Likelihoods for pairs of markers

Hilde and Thore

2019-11-23

1 Objectives

We would like to estimate Jacquard coefficients and calculate LR's for linked markers in linkage disequilibrium. Based on this, the impact of linkage and LD can be evaluated. This ambitious goal is simplified by considering pairs independent markers for pairs of individuals.

2 Likelihoods and implementation assuming LE

The likelihood of two individuals being related according to $\mathbf{\Delta}$, given their genotypes $G = (g_1, g_2)$ at a marker is obtained by conditioning on the Jacquard state:

$$L(\mathbf{\Delta} \mid G) = \sum_{i=1}^{9} \Delta_i P(G \mid J_i). \tag{1}$$

This likelihood is calculated efficiently for a large number of markers in `inbred::likJ`, a function coded by Magnus.

The purpose of this section is to describe the extension to pairs of linked markers. The further extension to independent pairs of such pairs of markers is trivial. Let $J^{(2)}$ denote the 9 by 9 matrix of identity states of a pair of pedigree members, for a given recombination rate. Furthermore, $\Delta^{(2)}$ is the 9*9 matrix of two-locus condensed identity coefficients, for a given recombination rate. The STR-marker data for the first individual is $g_1 = (g_{11}, g_{12}) = (a_1/b_1, a_2/b_2)$ and for the second $g_2 = (g_{21}, g_{22}) = (c_1/d_1, c_2/d_2)$. Now $G = (g_1, g_2)$ contains marker data for both loci. By conditionining, the likelihood may be written

$$L(\mathbf{\Delta}, \mathbf{\Delta}^{(2)} \mid G) = \sum_{s,t=1}^{9} \Delta_{s,t}^{(2)} P(G \mid J_{s,t}^{(2)}). \tag{2}$$

Note that $\mathbf{\Delta}^{(2)}$ can be calculated exactly numerically as explained below. We therefore consider,

$$P(G \mid J_{s,t}^{(2)}) = P(g_1, g_2 \mid J_{s,t}^{(2)}) =^{LE} P(g_{11}, g_{21} \mid J_s) P(g_{12}, g_{22} \mid J_t) \tag{3}$$

where the LE assumption is indicated (in the next section we consider LD).

This can be written on matrix form, convenient for implementation. To this end, we let

$$u = (P(g_{11}, g_{21} \mid J_1), \ldots, P(g_{11}, g_{21} \mid J_9))$$
$$v = (P(g_{12}, g_{22} \mid J_1), \ldots, P(g_{12}, g_{22} \mid J_9))$$

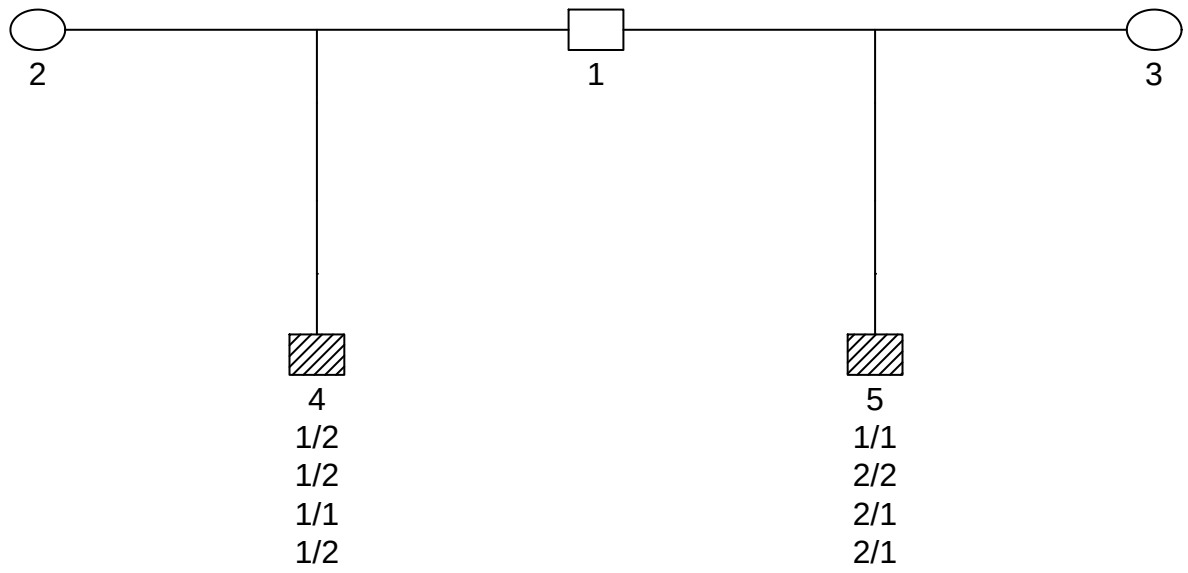
and write

$$L(\mathbf{\Delta}, \mathbf{\Delta}^{(2)} \mid G) = u \Delta^{(2)} v^T. \tag{4}$$

2.1 Implementation and examples

The numerical values in the below example is confirmed by FamLink http://famlink.se/f_index.html, Merlin and `pedprobr::likelihood`. First, the numerator hypothesis is plotted

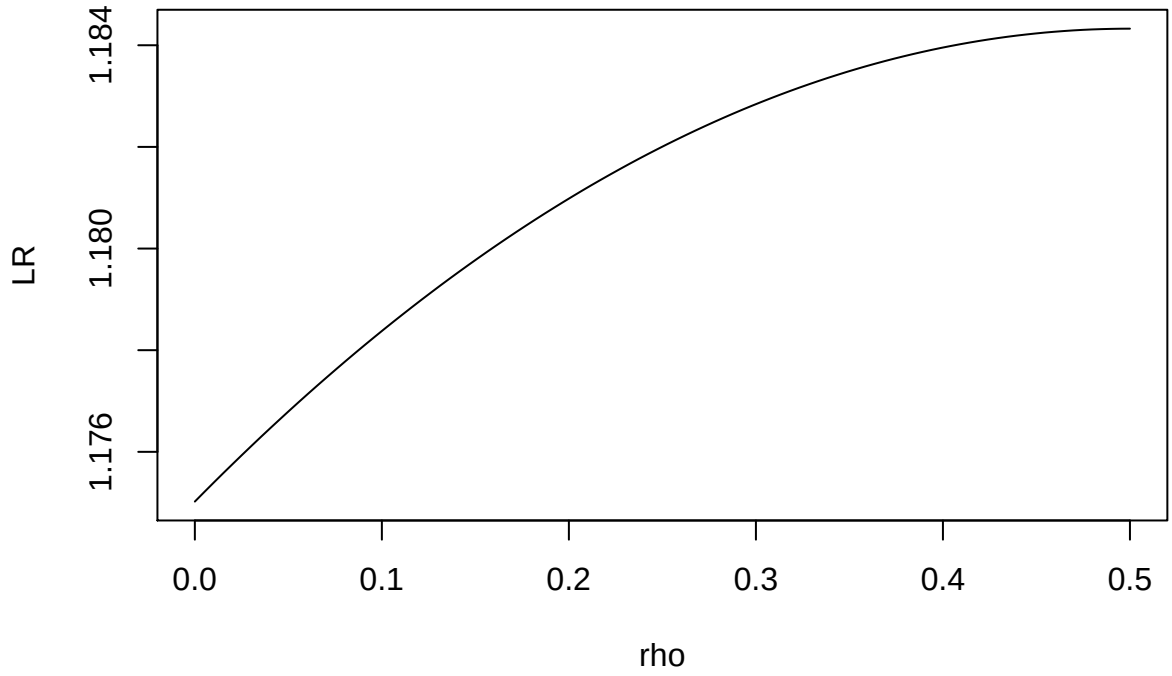
```
library(inbred) # https://github.com/thoree/inbred
library(pedtools)
library(ribd)
library(pedprobr)
p = c(0.4,  0.6)
a  = c(1, 1, 1, 1)
b  = c(2, 2, 1, 2)
cc = c(1, 2, 2, 2)
d  = c(1, 2, 1, 1)
pa = p[a]
pb = p[b]
pc = p[cc]
pd = p[d]
H1 = halfSibPed(1)
als = 1:length(p)
m = list()
for (i in 1:length(a))
   m[[i]] = marker(H1, afreq = p, alleles = als,
             "4" = c(a[i], b[i]), "5" = c(cc[i], d[i]) )
H1 = setMarkers(H1, m)
plot(H1,m, skip.empty.genotypes = TRUE, shaded = typedMembers(H1))
```



Next, the likelihood ratio is plotted as a function of the recombination rate in [0,0.5]

```
Delta2 = matrix(0, ncol = 9, nrow = 9); Delta2[9,9] = 1
Delta1 = ribd::condensedIdentity(H1, c(4,5))
```

```
lik2 = likPairs(a,b,cc,d, pa, pb, pc, pd, Delta = Delta1, DeltaMatrix = Delta2)
denominator = prod(lik2)
rho = seq(0, 0.5, length = 100)
LRs = rep(NA, 100)
Delta1 = ribd::condensedIdentity(H1, c(1, 2))
for (i in 1:100){
  Delta2 = twoLocusIdentity(H1, c(4,5), rho[i])
  lik1 = likPairs(a,b,cc,d, pa, pb, pc, pd, Delta = Delta1, DeltaMatrix = Delta2)
  LRs[i] = prod(lik1)/denominator
}
plot(rho, LRs, type = "l", xlab = "rho", ylab = "LR")
```



2.1.1 Function with `ped suite` input

We check against `pedprobr::likelihood`:

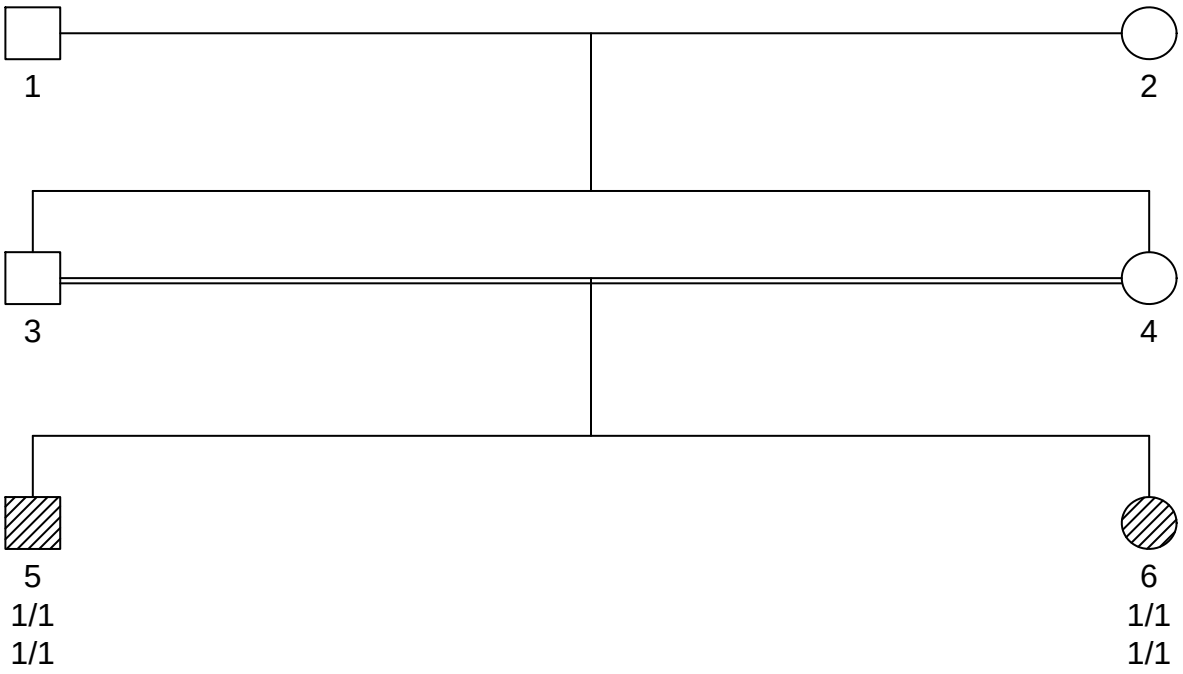
```
rho = 0.01
l1 = likelihood(H1, m[[1]], m[[2]], theta = rho)*
     likelihood(H1, m[[3]], m[[4]], theta = rho)
Delta2 = twoLocusIdentity(H1, c(4,5), rho)
l2 = likPairsPed(H1, c(4,5),  Delta = Delta1, DeltaMatrix = Delta2)
l2 = prod(l2)
abs(l1-l2) < 1e-12
```

```
## [1] TRUE
```

Next follows a case with inbreeding. First a plot

```
p = c(0.99,  0.01)
als = 1:length(p)
set.seed(17)
a  = sample(als, 2, rep = T)
b  = sample(als, 2, rep = T)
cc = sample(als, 2, rep = T)
d  = sample(als, 2, rep = T)
a  = c(1,1)
b  = c(1,1)
cc = c(1,1)
d  = c(1,1)
pa = p[a]
pb = p[b]
pc = p[cc]
pd = p[d]
H1 = fullSibMating(1)

m = list()
for (i in 1:length(a))
   m[[i]] = marker(H1, afreq = p, alleles = als,
             "5" = c(a[i], b[i]), "6" = c(cc[i], d[i]) )
H1 = setMarkers(H1, m)
plot(H1,m, skip.empty.genotypes = TRUE, shaded = typedMembers(H1))
```



Next, checking:

```
rho = 0.5
l1 = likelihood(H1, m[[1]], m[[2]], theta = rho)
ids = leaves(H1)
Delta1 = condensedIdentity(H1, ids)
Delta2 = twoLocusIdentity(H1, ids, rho)
l2 = likPairsPed(H1, ids, Delta1, Delta2)
abs(l1-l2) < 1e-12
```

```
## [1] TRUE
```

To see that the tailored implementation is, as it should be, quicker that the general `pedprobr::likelihood`, one can run

```
nM = 2
foo1 = function(nM = 100, H1, m , rho ){
  m[[1]][5:6,] = 1
  m[[1]][5:6,] = 1
  for ( i in 1:nM)
    l = likelihood(H1, m[[1]], m[[2]], theta = rho)
  l
}
foo2 = function(nM = 100, H1, Delta1, Delta2){
  m = list()
  for (i in 1:nM)
    m[[i]] = marker(H1, afreq = p, alleles = als,
                "5" = c(1, 1), "6" = c(1,1) )
  H1 = setMarkers(H1, m)
  l = likPairsPed(H1, c(5,6), Delta1, Delta2)
  l
}
system.time(foo1(nM = nM, H1, m, rho))
```

```
##    user  system elapsed
##    0.12    0.00    0.14
```

```
system.time(foo2(nM = nM, H1, Delta1, Delta2))
```

```
##    user  system elapsed
##    0.02    0.00    0.01
```

3 Dealing with LD

We continue to consider pairs of independent markers. We assume that allele frequencies are known.

3.1 A database of haplotype counts is available

Let $c_{ij},\ i,j=1,2,\ C=\sum_{i,j=1,2} c_{i,j}$ denote the counts of the corresponding haplotypes $[i-j]$ and the total number. If we trust the database and all haplotypes are observed, we can use the haplotype frequencies $h_{ij}=c_{ij}/C$. For STR-markers, with many alleles, the direct approach based on the database is not likely to work since chances are that some haplotypes will not be observed or at least not reliably estimated. Alternatively, we can use the lambda-model described in Ch 4 and 6 of Egeland, Kling and Mostad. This model is based on a Dirichlet prior for the haplotype frequencies. The counts are multinomial given the haplotype frequencies. This leads to (using that the Dirichlet and the Multinomial is a *conjugate* pair) haplotype frequencies

$$h_{ij}=\frac{c_{ij}+\lambda p_{ij}}{C+\lambda}$$

where p_{ij} are the haplotype frequencies LE would give. We see that large values of λ gives LE estimates while λ close to 0 produces the count, database, estimate. At any rate, below we assume h_{ij} are available.

It would be nice to express the likelihood parametrically, say as a function of recombination rate and some measure of LD: recall that our goal is to study how $LR(\rho, \lambda)$ or some estimate $\widehat{\theta}(\rho, \lambda)$ depend on linkage and LD, as measured by λ in the previous estimate.

3.2 Likelihood with LD

Consider once more

$$A(s,t) = P(G \mid J^{(2)}_{s,t}) = P(g_1, g_2 \mid J^{(2)}_{s,t}) = P(g_1 \mid J^{(2)}_{s,t}, g_2) P(g_2 \mid J^{(2)}_{s,t})$$

In presence of LD we cannot consider the loci independently even if we have complete information on IBD status.

We need to calculate the 91 terms of the matrix A.

Hilde: Sett inn det du skrev "Define ...."

3.2.1 Example 1

Consider first unrelated individuals, but with markers in LD. Obviously, linkage is irrelevant in this case. Then Eq (2) simplifies to

$$L(\boldsymbol{\Delta}, \boldsymbol{\Delta}^{(2)} \mid G) = P(g_1)P(g_2)$$

Consider a pair of markers with haplotype frequencies h_{ij}.

Let $I_l = 1$ if and individual is homozygous at marker $l = 1, 2$ and 0 otherwise. Then for individual 1 (and similarly for individual 2)

$$\begin{aligned} P(g_i = (a_1/b_1, a_2/b_2)) & \\ &= I_1 I_2 h^2_{a_1,a_2} \\ &+ 2(1-I_1) I_2 h_{a_1,a_2} h_{b_1,a_2} \\ &+ 2 I_1 (1-I_2) h_{a_1,b_1} h_{a_1,b_2} \\ &+ 2(1-I_1)(1-I_2)(h_{a_1,a_2} h_{b_1,b_2} + h_{a_1,b_2} h_{b_2,a_2}). \end{aligned}$$

Below we show that the probabilities add up to one as they should for a pair of SNP-markers. Next the function for given marker data is defined and tested for LE and LE:

```
# Calculate P(g = (a1/b1,a2/b2))
pG = function(a1, a2, b1, b2, h){
# a1 First allele, first marker
# a2 First allele, second marker
# b1 Second allele, first marker
# b2 Second allele first marker
# h matrix of haplotype probabilities
  I1 = a1 == b1
  I2 = a2 == b2
  q11 = I1*I2*h[a1,a2]^2
  q21 = 2*(1-I1)*I2*(h[a1,a2]*h[b1,a2])
  q12 = 2*I1*(1-I2)*h[a1,a2]*h[a1,b2]
  q22 = 2*(1-I1)*(1-I2)*(h[a1,a2]*h[b1,b2]+h[a1,b2]*h[b2, a2])
  q11 + q21 + q12 + q22
}
```

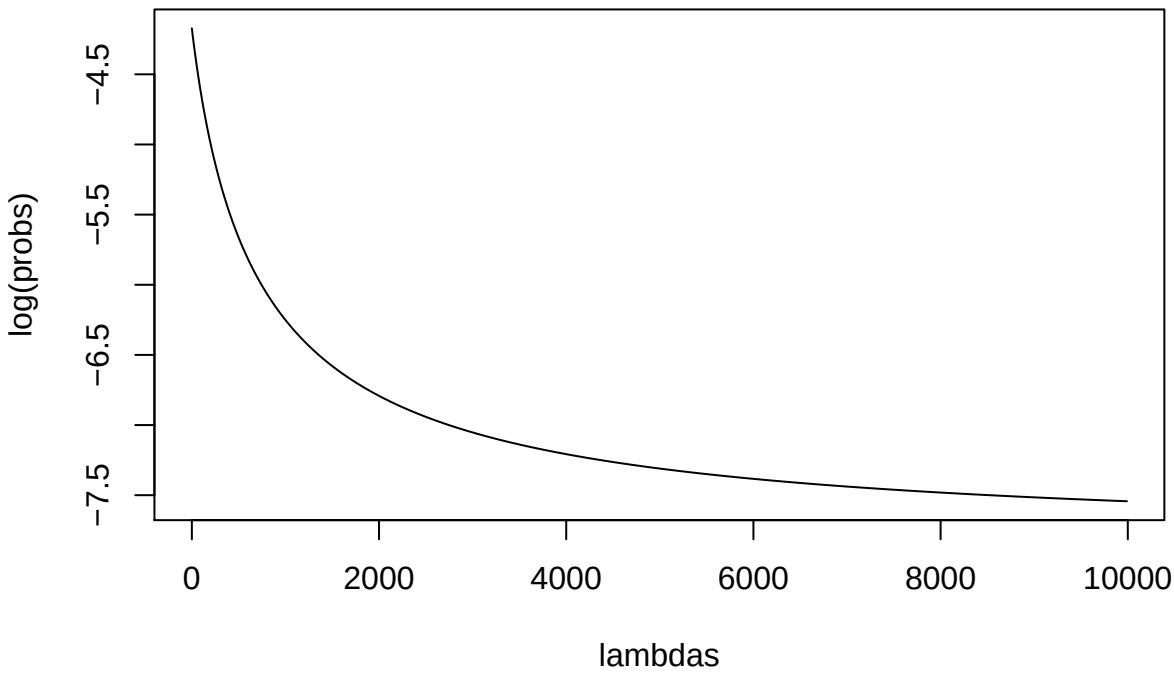
```r
p = c(0.4, 0.6); q = c(0.5, 0.5)
h = p%o%q #Haploptype frequencies with LE
# abs(pG(1, 1, 1, 1, h) - p[1]^2*q[1]^2) < 1e-15
# abs(pG(1, 1, 2, 1, h) - 2*p[1]*p[2]*q[1]^2) < 1e-15
# abs(pG(1, 1, 2, 2, h) - 2*p[1]*p[2]*2*q[1]*q[2]) < 1e-15
# All possibilities
S = rbind( c(1, 1, 1, 1),
           c(1, 2, 1, 1),
           c(2, 2, 1, 1),
           c(1, 1, 1, 2),
           c(1, 2, 1, 2),
           c(2, 2, 1, 2),
           c(1, 1, 2, 2),
           c(1, 2, 2, 2),
           c(2, 2, 2, 2)
           )
prob1 = apply(S, 1, function(x,h) pG(x[1], x[3], x[2], x[4],h), h)
abs(sum(prob1)-1) < 1e-15
```

```
## [1] TRUE
```

```r
h = diag(c(0.4, 0.6)) # Strong LD
prob2 = apply(S, 1, function(x,h) pG(x[1], x[3], x[2], x[4],h), h)
h = matrix(c(0.0, 0.2,0.8,0.0), ncol = 2, nrow = 2) # Strong LD
prob3 = apply(S, 1, function(x,h) pG(x[1], x[3], x[2], x[4],h), h)
```

Assume next we have haplotype data

```r
# Generata haplotypes for triallelic marker according to lambda model
set.seed(17)
dat = matrix(ncol = 3, nrow = 3, sample(20:50, 9, rep = T))
C = sum(dat)
freq1 = c(0.1,0.5, 0.4); freq2 = c(0.2, 0.3,0.5)
p = freq1%o%freq2
lambdas = c(0, seq(1, 10000, by = 10))
probs = rep(NA, length(lambdas))
i = 0
for (lambda in lambdas){
  i = i + 1
  h = (dat +lambda*p)/(C+lambda)
  probs[i] = pG(1,1,1,1,h)
}
plot(lambdas, log(probs), type = "l")
```




```
freq1[1]^2*freq2[1]^2 -probs[length(probs)]# LE prob
```

```
## [1] -0.0001297737
```

3.2.2 Example 2

Barely started In search for a general formula of practical implementation, we consider an example. Assume that the individuals are homozygous for a for the first marker and homozygous for b for the second. In this case the number of possibilities to consider is reduced since $A(s,t) = A(t,s)$.

3.3 What next

Consider only linkage first

1. More examples, plot as function of recombination rate.
2. Simulate data in Merlin. Calculate LR in Merlin. Read data into R. How well does the pairwise approximation do?
3. Expand code to allow for different recombination rates between markers.

Consider next LD.

4. Complete above example. Ignore inbreeding. Check for LE.
5. Try to implement with LD. The code in the function `likJ` can determine the "case". Can we based on this get the general implementaton.

Other software

6. Experiment with FamLink
7. Experiment with Merlin

3.4 Obsolete?

Consider the first term.

Let $I_1 = 1$ if all alleles of the first marker are the same, say a. Similarly, $I_2 = 1$ if all alleles of the second marker are the same, say b. Then $P(G \mid J^{(2)}_{1,1}) = I_1 I_2 h_{a,b}$. Simillar expression are needed for the remaining terms. The function `inbred::likJ` can be used, perhaps be extended to find all terms.

Try likelihood for first Jacquard state. *Magnus*: likelihood with selfing

```
x = selfingPed(2, sex = 1)
p = c(0.4, 0.6)
als = 1:length(p)
m1 = marker(x, "2" = 1, "3" = 1, alleles = als, afreq = p)
m2 = marker(x, "2" = 1, "3" = 1, alleles = als, afreq = p)
x = setMarkers(x, list(m1, m2))
founderInbreeding(x,1) = 1
rho = 0.5
Delta2 = twoLocusIdentity(x, c(2,3), rho)
Delta1 = condensedIdentity(x, c(2,3))
likelihood(x, m1, m2, theta = rho)
```

```
## [1] 0.00065536
```

```
likPairsPed(x, c(2,3), Delta1, Delta2)
```

```
## [1] 0.16
```

Another try