Likelihoods for pairs of markers

Thore

2019-11-18

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 of SNP-markers for pairs of individuals. The pairs of markers are assumed independent.

2 Likelihoods and implementation assuming LE

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

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

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. Now G contains marker data for both loci. By conditionining, the likelihood may be written

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

Furthermore,if we assume LE

$$L(\boldsymbol{\Delta}, \boldsymbol{\Delta}^{(2)} \mid G) = \sum_{s,t=1}^{9} \Delta_{s,t}^{(2)} P(g_1 \mid J_s) P(g_2 \mid J_t). \tag{3}$$

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

$$u = u(\boldsymbol{\Delta}) = (P(g_1 \mid J_1), \ldots, P(g_1 \mid J_9))$$
$$v = v(\boldsymbol{\Delta}) = (P(g_2 \mid J_1), \ldots, P(g_2 \mid J_9))$$

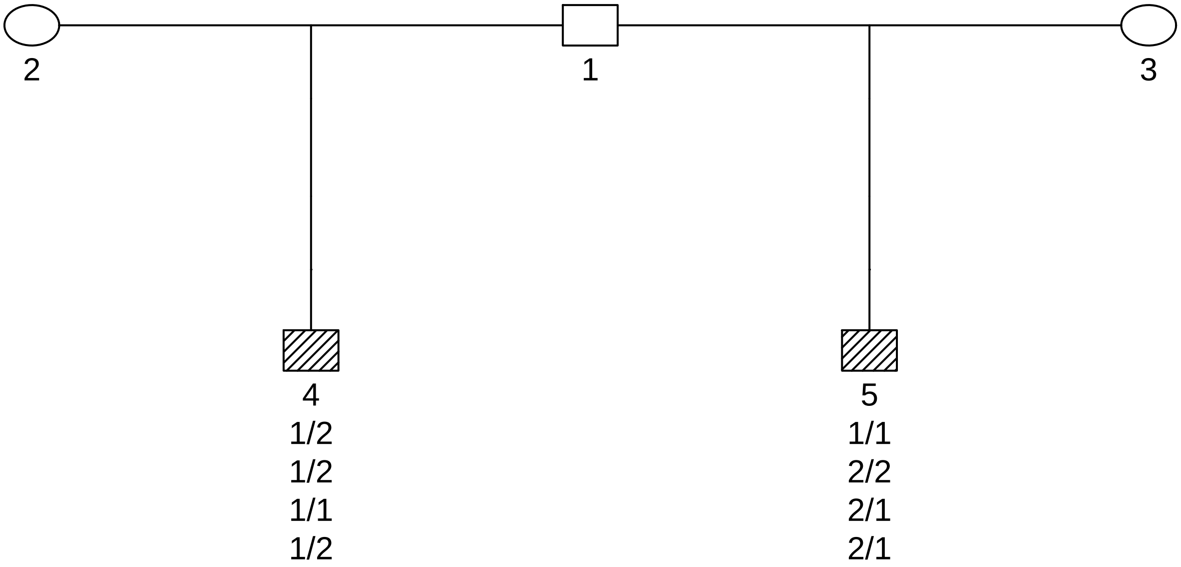
and write

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

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

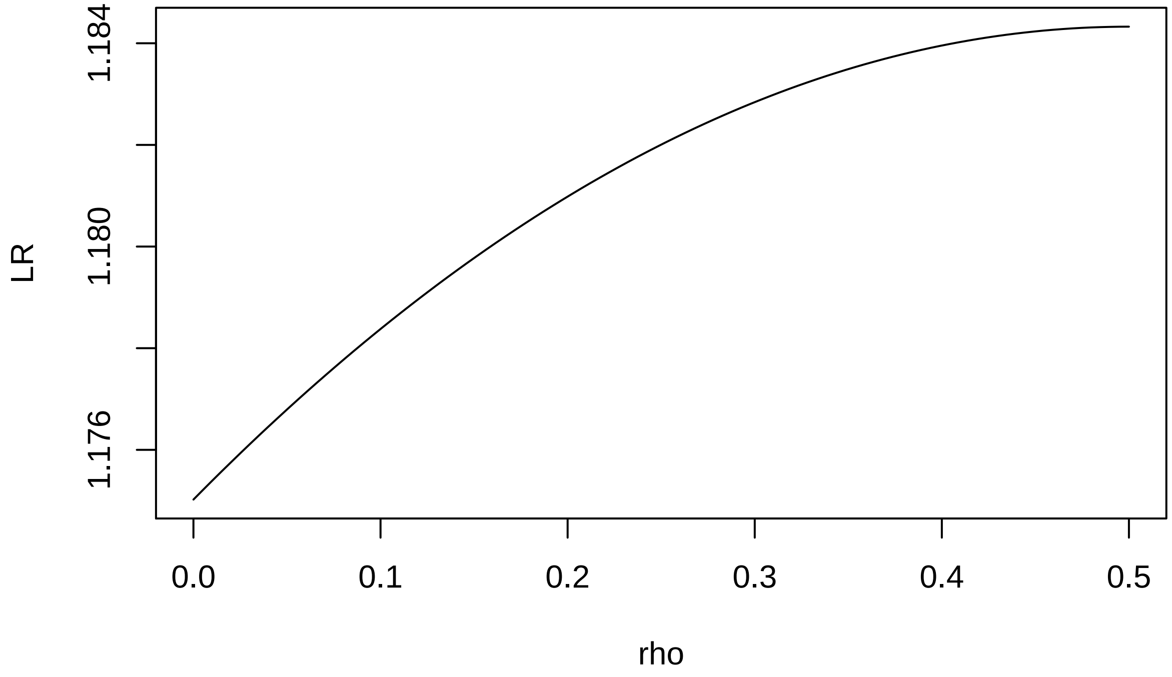
```r
library(inbred) # https://github.com/thoree/inbred
library(pedtools)
library(ribd)
library(pedprobr)
p = c(0.4,  0.6)
a  = c(1, 1, 1, 1)
b  = c(2, 2, 1, 2)
cc = c(1, 2, 2, 2)
d  = c(1, 2, 1, 1)
pa = p[a]
pb = p[b]
pc = p[cc]
pd = p[d]
H1 = halfSibPed(1)
als = 1:length(p)
m = list()
for (i in 1:length(a))
   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]

```r
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`:

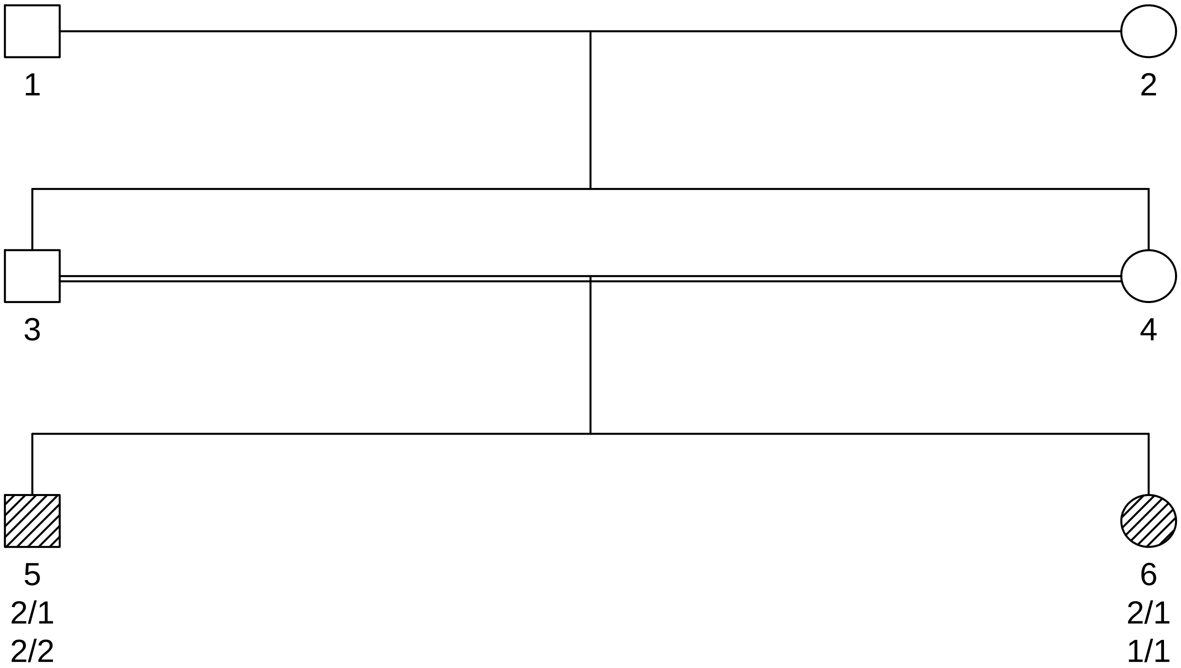
```
rho = 0.01
l1 = likelihood(H1, m[[1]], m[[2]], theta = rho)*
     likelihood(H1, m[[3]], m[[4]], theta = rho)
Delta2 = twoLocusIdentity(H1, c(4,5), rho)
l2 = likPairsPed(H1, c(4,5),  Delta = Delta1, DeltaMatrix = Delta2)
l2 = prod(l2)
abs(l1-l2) < 1e-12
```

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

Next follows a case with inbreeding. First a plot

```
p = c(0.4,  0.6)
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)
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 = 0.01
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
##    0.13    0.00    0.13
```

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

```
##    user  system elapsed
##       0       0       0
```

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

$$L(\boldsymbol{\Delta},\boldsymbol{\Delta}^{(2)} \mid G) = \sum_{s,t=1}^{9} \Delta_{s,t}^{(2)} P(G \mid J_{s,t}^{(2)}). \tag{5}$$

We need to calculate the 91 terms (reduced to 9 terms if inbreeding is ignored, perhaps a wise simplification not pursued now) $P(G \mid J_{s,t}^{(2)})$. 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 $P(G \mid J_{1,1}^{(2)}) = I_1 I_2 h_{a,b}$. Simillar expression are needed for the remaining terms. The function `inbred::likJ` can be used, perhaps be extended to find all terms.