The Stata Journal (2010)
10, Number 3, pp. 359–368

Using Stata with PHASE and Haploview: Commands for importing and exporting data

J. Charles Huber Jr.
Department of Epidemiology and Biostatistics
Texas A&M Health Science Center School of Rural Public Health
College Station, TX
jchuber@srph.tamhsc.edu

Abstract. Modern genetics studies require the use of many specialty software programs for various aspects of the statistical analysis. PHASE is a program often used to reconstruct haplotypes from genotype data, and Haploview is a program often used to visualize and analyze single nucleotide polymorphism data. Three new commands are described for performing these three steps: 1) exporting genotype data stored in Stata to PHASE, 2) importing the resulting inferred haplotypes back into Stata, and 3) exporting the haplotype/single nucleotide polymorphism data from Stata to Haploview.

Keywords: st0199, phaseout, phasein, haploviewout, genetics, haplotypes, SNPs, PHASE, haploview

1 Introduction

For a variety of reasons, including favorable power for detecting small effects and the low cost of genotyping, association studies based on single nucleotide polymorphism (SNP, pronounced "snip") markers have become common in genetic epidemiology (Cordell and Clayton 2005). SNP markers are positions along a chromosome that can have four forms called alleles: adenine, cytosine, guanine, and thymine, which are denoted A, C, G, and T, respectively. Humans are diploid organisms, meaning that we have two copies of each of our chromosomes; thus each SNP is composed of a pair of alleles called a genotype.

For example, a SNP might have an adenine (A) molecule on one chromosome paired with a cytosine (C) molecule on the other chromosome. This is often described as an A/C genotype. When two SNP markers are physically close to one another, a pair of alleles found on the same chromosome forms a haplotype. For example, a person might have an A/C genotype for SNP1 and a G/T genotype for SNP2. If the A allele from SNP1 and the G allele from SNP2 are physically located on the same chromosome, they are said to form an AG haplotype. Similarly, the C allele from SNP1 and the T allele from SNP2 would form a CT haplotype.

It has been shown that association studies based on haplotypes are often more powerful than similar studies based on individual SNPs (Akey, Jin, and Xiong 2001). Unfortunately, haplotypes are not observed directly using typical low-cost, high-throughput laboratory techniques. However, haplotypes can be inferred statistically based on the observed genotypes.

David G. Clayton of the University of Cambridge has written a useful command for Stata (`snp2hap`) that infers haplotypes for pairs of SNPs. In theory, this program could be used iteratively to infer haplotypes across many SNPs. However, several sophisticated algorithms have been developed for statistically inferring haplotypes from many SNP genotypes simultaneously. These algorithms and the software that implement them have been reviewed and compared elsewhere (Marchini et al. 2006; Stephens and Donnelly 2003; The International HapMap Consortium 2005). In most comparisons, the algorithm used in the PHASE program (Stephens, Smith, and Donnelly 2001) was found to be the most accurate and is arguably the most frequently used.

Rather than attempt the daunting task of creating a Stata command to implement the algorithm used in PHASE, a Stata command (`phaseout`) was developed for exporting genotype data stored in Stata to an ASCII file formatted as a PHASE input file. A second program (`phasein`) was developed to import the inferred haplotype data back into Stata for subsequent association analyses with programs such as `haplologit` (Marchenko et al. 2008). These commands use a group of Stata's low-level file commands including `file open`, `file write`, `file read`, and `file close`.

Once the haplotypes have been inferred for a set of genotypes, one would often like to know certain attributes of the haplotypes. For example, the alleles of some pairs of SNPs along a haplotype may tend to be transmitted together from parent to offspring more frequently than alleles of other pairs of SNPs. This phenomenon, known as linkage disequilibrium (Devlin and Risch 1995), is often quantified by the r^2 or D' statistics. Similarly, some contiguous groups of SNPs often called haplotype blocks, may exhibit high levels of pairwise linkage disequilibrium (Gabriel et al. 2002; Goldstein 2001). High levels of linkage disequilibrium between two SNPs indicate that much of their statistical information is redundant, so both SNPs are not necessary for association analyses. One of the SNPs, called a tagSNP (Zhang et al. 2004), can be selected using one of several algorithms. A tagSNP can be used in place of the group of redundant SNPs. Typically, there are several tagSNPs in a group of contiguous SNPs found on a chromosome.

Haploview (Barrett et al. 2005) is a popular software package used for calculating and visualizing the linkage disequilibrium statistics r^2 and D', as well as for identifying haplotype blocks and tagSNPs. The new Stata `haploviewout` program exports haplotype data from Stata to a pair of ASCII files formatted as Haploview input files: a haps format data file and a haps format locus information file.

The dataset used for the following examples was downloaded from the SeattleSNPs website (SeattleSNPs 2009) and was modified to include missing data. Genotypes for 47 individuals of African and European descent include 22 SNPs from the vascular endothelial growth factor (VEGF) gene located on chromosome six.

2 The phaseout command

Genotype data stored in Stata are often formatted in a way that is similar to the following example. In this example, the variable `id` contains individual identification numbers, and the variables `rs1413711`, `rs3024987`, and `rs3024989` contain data on three SNPs. The genotype `X/X` indicates that the genotype is missing. The following example uses fictitious data:

```
. list id rs1413711 rs3024987 rs3024989 in 1/2
```

	id	rs1413711	rs3024987	rs3024989
1.	D001	C/C	C/T	T/T
2.	D002	C/T	X/X	T/T

The input file for PHASE requires the data to be formatted in an ASCII file that contains header information about the number of samples and the number and types of markers (SNP or multiallelic), as well as the actual data:

```
47                (There are 47 samples in the entire file.)
3                 (There are three markers in the file.)
P 674 836 1955    (Positions are listed.)
SSS               (All three markers are biallelic SNPs.)
D001              (The data begin with the first ID.)
 C C T            (The genotype data are stored in two rows.)
 C T T            (These are not haplotypes yet.)
D002              (The data begin with the second ID.)
 C ? T            (The missing SNP data are
 T ? T              stored as question marks.)
```

The `phaseout` command calculates the header information, converts the ID and genotype data to rows, and writes this data to the ASCII file. The types of markers—SNPs or multiallelic markers—are automatically determined by tabulating the genotypes and by examining the length of the genotype in the first record. If a marker has three or fewer genotypes (for example, `C/C`, `C/T`, `T/T`) and the length of the genotype in the first record is fewer than five alleles, the marker is treated as a SNP. All other markers are treated as multiallelic.

2.1 Syntax

`phaseout` *SNPlist*`,` `idvariable(`*varname*`)` `filename(`*filename*`)` [`missing(`*string*`)` `separator(`*string*`)` `positions(`*string*`)`]

SNPlist is a list of variables containing SNP genotypes.

2.2 Options

`idvariable(`*varname*`)` is required to specify the variable that contains the individual identifiers.

`filename(`*filename*`)` is required to name the ASCII file that will be created. It is conventional, though not necessary, to name PHASE input files with the extension `.inp`.

`missing(`*string*`)` may be used to provide a list of genotypes that indicate missing data. For example, missing data might be included in the dataset as `X/X` for SNPs and as `999/999` for multiallelic markers. Multiple missing values may be specified by placing a space between them (for example, `missing("X/X 9/9 999/999")`). PHASE requires missing SNP alleles to be coded as "?" and missing multiallelic alleles to be coded as "−1". It is not necessary to preprocess your data because `phaseout` will automatically convert each genotype contained in the `missing()` list to its appropriate PHASE missing value.

`separator(`*string*`)` specifies the separator to use when storing genotype data. Genotype data are often stored with a separator between the two alleles. For data stored in the format `C/G`, the `separator()` option would look like `separator("/")`. If SNP data are stored without a separator (for example, `CG`) then the `separator()` option is unnecessary, and `phaseout` will assume that the left character is allele 1 and the right character is allele 2.

`positions(`*string*`)` provides a list of the marker positions for use by PHASE when inferring haplotypes from the genotype data. If the `positions()` option is not specified, PHASE will assume that the markers are equally spaced.

2.3 Output files

`phaseout` saves two ASCII files for subsequent use by the commands `phasein` and `haploviewout`:

- `MarkerList.txt` contains a space-delimited list of marker names.
- `PositionList.txt` contains a space-delimited list of marker positions.

2.4 Examples

Markers and positions may be specified in the command itself:

```
. phaseout rs1413711 rs3024987 rs3024989, idvariable("id") filename("VEGF.inp")
> missing("X/X 9/9") positions("674 836 1955") separator("/")
```

`phaseout` may use markers and positions saved in local macros:

```
. local SNPList "rs1413711 rs3024987 rs3024989 rs833068 rs3024990"
. local PositionsList "674 836 1955 2523 3031"
. phaseout `SNPList´, idvariable("id") filename("VEGF.inp") missing("X/X 9/9")
> positions(`PositionsList´) separator("/")
```

3 The phasein command

PHASE saves the inferred haplotypes for each pair of chromosomes in a file with the extension `.out`, and because there is a great deal of other information saved in the file, the `phasein` command uses the keywords `BEGIN BESTPAIRS1` and `END BESTPAIRS1` to identify the part of the file that contains the haplotypes:

```
BEGIN BESTPAIRS1
0 D001
C C T
C T T
......
......
0 E023
C C T
C C T
END BESTPAIRS1.
```

The data are imported into Stata in "long" format with one row per chromosome (two rows per ID). The haplotypes are imported into a variable named `haplotype`, and each of the markers that make up the haplotype are saved in an individual variable. If the `markers()` option is specified, the marker variables will be renamed using their original names.

```
. list id haplotype rs1413711 rs3024987 rs3024989 in 1/2
```

	id	haplotype	rs1413711	rs3024987	rs3024989
1.	D001	CCT	C	C	T
2.	D001	CTT	C	T	T

If the `positions()` option is used, the positions will be placed in the variable label of each marker variable:

(*Continued on next page*)

```
. describe
Contains data from VEGF_Haplotypes.dta
  obs:            94
 vars:             5                          8 Jul 2010 13:09
 size:         1,692 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
id              str4   %9s
haplotype       str3   %9s
rs1413711       str1   %9s                    position=674
rs3024987       str1   %9s                    position=836
rs3024989       str1   %9s                    position=1955
-------------------------------------------------------------------------------
Sorted by:
```

3.1 Syntax

`phasein` *PhaseOutputFile* [, `markers(`*filename*`)` `positions(`*filename*`)`]

PhaseOutputFile is the name of the PHASE output file that contains the inferred haplotypes. It will have the file extension `.out`.

3.2 Options

`markers(`*filename*`)` allows the user to specify an ASCII file that contains the names of the markers included in the haplotype. If the original genotype data were exported to PHASE using the `phaseout` command, the marker names will be automatically saved to a file named `MarkerList.txt`. If that is the case, then the option would be `markers("MarkerList.txt")`. Alternatively, the user can save a space-delimited list of marker names in an ASCII file and use the `markers("`*filename*`.txt")` option.

`positions(`*filename*`)` allows the user to specify an ASCII file that contains the positions of the markers. If the original genotype data were exported to PHASE using the `phaseout` command, the marker positions will be automatically saved to a file named `PositionList.txt`. If that is the case, then the option would be `positions("PositionList.txt")`. Alternatively, the user can save a space-delimited list of marker positions in an ASCII file and use the `positions("`*filename*`.txt")` option.

3.3 Examples

Using the default files created by `phaseout`:

```
. phasein VEGF.out, markers("MarkerList.txt") positions("PositionList.txt")
```

Using the files created by the user:

```
. phasein VEGF.out, markers("UserMarkerList.txt") positions("UserPositionList.txt")
```

4 The haploviewout command

The `haploviewout` command exports haplotype data from Stata to a pair of files. The file `Filename_DataInput.txt` contains the marker data for each individual, with the alleles recoded as follows: missing = 0, A = 1, C = 2, G = 3, and T = 4.

```
D001 D001  2 2 4
D001 D001  2 4 4
D002 D002  2 2 4
D002 D002  4 2 4
```

The file `Filename_MarkerInput.txt` contains the marker names and positions in two columns:

```
rs1413711   674
rs3024987   836
rs3024989   1955
```

4.1 Syntax

`haploviewout` *SNPlist*, `idvariable(`*varname*`) filename(`*filename*`)`
`[positions(`*string*`) familyid(`*variable*`) poslabel]`

SNPlist is a list of SNP variables in long format (that is, one row per chromosome). If your data are in wide format, you can convert them to long format by using the `reshape` command.

Haploview will not accept multiallelic markers.

4.2 Options

`idvariable(`*varname*`)` is required to specify the variable that contains the individual identifiers.

`filename(`*filename*`)` is required to name the two ASCII files that will be created. Those files will have the extensions `_DataInput.txt` and `_MarkerInput.txt` appended to *filename*. For example, the `filename("VEGF")` option will create a file named `VEGF_DataInput.txt` and a file named `VEGF_MarkerInput.txt`. To open the files in Haploview, select **File** > **Open new data** and click on the tab labeled *Haps Format*. Click on the Browse button next to the box labeled *Data File* and select the file `VEGF_DataInput.txt`. Next click on the Browse button next to the box labeled *Locus Information File* and select the file `VEGF_MarkerInput.txt`.

`positions(`*string*`)` allows the user to specify a space-delimited list of the marker positions.

`familyid(`*variable*`)` allows the user to specify the variable that contains family identifiers if relatives are included in the dataset. If `familyid()` is omitted, the `idvariable()` will be automatically substituted for the `familyid()`.

`poslabel` will automatically extract the SNP positions from the variable label of each SNP if the haplotype data were created using the commands `phaseout` and `phasein`. The positions for each marker are stored in the variable label of each SNP.

4.3 Examples

Using the default files created by `phaseout`:

```
. phaseout rs1413711 rs3024987 rs3024989, idvariable("id") filename("VEGF.inp")
> missing("X/X 9/9") positions("674 836 1955") separator("/")
. phasein VEGF.out, markers("MarkerList.txt") positions("PositionList.txt")
. haploviewout rs1413711 rs3024987 rs3024989, idvariable(id) filename("VEGF")
> poslabel
```

Using the files created by the user:

```
. haploviewout rs1413711 rs3024987 rs3024989, idvariable(id) filename("VEGF")
> positions("674 836 1955")
```

5 Discussion

Many young and rapidly evolving fields of inquiry, including genetic association studies, use a variety of boutique software packages. While it would be very convenient to have Stata commands that accomplish the same tasks, the time and programming expertise required does not make this a practical option. However, a suite of commands that allows easy exporting and importing of data from Stata to other specialized software seems to be an efficient way for Stata users to accomplish specialized analytical tasks.

6 Acknowledgments

This work was supported in part by grant 1 R01 DK073618-02 from the National Institute of Diabetes and Digestive and Kidney Diseases and by grant 2006-35205-16715 from the United States Department of Agriculture. The author would like to thank Drs. Loren Skow, Krista Fritz, and Candice Brinkmeyer-Langford of the Texas A&M College of Veterinary Medicine and Roger Newson of the Imperial College London for their very useful feedback.

7 References

Akey, J., L. Jin, and M. Xiong. 2001. Haplotypes vs single marker linkage disequilibrium tests: What do we gain? *European Journal of Human Genetics* 9: 291–300.

Barrett, J. C., B. Fry, J. Maller, and M. J. Daly. 2005. Haploview: Analysis and visualization of LD and haplotype maps. *Bioinformatics* 21: 263–265.

Cordell, H. J., and D. G. Clayton. 2005. Genetic association studies. *Lancet* 366: 1121–1131.

Devlin, B., and N. Risch. 1995. A comparison of linkage disequilibrium measures for fine-scale mapping. *Genomics* 29: 311–322.

Gabriel, S. B., S. F. Schaffner, H. Nguyen, J. M. Moore, J. Roy, B. Blumenstiel, J. Higgins, M. DeFelice, A. Lochner, M. Faggart, S. N. Liu-Cordero, C. Rotimi, A. Adeyemo, R. Cooper, R. Ward, E. S. Lander, M. J. Daly, and D. Altshuler. 2002. The structure of haplotype blocks in the human genome. *Science* 296: 2225–2229.

Goldstein, D. B. 2001. Islands of linkage disequilibrium. *Nature Genetics* 29: 109–111.

Marchenko, Y. V., R. J. Carroll, D. Y. Lin, C. I. Amos, and R. G. Gutierrez. 2008. Semiparametric analysis of case–control genetic data in the presence of environmental factors. *Stata Journal* 8: 305–333.

Marchini, J., D. Cutler, N. Patterson, M. Stephens, E. Eskin, E. Halperin, S. Lin, Z. S. Qin, H. M. Munro, G. R. Abecasis, P. Donnelly, and The International HapMap Consortium. 2006. A comparison of phasing algorithms for trios and unrelated individuals. *American Journal of Human Genetics* 78: 437–450.

SeattleSNPs. 2009. NHLBI Program for Genomic Applications. http://pga.gs.washington.edu.

Stephens, M., and P. Donnelly. 2003. A comparison of Bayesian methods for haplotype reconstruction from population genotype data. *American Journal of Human Genetics* 73: 1162–1169.

Stephens, M., N. J. Smith, and P. Donnelly. 2001. A new statistical method for haplotype reconstruction from population data. *American Journal of Human Genetics* 68: 978–989.

The International HapMap Consortium. 2005. A haplotype map of the human genome. *Nature* 437: 1299–1320.

Zhang, K., Z. S. Qin, J. S. Liu, T. Chen, M. S. Waterman, and F. Sun. 2004. Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies. *Genome Research* 14: 908–916.

About the author

Chuck Huber is an assistant professor of biostatistics at the Texas A&M Health Science Center School of Rural Public Health in the Department of Epidemiology and Biostatistics. He works on projects in a variety of topical areas, but his primary area of interest is statistical genetics.