The Mystery of (the) Unknown (revised for version 4.1P)

Alejandro A. Schäffer
formerly Rice University
and
currently National Institutes of Health

This document describes some aspects of the UNKNOWN auxiliary program that is distributed with LINKAGE and FASTLINK. This document was originally written to accompany FASTLINK, version 2.2 and beyond. This version is a revision meant to accompany FASTLINK, version 4.1P and beyond. Due to the changes in version 3.0P, the contents of this document are logically intertwined with the contents of loops.ps. I recommend that you read loops.ps first, unless you want to know what UNKNOWN does only on loopless pedigrees. This material is a significant expansion of the discussion on pages 25–26 of *Handbook of Human Genetic Linkage* by Joseph Douglas Terwilliger and Jurg Ott. The emphasis there is on usage of UNKNOWN and all algorithmic aspects are suppressed. As with the documents on traversals and loops that first accompanied FASTLINK 2.1, this document is very informal and directed at those who may wish to modify the code.

Thanks to Jerry Halpern (Stanford) for suggesting that I prepare this document. Thanks to Joe Terwilliger (Columbia) for straightening out my confusion about lots of LINKAGE things, including the different versions of UNKNOWN.

UNKNOWN from a User's Perspective

The main purpose of UNKNOWN is to rapidly identify which genotypes are possible for individuals typed as unknowns in the input pedigree. Starting with FASTLINK 4.0P an additional purpose of UNKNOWN is to select loop breakers for looped pedigrees (see README.lselect and loops.ps). It is a good idea to run UNKNOWN just before running any of the main programs (i.e. LODSCORE, ILINK, LINKMAP, or MLINK) in LINKAGE/FASTLINK. Most versions in circulation of the main programs actually require that UNKNOWN be run due to file name conventions, in the sense that they expect to find the output files that UNKNOWN produces available as input files. The shell scripts produced by LCP for ILINK, LINKMAP, and MLINK will call UNKNOWN by default.

More details on UNKNOWN Specifications

We make one terminology convention. For the rest of this document, the term *joint genotype* refers to the multilocus joint genotype at the loci specified for the analysis; in particular, the genetic information for loci specified in `pedin.dat`, but ignored in the analysis is not incorporated. The term *genotype* when not preceded by "joint" refers to the genotype at a single locus. This convention is used for this document only and is actually rather inconvenient for discussing other aspects of LINKAGE/FASTLINK.

UNKNOWN produces four files: `speedfile.dat`, `ipedfile.dat`, `newspeedfile.dat`, and `loopfile.dat`; the last two are not used in LINKAGE versions of UNKNOWN or in versions of FASTLINK before 3.0P. On systems that limit file names to 8 characters, `speedfile.dat` and `newspeedfile.dat` are called `speedfil.dat` and `newspeedfil.dat`. On many systems, the default shell scripts produced by LCP delete the files `speedfile.dat` and `ipedfile.dat` before the run is completed.

UNKNOWN expects that for each pedigree in `pedin.dat`, the individuals in that pedigree are numbered $1, 2, 3$, in increasing order with no numbers skipped. This assumption is plausible because it will always hold if `pedin.dat` is prepared with the LINKAGE auxiliary program MAKEPED. Attempts to cook `pedin.dat` by hand often lead to errors when it is fed to a script produced by LCP. See README.trouble for guidance on troubleshooting when such errors arise.

`ipedfile.dat` is a pedigree file that looks very much like `pedin.dat`, which is the initial input pedigree file. There are at least four notable differences between `pedin.dat` and `ipedfile.dat`.

1. Genotype information in `ipedfile.dat` is restricted to those loci being used in the current analysis, while `pedin.dat` may have genotype information for lots of loci.

2. Some genotypes may be filled in.

3. Text comments in `pedin.dat` are not copied into `ipedfile.dat`

4. Spacing and indentation may differ even in the case where all the loci in `pedin.dat` are used for the analysis.

FASTLINK 4.0P introduced a much more fundamental distinction between `pedin.dat` and `ipedfile.dat` for looped pedigrees. UNKNOWN now attempts to improve on the user's selection of loop breakers. If such an improvement is deemed possible, it is achieved by rearranging and renumbering the people in the pedigree file. For more information on loop breaker selection see `loops.ps`.

FASTLINK 4.1P introduced the ability to use UNKNOWN to choose loop breakers from scratch. This replaces the complicated method suggested on pages 93-96 of *Handbook of Human Genetic Linkage* by Terilliger and Ott. The new method uses a special flag for UNKNOWN, which is "-l". If you run the command `unknown -l` either when `pedfile.dat` has loop breakers selected or when `pedfile.dat` has no loop breakers selected, the program will write out a new pedigree file with a good set of loop breakers selected automagically for you into `lpedfile.dat`. See README.lselect for more complete and precise instructions.

Although this is not enforced syntactically between programs, the same conventions should be used to identify unknown genotypes in `pedin.dat` and `ipedfile.dat`. In particular,

1. For loci specified by affection status, the constant `missaff` specifies the value used for unknown. By convention, `missaff` is defined to 0.
2. For loci specified by quantitative measures, the constant `missval` specifies the value used for unknown. By convention, `missval` is set to 0.0.
3. For loci specified by binary factors, any combination of binary factors which is not one of the possibilities listed in datain.dat for that locus will be treated as unknown.

Changing the definitions of `missaff` or `missval` in any of the source code files is likely to lead to computational disasters.

We define a person to be *speedfile–unknown* if that person's joint genotype is not completely specified in `pedin.dat` and cannot be inferred from the genotype information of relatives. Otherwise a person is *speedfile-known.*

For all speedfile-known individuals the complete joint genotype appears in `ipedfile.dat` and no information appears in `speedfile.dat`.

For all speedfile-unknown individuals that have at least one child in the pedigree, information about their possible genotypes at each locus is given in `speedfile.dat`. In `speedfile.dat` individuals are numbered from 1 to the number of individuals in all pedigrees together; no pedigree numbers are shown. For each speedfile-unknown individual a list of triples is displayed. In a triple, the first number is a locus number. The second and third numbers are possible alleles at that locus. For example, the triple

$$3 \quad 1 \quad 3$$

means that at locus 3, the genotype 1 3 is possible. The possible triples are written out to `speedfile.dat` in the routine `writespeed`.

It is important to clarify several subtle points about the triple representation.

First, the loci are numbered 1 to number of loci in the analysis, and are not numbered with respect to `pedin.dat`.

Second, all loci are encoded by allele numbers in the guts of the computations in UNKNOWN and all the LINKAGE programs, regardless of which format is used to enter the data. Hence allele numbers are used for `speedfile.dat` output. In contrast, `ipedfile.dat` preserves whatever format is used in `pedin.dat` for each locus.

Third, unlike many places in the LINKAGE programs, UNKNOWN treats genotypes as *ordered* pairs. Thus if the genotype 1 3 is possible, then the genotype 3 1 is also possible and will be listed in a separate triple.

Fourth, if a person's genotype can be inferred at some loci, but not at other's, the possible genotypes at all loci will be listed. For those loci where the genotype is known, `speedfile.dat` will contain 1 or 2 triples depending on whether the known genotype is homozygous or heterozygous.

Fifth, if a person's genotype can be partially, but incompletely inferred to the extent that one allele is known, the known allele does not show up in `ipedfile.dat`. Only when the full genotype at a locus is known does the information appear in `ipedfile.dat`.

Sixth, starting with FASTLINK 3.0P, the code implements allele amalgamation, if the constant `ALLELE_SPEED` is set to 1. Allele amalgamation applies when at least two alleles at the same locus are unused in the same pedigree. Details about allele amalgamation can be found in `README.allele`. What is significant for users of UNKNOWN is that `ipedfile.dat` and `speedfile.dat` use pre-amalgamation allele numbers, while `newspeedfile.dat` and `loopfile.dat` (see section on Loops below) use post-amalgamation allele numbers.

Error Detection in UNKNOWN

One of the principal purposes of UNKNOWN is to detect genetic errors in input pedigrees. A true incompatibility error occurs precisely when the specified genotypes are not consistent with Mendel's rules of inheritance. [Pedantic aside: I carefully wrote "Mendel's rules of inheritance" rather than "Mendelian inheritance" to emphasize that UNKNOWN looks at one locus at a time, and therefore, the effects of recombination, a phenomenon

UNKNOWN to Mendel, are irrelevant.] However, I have found that misformatted input can sometimes cause UNKNOWN to generate incompatibility errors, rather than reporting a formatting error. In particular, when UNKNOWN reports incompatibility errors for most of the individuals in a pedigree, this is usually a formatting error in `pedin.dat`.

In version 2.3P of FASTLINK, I improved UNKNOWN so that it would pinpoint the nuclear family in which an incompatibility occurs, provided the input pedigree is loopless. When multiple nuclear families are reported as erroneous, one can be assured only that the first one printed has an error. The others may have errors or may be propagated consequences of the first error. One may get UNKNOWN to print only the first error by changing the definition of the constant `ONE_ERROR_ONLY` from (the default) 0 to 1.

In version 3.0P of FASTLINK, UNKNOWN detects incompatibility errors in looped pedigrees, which was never done in previous versions. However, in looped pedigrees it only reports the problematic pedigree. To get a more detailed error diagnostic, use a copy of the pedigree file, change all entries in column 9 that are 2 or higher to 0, and rerun UNKNOWN with the modified pedigree file. Changing column 9 in this way has the effect of artificially eliminating all the loops (see loops.ps).

In version 2.3P of FASTLINK, I added an error checking diagnostic to detect most pedigrees with unbroken loops. Thanks to Frank Visser for suggesting this improvement. All previous versions of UNKNOWN would crash if the pedigree had an unbroken loop. Starting with FASTLINK 4.1P, UNKNOWN can break loops for you automatically and systematically (see README.lselect). I have left the diagnostic for unbroken loops in the code, but it should never be needed again.

Ken Morgan showed that my initial test for unbroken loops was insufficient, and a better diagnostic was put into FASTLINK 3.0P.

There is a dual flaw in LINKAGE and all versions of FASTLINK, through 3.0P. The programs will accept pedigrees that are disconnected because they have too many loop breakers. This flaw has been corrected implicitly, with no diagnostics, in FASTLINK 4.0P by the introduction of a sophisticated algorithm to choose loop breakers. See paper6.ps for much more information. Moreover, in FASTLINK 4.1, UNKNOWN can choose the initial loop breaker set for you (see README.lselect) so this error should not occur.

There are two standard routines called `inputerror` and `inputwarning` that list most of the input errors that can be detected. These routines are standard in the sense that they are shared by all the LINKAGE/FASTLINK programs. Even though `inputerror` lists over 40 different possible errors, only about 15 of these can occur in UNKNOWN. A detailed listing of all the error messages and what they mean is given in README.trouble.

In recent versions of UNKNOWN, the routine `respond` is called whenever an error is detected to ask the user if the run of UNKNOWN should continue. If the user wishes to continue, the user should press the ENTER key. Otherwise CTRL–C or whatever kills a process can be used to stop the run.

Loops

It was a well-kept secret that until FASTLINK 3.0P, UNKNOWN did no genotype inference or incompatibility detection for looped pedigrees. These flaws have been corrected, and the new code is used provided that the constant LOOPSPEED is set to 1. The algorithmic aspects are discussed in paper5.ps and loops.ps. Here we add a few salient points about how the new algorithms affect the usage of UNKNOWN.

UNKNOWN now outputs a file called `loopfile.dat`, which represents the results of its genotype inference. loopfile.dat is output regardless of whether any of the input pedigrees have loops. The syntax of `loopfile.dat` is described with an example in `README.loopfile`.

UNKNOWN is now faster on loopless pedigrees because some of the algorithmic improvements originally done in FASTLINK 1.0 have been incorporated into UNKNOWN. However, UNKNOWN may now take significantly longer than before on looped pedigrees because of the extra time needed to do genotype inference and incompatibility checking. On inputs for which UNKNOWN takes much longer than before, it is likely that a much larger amount of time will be saved in the main FASTLINK program.

The genotype inference for looped pedigrees uses space that grows exponentially with the number of loops. One can trade space for time by reducing the constant max_vectors_considered. When space is a problem, the program will output how much lower this constant needs to be to reduce the space usage. What the lower value means in effect is that one fewer loop will be considered in the genotype inference algorithm. This means that the fact that one of the loop breakers might not have all genotypes possible, will be ignored.

UNKNOWN now includes constants `LOOPSPEED` and `ALLELE_SPEED` that control whether the new loop algorithms are used and whether allele amalgamation is used. Both are set to 1 by default. The only reason I know of to change `LOOPSPEED` to 0 would be to see how much slower the code is without genotype inference for loops. The only two reasons I know of to change `ALLLELE_SPEED` to 0 would be to see how much slower the code is without allele amalgamation, and if one wants to do allele frequency estimation with ILINK. Both these constants appear in commondefs.h also. The new code enforces that they have the same settings in UNKNOWN and the main programs.

Versions of UNKNOWN

There seem to be a variety of versions of UNKNOWN in circulation. Joe Terwilliger has kindly pointed out to me that many of these versions are buggy. Some of the problems are discussed at length in the recently issued *Handbook of Human Genetic Linkage* by Terwilliger and Ott. They strongly recommend using versions prepared at Columbia after July 1993.

Part of the FASTLINK distribution is a C version of UNKNOWN. This is not really intended to be part of FASTLINK, but is distributed as a courtesy to FASTLINK users who want to completely avoid the need for a PASCAL compiler. Starting with version 2.2 of FASTLINK, the C version of UNKNOWN is based on the OS/2 PASCAL version from Columbia (following the above recommendation). I made it by using the p2c program to translate the PASCAL version to C and then applying some simple syntactic transformations to remove the need for the p2c library.

The newer versions of UNKNOWN have added some nice user-interface features. A diagnostic is now printed after each pedigree has been processed. More error checking has been added. When an error in the data is detected, the user is asked if the program should continue or not. Because of this need for user input, it is slightly dangerous to do your LINKAGE/FASTLINK runs in the background if you are not sure whether your input will pass through UNKNOWN without errors.

I added better error-checking tests in UNKNOWN with FASTLINK, 2.3P.

The UNKNOWN that comes with version 3.0P is drastically changed in many ways. To benefit from the new speedups in FASTLINK 3.0P (see paper5.ps), it is required to use the new UNKNOWN with the new main programs, and this is enforced syntactically.

The UNKNOWN that comes with version 4.0P an beyond adds new algorithms to select loop breakers. See README.lselect, loops.ps, paper6.ps, and paper7.ps for more details. The new code is in the file `loopbrk.c`, which was written mostly by Ann Becker and partly by me.

UNKNOWN from an Algorithmicist's Perspective

The significance of identifying possible genotypes for unknown individuals, is that in many cases the list of possible genotypes is quite small compared to the list of all genotypes. From the point of view of running time, making the list of possible genotypes as small as possible is crucial because it makes the essential arrays that encode the conditional probability of each genotype sparse. Even in cases where the genotype cannot be completely inferred (and it appears as unknown in `ipedfile.dat`), the list of possible genotypes in `speedfile.dat` can be extremely helpful in reducing computation in the main program. The algorithmic improvements in FASTLINK have redoubled the significance of sparsity for the impatient linkage analyst (see paper1.ps).

The UNKNOWN program tries to infer as much as possible for each unknown genotype of each individual. To do this it does a pedigree traversal with that individual as proband very much like the traversals in the main programs. See the FASTLINK document traverse.ps for more information on pedigree traversals. The traversals are orchestrated by a routine called `iterpeds`. To be similar to the main programs, each traversal for a (person, unknown genotype) pair is done by a routine called `likelihood`, even though no likelihood of anything is computed in the `likelihood` routine in UNKNOWN.

The main differences between the traversal in UNKNOWN and the regular traversals are that:

- Only one locus is considered at a time, hence the effects of recombination can be ignored, and the number of possible genotypes is usually small.
- Boolean logic is used to indicate which genotypes are possible rather than actually computing their conditional probabilities. That is, UNKNOWN makes no distinction between genotypes that could have different nonzero conditional probabilities in a traversal as done by the main programs.

The two points above account for why UNKNOWN can examine the same pedigrees much more quickly than LINKAGE/FASTLINK. The significance of using Boolean arithmetic instead of regular arithmetic is discussed at some length in paper1.ps.

UNKNOWN uses routines `collapsedown`, `collapseup`, `seg`, `segdown` and `segup` much like those in the original main LINKAGE programs. However, these `segup` and `segdown` routines are algorithmically much simpler because of the two points above.

One other distinction is that in UNKNOWN (unlike LINKAGE/FASTLINK), the same routines are used to handle both autosomal and sexlinked data.

When the traversal is done, the array of genotypes for proband is examined, to find which genotypes are possible. These are then stored in an array called `possible` to be printed later in the `writespeed` routine.

UNKNOWN in FASTLINK 4.0P includes an algorithmic solution to the problem of selecting loop breakers. This is a longstanding open problem in linkage analysis. It turns out that the linkage analysis version of the problem is a special case of a problem that has been studied in computer science since at least the mid-1980's. Ann Becker, Dan Geiger, and I have applied some of the techniques developed for the more general problem into the new version of UNKNOWN. See paper6.ps for the interesting history of the loop breaker selection problem and for information on how it can be solved algorithmically.