one computes allele combinations (a_{i1}, a_{i2}) complying with data, and such that $p(a_{11}, a_{12}, \ldots a_{i1}, a_{i2}) > 0$. Pairs (a_{j1}, a_{j2}) with $j > i$ are only checked for such cases. This is the algorithm employed when no cutsets are input by the user.

Modifications to account for mutations

The **pater** algorithm may be modified to include quite general mutation models by changing the definition of b_j in Equation (A.4). Currently, only a simple mutation model has been implemented. A mutation probability M is specified for each system as well as the total number n of alleles in the system. An allele mutates to any of the other $n-1$ equally probably. The required modification amounts to replacing (A.4) by $b_j = \frac{1}{2}g(a_1, b) + \frac{1}{2}g(a_2, b)$ where a_1 and a_2 are the alleles of the parent and b is the relevant allele of the child, k the number of specified alleles, and

$$g(a, b) = \begin{cases} 1 - M & \text{if } a \neq t_0, b = a, \\ \frac{M}{n-1} & \text{if } a \neq t_0, b \neq a, \\ 1 - M\frac{k}{n-1} & \text{if } a = t_0, b = a, \\ \frac{M}{n-1}(n-k) & \text{if } a = t_0, b \neq a. \end{cases}$$

Cutsets

The algorithm outlined above works quite well as long as the mutation rate is 0. For nonzero mutation rates, all configurations have nonzero probabilities, and the above approach to efficient calculations breaks down. If we look at Equation A.1 again, we see that many of the conditional probabilities will in fact be independent of most of the previous alleles. The dependence will only be on those to which a person is either a parent or a child. The trick with cutsets is to determine a small set of persons (the cutset) such that all subsequent persons are only dependent on those. Then the entire computation for the subsequent persons may be done only once, and the result stored in a table as a function of the genotype of the cutset. This greatly cuts down on the size of the computations, but may require large tables to hold the intermediate results.