HAPLORE - A Program for Haplotype Reconstruction in General Pedigrees without Recombination Events and Genotyping Errors (Updated Version)

Kui Zhang, Fengzhu Sun, Hongyu Zhao

October 29, 2004

This document describes how to use HAPLORE, a program to infer haplotypes carried by each individual and to estimate the haplotype frequencies using the genotype data in general pedigrees. This program allows us to: (1) infer the complete and partial haplotypes carried by each individual with missing data imputation using a set of logic rules; (2) get all compatible haplotype configurations using the haplotype elimination algorithm; and (3) estimate the haplotype frequencies using the PL-EM algorithm and get all compatible haplotype configurations comprised by the haplotypes with frequencies greater than a threshold. In addition, a heuristic algorithm may be employed to find a minimum set of haplotypes that can give compatible haplotype configurations on the basis of complete haplotypes obtained by the set of logic rules in the first step. However, HAPLORE can only deal with simple pedigrees (the pedigrees without loop) without recombination events and genotyping errors among the specified marker loci. For the detailed description of those algorithms used in HAPLORE, please refer our paper (Zhang et al., 2003) and online documentation from our web site http://zhao.med.yale.edu/softwarelist.html.

If you use HAPLORE for published research, the appropriate citation is Zhang et al. (2003).

The Pre-Compiled Files

Two pre-compiled files, named as "HAPLORE_WIN.exe" and "HAPLORE_UNIX", are provided in the current stage. The first is intended to run on various Microsoft Windows operating systems, including Windows 95, Windows 98, Windows NT, Windows 2000, Windows ME and Windows XP. The other is intended to run on Unix, or Unix-like operating systems, such as Linux. After you download them on your computer and before you can run it on Unix or Unix-like operating systems, you may need to change its mode using the command "**chmod 777 HAPLORE_UNIX**".

The Command Line and The Arguments

The program runs under a command line. The command line "**HAPLORE –iTestGeno.dat**" shows the simplest usage of HAPLORE, by which the haplotypes carried by each individual are inferred using only the rule-based algorithm on the basis of the genotype data stored in the file named "TestGeno.dat". In addition, the program may find a minimum set of haplotypes that can give compatible haplotype configurations in

pedigrees using a simple heuristic method on the basis of haplotypes obtained by the logic rules. The file name of input data must appear in the command line after the switch "-i". The default file for output is "TestGeno.dat.out". When we use another command, "**HAPLORE –iTest.in –oTest.out –h1 –e1",** all tasks will be accomplished: (1) The program will infer the haplotypes carried by each individual using the logic rules; (2) The program will list all compatible haplotype configurations conformed to the genotype data and pedigrees for each individual; (3) The program will estimate the haplotype frequencies and list all compatible haplotypes with frequencies greater than a pre-specified threshold for each individual. The input and the output file are "Test.in" and "Test.out", respectively.

The arguments that can appear in the command line and their brief descriptions are listed in table 1:

Table 1: The arguments used in HAPLORE and their descriptions.

The Option	The Type	The Description	The Default
-i	A character string with length not greater than 60	The file name of the input data. It must appear in the arguments.	Does not have a default.
-o	A character string with length not greater than 60	The file name of output data.	The default is the concatenation of the input file name and the character string ".out". For example, if the input file name is "test", then the default file name for output is "test.out".
-h	A boolean variable	To indicate if we perform the haplotype elimination algorithm.	The default is 0 and not to perform the haplotype elimination algorithm.
-e	A boolean variable	To indicate if we perform the PL-EM algorithm to estimate the haplotype frequency.	The default is 0 and not to perform the PL-EM algorithm.
-m	A boolean variable	To indicate if we perform the heuristic algorithm to find a minimum set of haplotypes that can give compatible haplotype configurations for all families	The default is 0 and not to perform the heuristic algorithm
-b	A non-negative integer	The number of haplotypes with frequencies less than a pre-specified threshold will be retained in each EM step.	The default is 0.
-p	An integer in [3,9]	The length of each unit when we perform the haplotype elimination algorithm or the PL-EM algorithm.	The default is 4.
-a	An integer in {0,1,2}	The overlap between two adjacent units when we perform the haplotype elimination algorithm or the PL-EM algorithm.	The default is 1.
-t	A positive double number	The convergent criteria in the EM iteration.	The default is 10^{-5}.
-f	A positive double number	The threshold for the haplotype frequency. Only those haplotypes	The default is 10^{-6}.

		with frequencies greater than it are retained.	
-l	A boolean variable	To indicate if we list all possible haplotype configurations and their corresponding posterior probabilities in each family after the PL-EM algorithm.	The default is 0 and not to list all possible haplotype configurations in each family. This option is only valid after the PL-EM algorithm is performed (option -e1 is selected in the command line)

Input File

To perform the HAPLORE, you only need to give a genotype data file with pedigree structure. Here is a simple example of the input file:

```
10   3
3      1      0      0      1      1      1      1      1      2      0      1
3      2      0      0      2      0      1      2      1      2      1      2
3      3      1      2      1      1      1      1      1      2      1      1
3      4      1      2      1      1      1      1      1      2      1      1
12     11     0      0      1      1      1      1      1      2      0      1
12     22     0      0      2      0      1      2      1      2      1      2
12     23     11     22     1      1      1      1      1      2      0      1
12     44     11     22     1      1      1      1      1      2      0      1
7      5      0      0      1      0      2      2      2      1      2      2
8      3      0      0      2      0      1      1      1      2      1      1
```

The first two positive integers represent the number of individuals in the whole pedigree and the number of marker loci, respectively. Then the following records in the file give the information and the genotype of each individual in the pedigree. For each individual, the family ID, the individual ID, the father ID, the mother ID, the gender and the disease status are given first, followed by the two alleles at each locus successively. The non-negative integer is used to represent such information in all fields. In the gender filed, 1 and 2 represents the male and the female, respectively. In the disease field, 1 and 0 represents the affected and the unaffected status, respectively. Although we do not use disease information to construct haplotype, to be consistent with other general genotype data file used in other linkage or linkage-like software, we still keep this filed in the file. At each marker locus, a positive number of represents an allele and 0 represents the missing data. But all alleles at a marker locus must be recoded as consecutive integers staring from 1. No matter how many families in pedigrees and the order of individuals in this file, HAPLORE will correctly get the proper pedigree structure and genotype data from file. The population samples are very helpful to improve the estimates for the haplotype frequencies and can be used in this file. At this situation, each individual has a unique family ID, a non-negative individual ID and is the only founder member in this family. In this example, the first four individuals form family 1 (with family ID 3) and the first and second individuals are the father and the mother of individual 3 and 4, respectively. The individual 9 and 10 (with family ID 7 and 8 and individual ID 5 and 3, respectively) are from a population. They are the only founders in family 7 and 8, respectively. This example also illustrate that the family ID and the individual ID are not necessary to be labeled by consecutive integers.

Output File

Depending on the arguments specified in the command line, the results can be divided into up to four parts. In the following, we will give illustrations for each part based on the output file obtained using the command line "HAPLORE –IExamp.in –OExamp.out –H1 –E1 -L1" with the above data.

The first part of the results is the haplotypes deducted by the logic rules and looks like this:

The haplotypes obtained by the logic rules are:
3   1   0   0   1   1
a: 1 1 0
a: 1 2 0
g: 1 1 1
g: 1 2 0
3   2   0   0   2   0
a: 1 -1 1
a: 2 -1 2
g: 1 1 1
g: 2 2 2
......

For each individual, the family ID, the people ID, the father ID, the mother ID, the gender and the disease status are listed first. Then the two haplotypes carried by each individual, followed the genotype of this individual as a comparison are listed. The symbol "a", "f" and "m" represent this haplotype is anonymous, from the father and from the mother, respectively. The haplotype can be complete or partial, in which the missing data (unassigned allele) is represented by "0" and "-1". However, if "-1" is in one of the haplotype, it must appear in the other haplotype and the genotype for this individual at this locus is complete and heterozygous. The genotype may be different from the original genotype data because some imputations could be made by this procedure. In the genotype, "0" is used to represent the missing data.

The other three parts of the results are very similar and look like this:

The haplotypes obtained by the heuristic method are:

3   1   0   0   1   1
Total number of haplotype pairs is: 3
Haplotype Pair   1 is: 1   1
Haplotype Pair   2 is: 1   3
Haplotype Pair   3 is: 2   4

3   2   0   0   2   0
Total number of haplotype pairs is: 2
Haplotype Pair   1 is: 1   5
Haplotype Pair   2 is: 2   6

......

8   3   0   0   2   0
Total number of haplotype pairs is: 1
Haplotype Pair   1 is: 1   2

The total of 8 haplotypes are:
haplotype   1 :   1   2   1   13.00000000
haplotype   2 :   1   1   1   13.00000000
haplotype   3 :   1   2   2   5.00000000
haplotype   4 :   1   1   2   5.00000000
……

Similar with the results for the first part, the family ID, the people ID, the father ID, the mother ID, the gender and the disease status for each individual are listed first. Then all the compatible haplotype pairs are given. We only give the code of haplotypes and do not distinguish the origin of haplotypes. So the haplotype pairs (1,2) and (2,1) are considered as same pairs. The corresponding haplotypes and their frequencies (if the PL-EM algorithm is used) are given at the end of this part.

The last part of the output is a list of all possible haplotype configurations and their posterior probabilities in each family. This part of results looks like this:

All possible haplotype configurations:

Family ID is: 3
The number of haplotype configurations in this family is: 4
The number of individuals in this family is: 4
The number of founders in this family is: 2
The number of non-founders in this family is: 2
ConfigID   1   2   3   4
**   1 (1, 2) (1, 5) (1, 2) (1, 2) 0.499998319**
**   2 (1, 2) (2, 6) (1, 2) (1, 2) 0.499998319**
**   3 (1, 3) (2, 6) (1, 2) (1, 2) 0.000001681**
**   4 (2, 4) (1, 5) (1, 2) (1, 2) 0.000001681**
……

In each family, it will report number of haplotype configurations, the number of individuals, the number of founders, and the number of non-founders in this family. Then it will give the people ID for each founder, followed by the people ID of non founders. At last, it will list all possible haplotype configurations. In the above example, the family "3" has 4 possible haplotype configurations. There are two founders (individuals 3, 4) and two non-founders (individuals 3, 4). One of the haplotype configurations is (1, 2), (1,5), (1,2) and (1,2) for the individual 1, 2, 3, and 4 in family "3", respectively.

Remarks

In this section, we highlight some possible solutions for the problems that may be encountered in running HAPLORE.

- When you specify the arguments in the command line, there is no space between the switch and the argument and no difference between the lowercase and the

uppercase in the switch. Thus, the command line "HAPLORE –iTest" and the command line "HAPLORE –ITest" will achieve the same goal.

- The alleles at a marker locus must be recoded as consecutive integers from 1. For example, if there are 4 alleles at a marker locus, then they must be labeled as 1, 2, 3 or 4.

- We assume that there are no recombination events and genotyping errors among the specified marker loci for HAPLORE. The program can ignore the families with genotyping errors or recombination events to infer haplotypes using the logic rules. However, the families with genotyping errors or recombination events must be excluded from the input file before performing the heuristic algorithm, the haplotype elimination algorithm, or the PL-EM algorithm.

- The program often fails in the heuristic algorithm for finding a minimum set of haplotypes that can give compatible haplotype configurations in pedigrees. The family ID and the individual ID for those individuals without compatible haplotypes are reported on the screen. You can exclude those families from the input file to rerun the program and to get a feasible solution.

- The program is terminated when there are no compatible haplotypes for an individual in performing the haplotype elimination algorithm. The family ID and the individual ID are reported on the screen. You can exclude the reported family from the input file to rerun the program again.

- The program is terminated when there are no compatible haplotypes for an individual in performing the PL-EM algorithm. The family ID and the individual ID is reported on the screen. You can (1) exclude the reported family from the input, or (2) increase the number in the option "-b", or (3) give a smaller number in the option "-f", to rerun the program again.

- The haplotype elimination algorithm can be very time-consuming. It can fail due to too many compatible haplotypes in the pedigree, which often happens in large pedigrees for a large number of loci with missing data. If this occurs, we recommend you try to perform it in small pedigrees for a small number of markers.

- The PL-EM algorithm is generally very computationally extensive. The running time is also affected by several arguments. Therefore, you may speed up it by: (1) using small number in option "-b"; (2) giving a large number in the option "-f"; (3) using different numbers in the option "-p" and "-a"; (4) specifying a large number in "-t".

- The options (-l) to list all possible haplotype configurations and their posterior probabilities is only valid after the PL-EM algorithm is performed.

Contact Information

This program and related materials can be downloaded through the following web site:

http://zhao.med.yale.edu/softwarelist.html

Bugs, comments or the request of source codes should be reported to:

Hongyu Zhao
Department of Epidemiology and Public Health
Yale University School of Medicine
60 College Street, New Haven, CT 06520-8034
Tel:(203) 785-6271
Fax: (203) 785-6912
E-Mail: hongyu.zhao@yale.edu

References

Kui Zhang, Fengzhu Sun, Hongyu Zhao. 2003. HAPLORE - A program for haplotype reconstruction in General Pedigrees. In preparation.