fam2r package tutorial
Version 1.2

Thore Egeland and Magnus Dehli Vigeland

March 29, 2017

1 Introduction

Familias is a program for probability calculations when inferring paternity and identification based on DNA data. In recent years, the windows version (the core version of the program is also available as an R package with the same name) of the program has been extended in several directions including functionality for DVI (Disaster Victim Identification) problems. Some problems are, however, most easily solved in R as there is so much available R–code. For this reason Familias produces files containing projects which can be loaded into R for further analysis. This document is a brief tutorial for the fam2r package which is designed to do plotting, simulation, calculations (likelihood ratio (LR), exclusion probabilities and more) based on Familias projects. If you would like to try the examples below, you should load the package:

```
> library(fam2r)
```

2 Basics

2.1 Data

Several datasets are provided within the fam2r package. Three examples are

- grandmother: There is one marker with alleles 1, 2 and 3. A grand mother (GM) is genotyped and we simulate the genotype of the grand son (GS).
- symmetric: There are two markers and three pedigrees, 'halsib', 'avuncular' and 'grandparent'. These pedigrees cannot be distinguished with the standard assumptions (indepndent markers, no mutations or artefacts).
- F21: There are 24 markers and five genotyped individuals.
- E004: There are 24 markers. Two individuals are genotyped for 15 of these markers.

The last two datasets are based on real cases (but changes have been made so that individuals cannot be identified) from the 'Missing grand children' (MGC) project, see Kling et al. (2017). The first data set is constructed to be as small as possible without being completely trivial. Let's have a look at some data

```
> data(grandmother)
> pedigrees = grandmother$pedigrees
> datamatrix = grandmother$datamatrix
> loci = grandmother$loci
```

If you have exported data from Familias to a file, named say, grandmother.R, the lines above can be replaced by pasting or sourcing this file into R. You can look at the data by typing

```
> grandmother
```

Note that the names of persons are available as

```
> rownames(datamatrix)
```

```
[1] "GM"  "FAT" "GS"
```

2.2 Problem Formulation

In all data sets there is a missing person (MP). For the first data set, the grand son is missing. The problem is to determine whether POI is indeed the missing person in the family. In other words, we consider the hypotheses H_1: "POI = MP" and H_2: "POI is an unrelated person". Several questions can be asked prior to genotyping POI. The main one is, losely formulated: "Will we be able to solve the case?". In practice this factors into two more specific questions: "Will we be able to exclude POI if he/she is in fact unrelated?" and "Will we be able to conclude that MP=POI if this is in fact?" As the reader will recognize these questions are related to power in the setting of classical hypothesis testing. To be able to formulate the problem more precisely and also provide precise answers, this package provides functionality, or examples from other packages, for

- plotting,
- calculation of exclusion probabilities,
- simulation, conditionally on genotyped individuals,
- LR calculations.

2.3 Likelihood ratios

In most applications there will only be two hypotheses and we first describe this situation which allows for simplified notation. The likelihood ratio is defined as $LR = Pr(data \mid H_1)/Pr(data \mid H_2)$. We will simulate genotype data, typically for the POI conditionally on the hypotheses and genotyped individuals. Based on the simulated likeihood ratios, we can make plots and calculate summary statistics like the median of the simulated values. The simulations depend on the hypotheses and this has to be reflected in the notation. We write $LR(H_1)$ for the random variable obtained by assuming H_1 is true, and similarly for $LR(H_2)$.

In general, however, there can be hypotheses $H_1, H_2, \ldots, H_n$. The user defines one of these, say number r, to be the reference, and then likelihood ratios $LR_{i,r} = Pr(data \mid H_i)/Pr(data \mid H_r)$ can be calculated for $i = 1, \ldots, n$. When we simulate from H_s, we get *realisations* of the random variable $LR_{i,r}(H_s)$.

2.4 From `Familias` to `linkdat`

We will use the `paramlink` package for plotting and certain computations. `paramlink` represents pedigree data in so-called `linkdat` objects, which differs from the way `Familias` does it. However, `paramlink` provides a simple conversion utility called `Familias2linkdat`. For several functions this transformation is hidden for the end user, but not always. For instance, prior to plotting we transform data by running

```
> x1 = Familias2linkdat(pedigrees, datamatrix, loci)
```

x1 is now a list of `linkdat` objects, the first is

```
> x1[[1]]
```

```
  ID FID MID SEX AFF  L1
1  1   0   0   2   1 1/1
2  2   4   1   1   1 -/-
3  3   2   5   1   1 -/-
4  4   0   0   1   1 -/-
5  5   0   0   2   1 -/-
```

For readers familliar with linkage software, this is recognised as the standard way of representating a pedigree and genotype data. By typing

```
> help(linkdat)
```

you will obtain explanation. There is one feature of 'paramlink' not present in simillar software, `singletons`, i.e., a special `linkdat` object whose pedigree contains 1 individual. For this data set, there are three individuals and parent–child relationships for the second pedigree. Therefore

```
> x1[[2]]

[[1]]
  ID FID MID SEX AFF  L1
1  1   0   0   2   1 1/1

[[2]]
  ID FID MID SEX AFF  L1
1  2   0   0   1   1 -/-

[[3]]
  ID FID MID SEX AFF  L1
1  3   0   0   1   1 -/-
```

contains three singletons.

The command below produces Figure 1 specifically designed for the MGC project

```
> missing.person.plot(x1[[1]], missing=3, marker = 1, dev.width=5,
+                     dev.height=3, fmar=0.03)
```

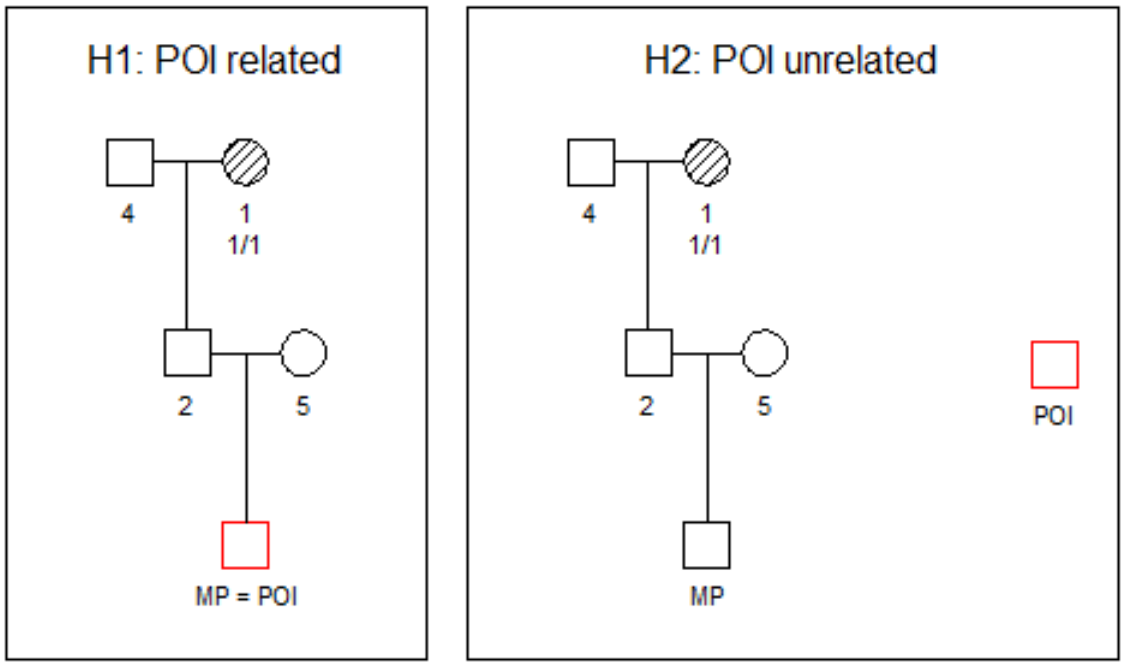

Figure 1: Basic plot for `grandmother` data

There are also more general plot functions in `paramlink` like (resulting plot not shown):

```
> plotPedList(x1, available="shaded", marker = 1, dev.width=5, dev.height=3.3)
```

Some effort is made by the `plotPedList` function above to guess a reasonable window size and margins, but in general the user must be prepared to do manual resizing of the plot window and change to `newdev=FALSE` for the final version or fix the size as above using the parameters `dev.width` and `dev.height`.

3 Exclusion probabilities

Assume the POI is unrelated to the reference family. In some cases it will be possible to exclude POI. This is possible if mutations are disregarded and sufficient information of the genotype of a parent of MP is known from relatives. For instance, if one sibling has genotype 1/2 and another has 3/4, then any genotype involving other alleles than 1, 2, 3, 4 is impossible for MP. We first consider a simple case where the probability of exclusion is $PE = (1 - p_1)^2$:

```
> datamatrix[2,] = c(1,1)
> x = Familias2linkdat(pedigrees, datamatrix, loci)
> PE = exclusionPower(ped_claim=x[[1]], ped_true=x[[2]], ids=3,
+                     markerindex=1, plot=FALSE)
> p1 = loci[[1]]$alleles[1]
> stopifnot(PE==(1-p1)^2)
```

Next follows an example based on a real case:

```
> data(F21)
> pedigrees = F21$pedigrees
> datamatrix = F21$datamatrix
> loci = F21$loci
> x2 = Familias2linkdat(pedigrees, datamatrix, loci)
```

```
> missing.person.plot(x2[[1]], missing=9, marker=c(1,2,5), fmar=0.03,
+   newdev=TRUE,  dev.width=5, dev.height=3.3, cex=0.8, id.labels="num")
```

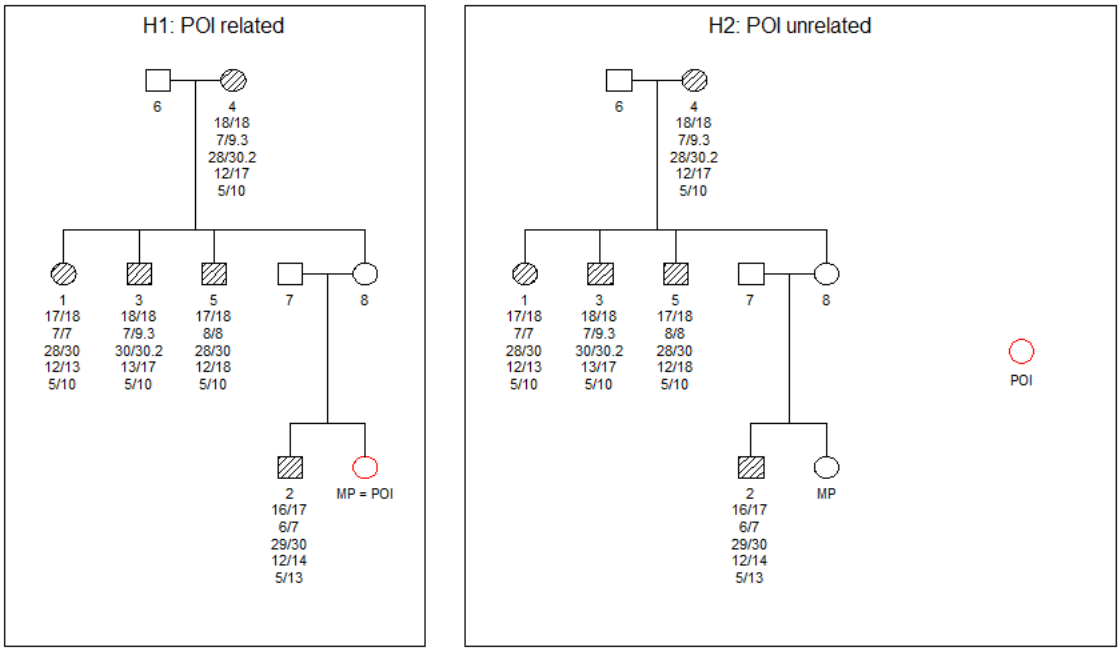

Figure 2: Markers 1,2 and 5 shown for the F21 data.

For marker one in Figure 2, the mother '8' must be 17/18. The allele frequencies are

```
> loci[[1]]$alleles[c("17","18")]

       17        18
0.1609171 0.1170281
```

Therefore the exclusion probability is $(1 - p_{17} - p_{18})^2 = 0.52$. For the second marker, mutations must be accounted for as we assume a mutation in transition from '4' to '5'. The exclusion probability is thus 0. For the third marker shown (which is the fifth of 24 markers), exclusion is never possible. The exclusion probability for the first marker can also be calculated using a `paramlink` function:

```
> x2 = Familias2linkdat(pedigrees, datamatrix, loci)
> PE1 = exclusionPower(ped_claim=x2[[1]], ped_true=x2[[2]], ids=9, markerindex=1, plot=FALSE)
> PE1

[1] 0.5213632
```

We obtain the result from all 24 markers by typing

```
> PE.all = sapply(1:24, function(i) exclusionPower(ped_claim=x2[[1]],
+                                ped_true=x2[[2]], ids=9, markerindex=i, plot=FALSE))
```

We can study the exclusion probabilities for the markers by typing

```
> names(PE.all) = lapply(loci, function(x) x$locusname)
> PE.all

   D3S1358       TH01     D21S11     D18S51    PENTA E     D5S818    D13S317     D7S820
0.52136317 0.00000000 0.16988273 0.47672909 0.00000000 0.00000000 0.00000000 0.00000000
   D16S539     CSF1PO    PENTA D        VWA    D8S1179       TPOX        FGA    D19S433
0.00000000 0.08965897 0.18889470 0.00000000 0.19612518 0.19250700 0.41761726 0.20519893
   D1S1656    D12S391    D2S1338    D6S1043   D22S1045     D2S441       SE33   D10S1248
0.00000000 0.28118534 0.50620741 0.00000000 0.03481956 0.21436900 0.00000000 0.23619600
```

and find the overall exclusion probability as follows:

```
> 1-prod(1-PE.all)

[1] 0.9905176
```

A wrapper function $PE$is available to do the above calculations for all markers and combined and also write results to a file:

```
> PE(pedigrees, datamatrix, loci, claim = 1, true = 2,
+   available = 9, file = NULL)

     marker         PE
1   D3S1358 0.52136317
2      TH01 0.00000000
3    D21S11 0.16988273
4    D18S51 0.47672909
5   PENTA E 0.00000000
6    D5S818 0.00000000
7   D13S317 0.00000000
8    D7S820 0.00000000
9   D16S539 0.00000000
10   CSF1PO 0.08965897
11  PENTA D 0.18889470
```

```
12      VWA 0.00000000
13  D8S1179 0.19612518
14     TPOX 0.19250700
15      FGA 0.41761726
16  D19S433 0.20519893
17  D1S1656 0.00000000
18  D12S391 0.28118534
19  D2S1338 0.50620741
20  D6S1043 0.00000000
21 D22S1045 0.03481956
22   D2S441 0.21436900
23     SE33 0.00000000
24 D10S1248 0.23619600
25 Combined 0.99051760
```

4 Simulation

There are several programs that can perform simulation of marker data on pedigrees including `Familias`. However, although algorithms for conditional simulations are quite old, it is hard to find implementations suitable for forensic data. With the `markerSim` function of `paramlink` this is now possible. A simple example follows (inbred alternatives and X-chromosomal markers are also handled):

```
> data(E004) # E zero zero four
> pedigrees = E004$pedigrees
> datamatrix = E004$datamatrix
> loci = E004$loci
> x3 = Familias2linkdat(pedigrees, datamatrix, loci)
> ped1 = x3[[2]][[1]]
> sim1 = markerSim(ped1, N=2, available=7, partialmarker=1, verbose=FALSE)
```

Plots of the data and a simulation the grand daughter can be made as follows

```
> plotPedList(list(ped1, sim1), marker=1, id.labels="num", available = "shaded", newdev=FALSE)
```

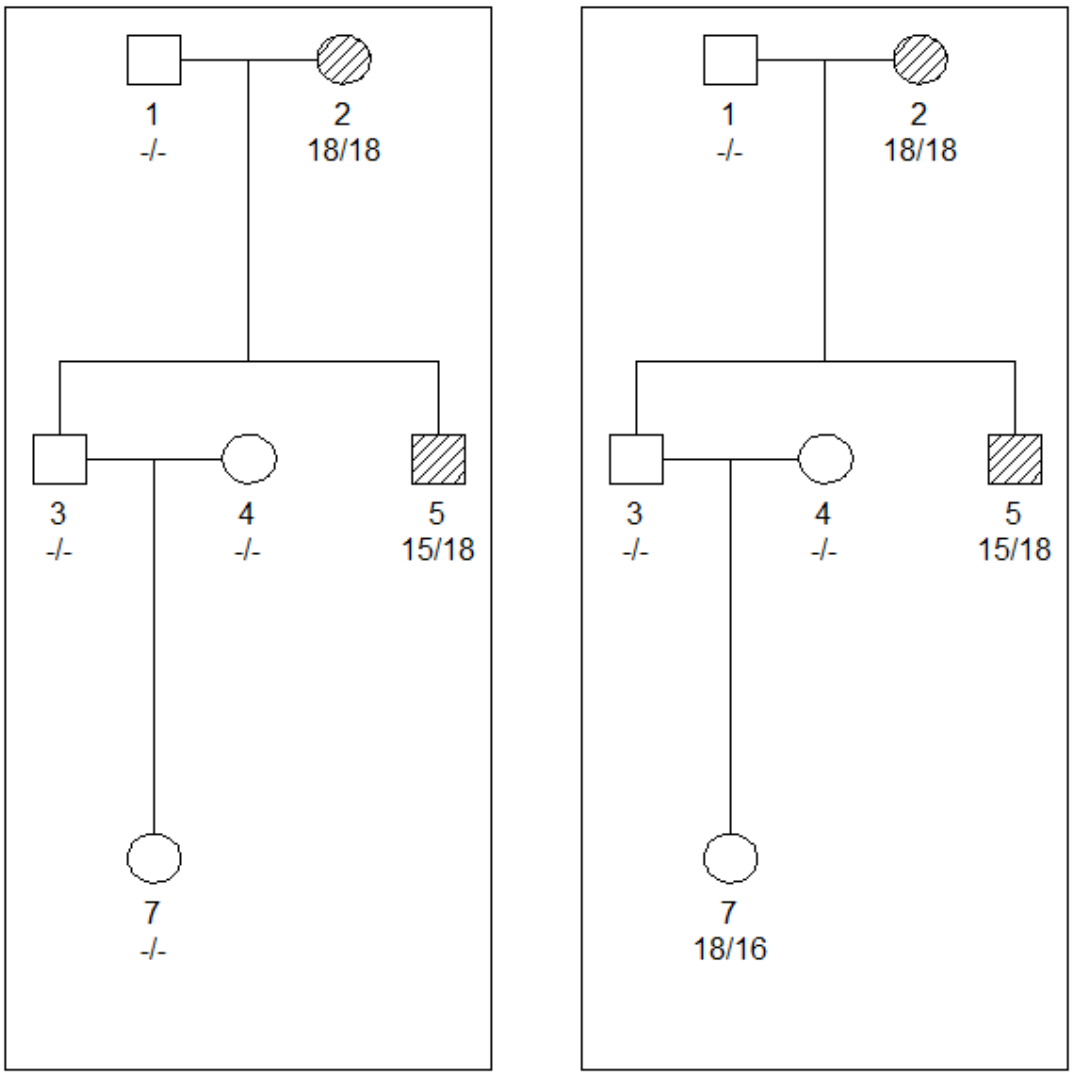

Figure 3: `E004` data. One simulation of first marker in the right panel for the grand daughter

5 Distribution of LR-s

The next topic is to simulate likelihood ratios and we first consider the standard, simplest case, with two hypotheses as those described in general terms in Section 2.3. We use the case descibed by Figure 2 to illustrate.

```
> data(F21)
> pedigrees = F21$pedigrees
> datamatrix = F21$datamatrix
> loci = F21$loci
> Nsim = 100
> res1 = conditionalLR(Nsim = Nsim, datamatrix,  loci, pedigrees, file=NULL,
+                      available = "Missing Person", seed=177, verbose = FALSE, simplify=TRUE)
```

The function `conditionalLR` is in `fam2r` and uses `markerSim`. Above 100 simulations are performed, normally we would like to do 1000 simulations. Genotype data for the `"Missing Person"` are simulated; recall that names can be obtained by typing

```
> rownames(datamatrix)
```

```
[1] "8.TIA MATERNA"         "9.HERMANO"             "10.TIO MATERNO"
[4] "4.ABUELA MATERNA"      "7.TIO MATERNO"         "3.ABUELO MATERNO "
[7] "5 PADRE DESAPARECIDO" "6.MADRE DESAPARECIDA" "Missing Person"
```

From this we see that we could alternatively use `available=9` above. The option `simplify=TRUE` is well defined when there are only two hypotheses (otherwise, as below, we must specify the numerator and denominator of LR), and leads to the following five first lines of output

```
> head(res1)
```

```
            LR.H1 LR.H2
[1,] 9.709296e+14     0
[2,] 3.377939e+11     0
[3,] 7.558242e+06     0
[4,] 6.764328e+07     0
[5,] 1.785500e+10     0
[6,] 2.503624e+10     0
```

The first column are likelihood ratios simulated assuming H_1 to be true, i.e., realisations of $LR(H_1)$. The values are large indicating that, if H_1 is indeed true, we will be able to provide strong evidence. The second column are simulations from H_2. In this case the missing person is simulated as an unrelated person. Chances are small that his genotype data will be consistent with other genotypes for all markers and therefore the likelihood ratio will be 0 most of the time. In fact the exclusion probability, calculated exactly in Section 3, can be estimated as

```
> length(res1[,2][res1[,2] == 0])/Nsim
```

```
[1] 0.97
```

We can summarise the distribution of the likelihoods ratio by plotting or by calculating summary statistics, for instance

```
>   apply(res1, 2, function(x) quantile(x, probs=c(0,0.05,0.5,0.95,1)))
```

```
            LR.H1        LR.H2
0%   8.223829e+04 0.0000000000
5%   8.695068e+06 0.0000000000
50%  9.652809e+12 0.0000000000
95%  1.546995e+18 0.0000000000
100% 7.066319e+19 0.0001551602
```

Here's another example:

```
> data(E004)
> pedigrees = E004$pedigrees
> datamatrix = E004$datamatrix
> loci = E004$loci
> res1 = conditionalLR(Nsim = Nsim, datamatrix,  loci, pedigrees, ref=1,
+                      available = "Missing person", seed=173, verbose = FALSE, simplify=TRUE)
> length(res1[,2][res1[,2]==0])/Nsim

[1] 0
```

It is not possibile to exclude a random person from being the missing person, as is estimated above and can be seen from Figure 3.

Next consider the case with more than two hypotheses. We let $LR_{i,r}(H_s)$ denote a random variable determined by hypothesis s as explained previously. We can estimate the the probability distribution $LR_{i,r}(H_s)$ by simulation as exemplified below: (plot not shown)

```
> data(symmetric)
> pedigrees = symmetric$pedigrees
> datamatrix = symmetric$datamatrix
> datamatrix[2,] = NA
> loci = symmetric$loci
> x4 = Familias2linkdat(pedigrees, datamatrix, loci)
```

Some effort may be needed for nice plots, some attempts follow:

```
> plotPedList(x4, newdev =TRUE, marker=1:2, cex=0.8,
+   available="shaded", dev.width=12, dev.height=3)
> plotPedList(list(x4[[1]][[1]],x4[[2]], x4[[3]][[1]]),  marker=1:3, cex=0.8,
+   available="shaded", dev.width=12, dev.height=3, skip.empty.genotypes = TRUE,
+   frametitles =c("H1: HS", "H2: aunt", "H3: grandparent" ))
```

With standard assumptions, including markers being unrelated, these three pedigrees cannot be distinguished. The proportional mutational model has been used and therefore H_2 : "Avuncular" can be distinguished from the the two others *in theory*, not inp practice, and we use this hypothesis as the reference below. The output is explained as a part of the output since `verbose=TRUE`:

```
> res1 = conditionalLR(Nsim = 5, datamatrix,  loci, pedigrees, ref=2, truePeds=1:3,
+                      available = "B", seed=17, verbose = TRUE, simplify=FALSE)

LR[,,i] is the likelihood ratio conditioned on pedigree i
LR[,,i] is a matrix with one row for each simulation and one column
for each pedigree. The denominator of the LR is pedigree no  2
```

6 Inconsistencies

There may be several reasons for inconsistencies (also called Mendelian errors or incompatibilities). Below we demonstrate how such problems can be detected and located using the `paramlink` function `mendelianCheck`

```
> data(F21)
> pedigrees = F21$pedigrees
> datamatrix = F21$datamatrix
> loci = F21$loci
> x2 = Familias2linkdat(pedigrees, datamatrix, loci)
> mendelianCheck(x2[[1]])
```

```
### Checking autosomal markers ###
Individual 5 incompatible with parents for 1 markers: 2
[1] 2

> x2[[1]]$plot.labels[5]

[1] "7.TIO MATERNO"
```

The output above shows that there is one inconsistency. It occurs in marker 2 for individual 5, i.e., the person listed as the fifth one in `Familias`. The last line above extracts the name of this person.

7 Predicting genotype in presence of mutations

In several applications it may be of interest to predict the genotype of persons (not yet) genotyped. Figure 4 shows one such example (thanks to Daniel Corach for the data and the problem formulation). The output of the code below shows the basic assumptions and that the 'Father' is estimated to be 18.3/20 with probability 0.5559 and 18.3/21 with probability 0.4416. Obviously these estimates depend crucially on the chosen mutation model. Here, for simplicity, the same model is used for females and males. The socalled 'Extended stepwise model' explained on p. 169 of Egeland, Kling and Mostad (2016) is implemented with parameters 0.005 ('Rate'), 0.1 ('Range') and 0.000001 ('Rate 2'). The last parameter controls the mutation probabilities between integer alleles, like 20 and and non-integer alleles like 18.3. As this parameter is small and much smaller than 'Rate', governing mutations not changing between integers and non-integers, it becomes likely that 'Father' has the allele 18.3.

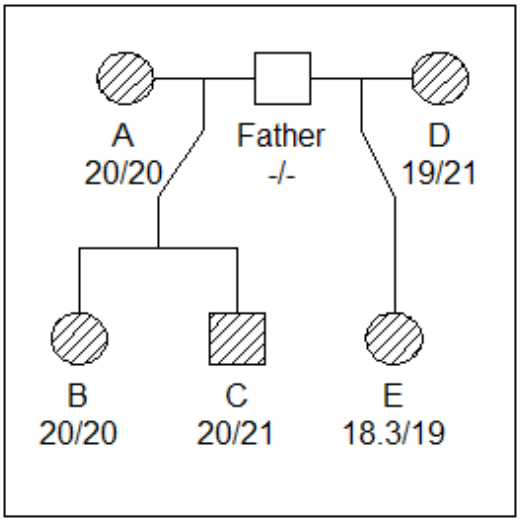

Figure 4: For simplicity we assume all alleles of the marker appear in the figure. As mutations are modelled, there are then ten possible genotypes for 'Father'.

```
> data(dc)
> pedigrees = dc$pedigrees
> datamatrix = dc$datamatrix
> loci = dc$loci
> x1 = Familias2linkdat(pedigrees, datamatrix, loci)
> p = oneMarkerDistribution(x1[[1]], ids=6, partialmarker=1)

Autosomal marker with the following partial data:
 ID D12S391
  1   20/20
  2   20/20
  3   20/21
  4   19/21
  5 18.3/19
  6     -/-

Marker allele frequencies:
      18.3         19         20         21
0.03124988 0.34136910 0.33482162 0.29255940

Mutation matrices:
$male
         18.3           19           20           21
18.3 0.999999 3.333333e-07 3.333333e-07 3.333333e-07
19   0.000001 9.949990e-01 4.545455e-03 4.545455e-04
20   0.000001 2.500000e-03 9.949990e-01 2.500000e-03
21   0.000001 4.545455e-04 4.545455e-03 9.949990e-01
attr(,"lumpability")
[1] NA

$female
         18.3           19           20           21
18.3 0.999999 3.333333e-07 3.333333e-07 3.333333e-07
19   0.000001 9.949990e-01 4.545455e-03 4.545455e-04
20   0.000001 2.500000e-03 9.949990e-01 2.500000e-03
21   0.000001 4.545455e-04 4.545455e-03 9.949990e-01
attr(,"lumpability")
[1] NA


Genotype probability distribution for individual 6:
18.3/18.3     19/19     20/20     21/21   18.3/19   18.3/20   18.3/21     19/20     19/21
   0.0000    0.0000    0.0000    0.0000    0.0002    0.5559    0.4416    0.0000    0.0000
    20/21
   0.0021

Total time used: 0 seconds.
```

Another example:

```
> data(grandmother)
> pedigrees = grandmother$pedigrees
> datamatrix = grandmother$datamatrix
> loci = grandmother$loci
```

```
> x1 = Familias2linkdat(pedigrees, datamatrix, loci)
> p1 = oneMarkerDistribution(x1[[1]], ids=3, partialmarker=1, verbose=FALSE)
```

8 R Session Information

```
> toLatex(sessionInfo())
```

- R version 3.3.3 (2017-03-06), x86_64-w64-mingw32
- Locale: LC_COLLATE=Norwegian (BokmÃël)_Norway.1252, LC_CTYPE=Norwegian (BokmÃël)_Norway.1252, LC_MONETARY=Norwegian (BokmÃël)_Norway.1252, LC_NUMERIC=C, LC_TIME=Norwegian (BokmÃël)_Norway.1252
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: fam2r 1.2, Familias 2.4, kinship2 1.6.4, Matrix 1.2-8, paramlink 1.1-0, quadprog 1.5-5, Rsolnp 1.16
- Loaded via a namespace (and not attached): assertthat 0.1, grid 3.3.3, lattice 0.20-34, maxLik 1.3-4, miscTools 0.6-16, parallel 3.3.3, sandwich 2.3-4, tools 3.3.3, truncnorm 1.0-7, zoo 1.7-13

References

[1] T Egeland, D Kling, and P Mostad. *Relationship Inference with Familias and R: Statistical Methods in Forensic Genetics.* Academic Press, 2015.