Loops in FASTLINK

Alejandro A. Schäffer
Rice University

This document is meant to accompany FASTLINK, version 2.0 and beyond. It describes some aspects of pedigree loops in FASTLINK. This material is based on pages 170–171 of *Analysis of Human Genetic Linkage* by Ott, Section 2.8 of Linkage Analysis Package II: User's Guide To Programs, and Section 4 of "Avoiding Recomputation in Linkage Analysis" by A. A. Schäffer, S. K. Gupta, K. Shriram, and R. W. Cottingham Jr. (this is paper2.ps that comes with the FASTLINK distribution starting at version 2.0). It differs from these sources in that the presentation here is more comprehensive and more colloquial in nature. Thanks to Brian Nichols (U. Iowa) for asking enough questions about loops to motivate me to write this document. Thanks to Sandeep Gupta (Rice University) for helpful comments on my first draft. Unless otherwise noted, comments apply both to LINKAGE and FASTLINK.

Inbreeding Loops vs. Marriage Loops

After reading many papers with pedigree pictures, I have come to the conclusion that geneticists think it is important to distinguish between inbreeding (consanguinity) loops and marriage loops. For the purposes of pedigree traversal algorithms in LINKAGE this distinction is irrelevant. Simply put, a loop represents a cycle in the nuclear family graph defined in the accompanying document *Pedigree Traversal in FASTLINK*.

Maxloop

LINKAGE and FASTLINK have a constant called maxloop. In order for the programs to run correctly, maxloop must be at least as large as the number of loops in any of the input pedigrees. There are no significant negative consequences for making maxloop larger than it needs to be. In LINKAGE 5.1 and all versions of FASTLINK, maxloop is set to 2 in the distribution; the user is free to increase the value. Rumor has it that some versions of LINKAGE have been distributed with maxloop set to 1.

So far as I can tell, there is no correctness harm in trying input pedigrees with arbitraily large number of loops, so long as maxloop is set at least as large as the number of loops. Depending on the data set, the running time may increase exponentially with the number of loops, but the space requirements will not increase substantially.

Counting and Breaking Loops

In terms of the nuclear family graph the number of loops is the minimum number of edges whose removal from the nuclear family graph makes that graph into a tree (i.e. a connected graph with no cycles). The way to break a loop is to make one of the participating individuals a into two people: one person acts as parent and the other acts as child and sibling. In the nuclear family graph this is equivalent to deleting the edge between the two vertices representing nuclear families in which person a is a member.

Encoding Loops in pedin.dat

I have learned the hard way that the way loops are encoded in pedin.dat is not prominently explained in the LINKAGE documentation. Here is my attempt at a comprehensive explanation.

The presence or absence of loops is encoded in column 9 of pedin.dat. The entries in column 9 range from 0 to (1 + number of loops). For each loop the user must designate one individual to be the "loop breaker" (this is not a standard term). The user should then make two copies of the row of data for the loop breaker. All the genotype data in those two rows is the same, but the first columns are different. In one instance the loop breaker has no parents, but has children, and in the other the loop breaker has children but no parents.

The two copies of the loop breaker for the i^{th} loop must have the number i+1 in column 9 unless one of them is also the proband in which case one copy has a 1 and the other has i+1 in column 9. I.e., the two copies of the loop breaker for the first loop have a 2, the two copies of a loop breaker for the second loop have a 3, etc. In each pedigree the user may designate one non-loop breaker to be the "proband" by giving that person a 1 in column 9. Any other individuals who are not the proband and not loop breakers get a 0 in column 9. Note that by this definition, each individual can be the loop breaker for at most one loop.

For example two lines that start:

1 3 0 0 7 0 0 1 2 [Genotype data here]
1 4 1 2 0 5 5 1 2 [same genotype data here]

would encode a loop. The 1 in column 1 indicates pedigree number 1. The 3 and 4 four in column 2 indicate the two numbers assigned to the two copies of the loop breaker. This person is a male due to the 1 in column 8 and is a loop breaker due to the 2 in column 9. Individual number 3 acts as a parent; the 7 column 5 indicates that individual 7 is the first child of the loop breaker. Individual number 4 acts as a child and sibling. The 1 in column 3 indicates that 1 is the father of the loop breaker. The 2 in column 4 indicates that 2 is the mother of the loop breaker. The 5 in columns 6 and 7 indicates that individual number 5 is the next paternal sibling and next maternal sibling of the loop breaker. The numbers assigned to the two copies of the loop breaker in column 2 need not be consecutive. The two lines for a loop breaker do not have to occur consecutively in pedin.dat.

Here is a tabular summary of what the columns encode:

Column 1: Pedigree number
Column 2: Individual number
Column 3: Number of father
Column 4: Number of mother
Column 5: First child
Column 6: Next sibling with same father
Column 7: Next sibling with same mother
Column 8: Male (encoded as 1) or female(encoded as 2)
Column 9: Neither proband nor loop breaker (0), proband (1), or loop-breaker for loop i $(i+1)$

Loop Ordering

One of the things that the loop encoding system implies is that the user gets to choose the numbering of the loops. For example, suppose there are three loops. Then there will be three loop breaker pairs of individuals (in pedin.dat). The user can choose which pair gets 2 in column 9, which pair gets 3 in column 9, and which pair gets 4 in column 9. The programs should give the same answer no matter which loop ordering is chosen.

However, in FASTLINK 2.0, the first loop, whose loop breakers get a 2 in column 9, is treated specially. What this means for the user is that the choice of which loop gets encoded as 2 can have a drastic effect on running time, but not correctness of the results.

I am still trying to get a sense of how many pedigrees in common usage have more than one loop. If this is sufficiently common, I will consider automating the choice of which loop gets treated as the first loop for algorithmic purposes, although this is not a trivial algorithmic task.

FASTLINK runs on pedigrees with loops can take a long time. When there is more than one loop, it is worth trying each loop as the first loop, to see which choice runs fastest. The choice of loop for the first which runs fastest should not depend on the number or choice of loci. Therefore, it is possible to do a simple 2-locus run with one candidate θ vector using MLINK to decide which loop is best as the first loop.

More generally, even when there are no loops, it is a good idea to start with just a simple LINKMAP run or MLINK run where you ask it to do only one candidate theta vector. This will tell you how long it takes per candidate theta and give you a good idea of how long the full scale run(s) you want to do will take.

Loop Algorithm

The basic algorithm for handling loops in LINKAGE is described in Ott's book, pages 170–171. We describe it here using more algorithmic language. As described in the accompanying document on *Pedigree Traversal in FASTLINK*, the basic goal is to compute for each individual x and each genotype g the conditional probability $P(x, g)$ that x has genotype g. This probability is conditioned on the parts of the pedigree traversed so far and the candidate θ.

The probability that x has genotype g may depend on the genotype of the loop breaker. Because the loop breaker is treated as two different people, we must force those two people to have the same genotype during a given traversal. This means that we must re-traverse the pedigree for each possible genotype of the loop breaker and sum the probabilities for each traversal. For example, suppose the loop breaker may have any of three genotypes $\{1, 2, 3\}$. Then for $i = 1, 2, 3$ we compute the conditional probability $P_i(x, g)$ that x has genotype g where we condition on the additional assumption that the loop breaker has genotype i. Then $P(x, g)$ would be computed as $(P_1(x, g) * B(1)) + (P_2(x, g) * (B_2)) + (P_3(x, g) * B_3)$, where $B(i)$ is the probability that the loop breaker has genotype i.

Since each possible genotype of the loop breaker requires another complete traversal of the loop, it is highly desirable to choose loop breakers that have the fewest number of possible genotypes. When there are multiple loops, the algorithm must do one traversal for each combination of possible genotypes of all the loop breakers. For example, if the first loop breaker has

3 possible genotypes and the second loop breaker has 4 possible genotypes, $3 \times 4 = 12$ traversals of the pedigree will be done for each likelihood evaluation. This explains why it may be impractical to handle pedigrees with more than 2 loops, which in turn explains why the code is distributed with maxloop set to 2.

In FASTLINK 2.0, I introduced the following optimization. Those parts of the pedigree whose conditional genotype probabilities are independent of the genotype of the loop breaker for the first loop (labeled as 2 in column 9 of pedin.dat) are only traversed once. In LINKAGE these parts are traversed once for each possible genotype of the loop breaker.