Probabilities of multilocus genotypes in SIB recombinant inbred lines

Kamel Jebreen, Marianyela Petrizzelli, Olivier C. Martin

1 Introduction

Set the working directory to the emplacement of the source file “PMG_SIB_RILs.R” and load it to the working environment. You can download the “PMG_SIB_RILs.R” file from https://github.com/olivier-c-martin/PMG_SIB_RILs.git,

```
setwd("the directory")
source("PMG_SIB_RILs.R.R")
```

Then load the required packages:

```
library(eply)
library(rlist)
library(rmarkdown)
```

2 Input

This code just needs as inputs the number of loci L_loci and the recombination rates for successive intervals which is a vector of length L_loci - 1. For example, for L_loci=3

```
L_loci = 3
recRates = c(0.4, 0.2)
```

3 The listing of variables used (all Q’s and all non-equivalent Q’s)

To gain time we find the list of inheritance indices (all Q’s) that contribute in the system (till L_loci =10)

```
allQs = list.load("allVarTillL=10.rds")
allvpForallup = list.load("allContrVarTillL=10.rds")

nonEquivalentQs = allQs[[L_loci]]$symQs
allQsMappedToNonEquivalent = allQs[[L_loci]]$nonsymQs
allvpForallup = allvpForallup[[L_loci]]
```

Or you can reconstruct all these objects by direct calculation:

```
allQs = systemVar(L_loci)
```

```
##
##  1. Find the inheritance indices that are contributing to this system, please wait: ...
```

Note that the non-equivalent Q’s are

```
nonEquivalentQs = allQs$symQs
```

and the 4^L Q’s are mapped to the non-equivalent ones via

```r
allQsMappedToNonEquivalent = allQs$indicesAllQs
```

Note that the first equation of the linear system is given by the fact that $\sum(Q's) = 1$

```r
multiplicityQs = table(allQsMappedToNonEquivalent)
```

This means that

```
## 4Q(000)+4Q(001)+8Q(002)+4Q(010)+4Q(011)+8Q(012)+8Q(020)+8Q(021)+8Q(022)+8Q(023)=1
```

Now determine the list of all v' given the u' to be used to construct the self-consistent equations.

```r
allvpForallup = allvprimeForEachuprime(nonEquivalentQs)
```

```
##
##  2. calculate all the the contributing v_prime values contributing to the self-consistent equation fo
## 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
```

4 Find the system AQ = B

The analytic expressions of the system of linear equations to be solved are

```r
  analyticEquations = twoWayRILsib(L_loci, nonEquivalentQs, allQsMappedToNonEquivalent, allvpForallup
```

```
##
##  # ====== 3  - Loci ====== #
##
##  3. The First equation in the system is:
## 4Q(000)+4Q(001)+8Q(002)+4Q(010)+4Q(011)+8Q(012)+8Q(020)+8Q(021)+8Q(022)+8Q(023)=1
##
##  4. Computing the self-consistent equations: ...
## 1 of 9
2 of 9
3 of 9
4 of 9
5 of 9
6 of 9
7 of 9
8 of 9
9 of 9
##  done
```

Hence, the matrix of equations is

```r
Amatrix = analyticEquations$A
Amatrix[1:3,1:2]
```

```
##       000                         001
## SumQs "4"                         "4"
## 000   "2*(0.5)*(1-r12)*(1-r23)-1" "0"
## 001   "2*(0.5)*(1-r12)*(r23)"     "-1"
```

and, the coefficient matrix is

```r
Bvector = analyticEquations$B
Bvector[1:3]
```

```
## [1] 1 0 0
```

Now substitute the numerical values of the recombination rates

```
numericAmatrix = evalMatrix(A = Amatrix, recRates = recRates)
```

```
##
##  5. Evaluate the input matrix, substituting the numerical values of the recombination rates:
##  r12 = 0.4     r23 = 0.2     r13 = 0.44
```

For example,

```
numericAmatrix[1:3,1:2]
```

```
##       [,1] [,2]
## [1,]  4.00    4
## [2,] -0.52    0
## [3,]  0.12   -1
```

Hence, the solution for the $N_Q(L)$ unknown (non-equivalent) Q's is

```
solution = solve(numericAmatrix, Bvector)
names(solution) = nonEquivalentQs
solution
```

```
##        000        001        002        010        011        012
## 0.03413811 0.01322519 0.01308306 0.01125016 0.02666891 0.01045223
##        020        021        022        023
## 0.01164653 0.01027273 0.02641467 0.01048960
```

This means,

```
##  Q(0,0,0) =  0.03413811 , Q(0,0,1) =  0.01322519 , Q(0,0,2) =  0.01308306
##  Q(0,1,0) =  0.01125016 , Q(0,1,1) =  0.02666891 , Q(0,1,2) =  0.01045223
##  Q(0,2,0) =  0.01164653 , Q(0,2,1) =  0.01027273 , Q(0,2,2) =  0.02641467
##  Q(0,2,3) =  0.0104896
```

5 Use the Q's to compute the RIL

We go from the Q's (RIL IBD probabilities) to RIL multilocus genotype probabilities by computing the 2^L probabilities of multilocus SIB RIL genotypes

```
allProbabilitiesOfRilGenotypes= QsToGenotypeProbabilities(L_loci, solution)
```

```
##
##  6. Going from  Q's (RIL IBD probabilities) to RIL multilocus genotype probabilities:
## 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
```

```
allProbabilitiesOfRilGenotypes
```

```
## [1] 0.17056473 0.09414115 0.08767703 0.14761709 0.14761709 0.08767703
## [7] 0.09414115 0.17056473
```

Note that the sum of these probabilities should be one

```
sumAllProbabilities = sum(allProbabilitiesOfRilGenotypes)
sumAllProbabilities
```

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

For more details see (Jebreen et al., 2019).

Bibliography

Jebreen, K., Petrizzelli, M., and Martin, O. C. (2019). Probabilities of multilocus genotypes in sib recombinant inbred lines. *Statistical Genetics and Methodology, a section of the journal Frontiers in Genetics(Submitted).*