BEAGLE 3.0

Brian L. Browning

Department of Statistics

The University of Auckland

Auckland

New Zealand

11 April 2009

Contents

1 Introduction

BEAGLE is a software program for imputing genotypes, inferring haplotype phase, and performing genetic association analysis. BEAGLE is designed to analyze large-scale data sets with hundreds of thousands of markers genotyped on thousands of samples. BEAGLE can

- phase genotype data (i.e. infer haplotypes) for unrelated individuals, parent-offspring pairs, and parent-offspring trios.
- infer sporadic missing genotype data.
- impute ungenotyped markers that have been genotyped in a reference panel.
- perform single marker and haplotypic association analysis.

BEAGLE 3.0 has two new capabilities: phasing of parent-offspring data (pairs or trios) and imputation of ungenotyped markers using a reference panel. With BEAGLE 3.0, you can mix-and-match four different kinds of data: phase-unknown genotype data, phase-known haplotype data, parent-offspring trio data, and parent-offspring pair data. BEAGLE will infer haplotypes and impute missing genotypes and ungenotyped markers for each kind of data.

In order to accommodate this additional capability, the command line and output files for BEAGLE 3.0 are slightly different than those for earlier versions. The BEAGLE input file format has also been expanded to accommodate parent-offspring data.

Beagle can be used in tandem with the PRESTO software package. PRESTO can compute empirical distributions of order statistics, analyze stratified data, and determine significance levels for one and two-stage genetic association studies. PRESTO is freely available from www.stat.auckland.ac.nz/~browning/presto/presto.html.

BEAGLE is written in Java and runs on most computing platforms (e.g. Windows, Unix, Linux, Solaris, and Mac). A java interpreter is probably already installed on your computer (type java -version at the command line prompt to check). However, if it is not installed or if it is not version 1.5 or later, the current java interpreter can be downloaded free of charge from the **java.sun.com** web site. The java interpreter is called the Java Standard Edition (SE) Runtime Environment (JRE).

1.1 Citing BEAGLE

If you use BEAGLE and publish your analysis, please report the version of the program used and cite the appropriate publication or publications given below.

1. BEAGLE's methods for imputing ungenotyped markers and phasing parent-offspring data are described in

   B L Browning and S R Browning (2009) A unified approach to genotype imputation and haplotype phase inference for large data sets of trios and unrelated individuals. Am J Hum Genet 84:210-223.

2. BEAGLE's methods for inferring haplotype phase and sporadic missing data are described in

S R Browning and B L Browning (2007) Rapid and accurate haplotype phasing and missing data inference for whole genome association studies using localized haplotype clustering. Am J Hum Genet 81:1084-1097.

3. BEAGLE's methods for association testing are described in

   Browning and Browning (2007) Efficient multilocus association testing for whole genome association studies using localized haplotype clustering. Genet Epidemiol 31:365-375.

4. BEAGLE's haplotype frequency model was first described in

   S R Browning (2006) Multilocus association mapping using variable-length Markov chains. Am J Hum Genet 78:903-13.

1.2 Files in the BEAGLE software distribution

BEAGLE is freely available and can be downloaded from the BEAGLE web site:

www.stat.auckland.ac.nz/~browning/beagle/beagle.html

The BEAGLE software distribution includes the following files and folders:

- beagle.jar - the BEAGLE 3.0 executable file.
- beagle_3.0.pdf - the BEAGLE 3.0 documentation.
- divide.sample - a unix shell script for dividing a sample and performing imputation in each subsample separately.
- example - a folder containing input and output files for two BEAGLE analyses described in Section 6 Example BEAGLE analyses.
- utility - a folder containing seven utility programs which are described in Section 7.1 Utility programs:
  - linkage2beagle.jar - a program for creating an unphased Beagle file from a data file and a pedigree file in linkage or QTDT format.
  - phased2beagle.jar - a program for creating a phased Beagle file from an output file of a haplotype phasing program.
  - cut.jar - a program for extracting columns of data from Beagle files.
  - paste.jar - a program for combining Beagle files that have disjoint sets of individuals and data for the same variables (e.g. phenotypes and genetic markers).
  - splitbeagle.jar - a program for splitting a Beagle file into two Beagle files according to the value of a variable in the Beagle file.
  - pseudomarker.jar - a program for creating a phased Beagle file of pseudomarkers from a Beagle output model (.dag) file.
  - cluster2haps.jar - a program for identifying the allele sequences that define a haplotype cluster that is associated with a trait.

2 Input file format

Input text files can be compressed with the gzip algorithm. BEAGLE assumes that any file that has a name ending in ".gz" is compressed with gzip.

2.1 Beagle file format

Beagle input files have a simple format: rows are variables and columns are individuals. Here is an example of a Beagle file with three individuals and three genotyped markers:

Example 1 - Sample Beagle file

```
I    id          1001  1001  1002  1002  1003  1003

A    diabetes    1     1     2     2     2     2
M    rs2289311   A     G     G     G     A     G
M    rs1248628   T     T     T     C     T     T
M    rs10762764  G     T     T     T     G     T
```

In a Beagle file, the first column describes the data on each line. The fields in the first column are typically single characters, but this is not required. The second column contains the name of the variable whose data is given on each line. Variable names should be unique. In Example 1 there are two columns for each individual: columns 3-4 give data for the first individual, columns 5-6 give data for the second individual, and so on.

In the Beagle file in Example 1, the first line (I ....) is called an identifier line and gives an identifier for each column of data. The identifiers are not required to be distinct, and you will typically use the identifier for both columns of each diploid individual. The second line (A ...) is called an affection status line and gives an affection status (1 = unaffected, 2 = affected) for each individual. The last three lines (M ...) are marker lines that give marker alleles for the three markers (rs2289311, rs1248628, and rs10762764). Note that an identifier and an affection status are given for each allele (column). For diploid data, the identifier and affection status will typically be the same for both alleles.

In the Beagle file in Example 1, the three individuals have identifiers 1001 (columns 3-4), 1002 (columns 5-6), and 1003 (columns 7-8). The first individual (columns 3-4) is unaffected, and the second and third individuals (columns 5-6 and 7-8) are affected.

A BEAGLE file can have either unphased data or phased data for unrelated individuals, parent-offspring trios, or parent-offspring pairs. Here is how each type of data is represented in a Beagle file:

- **Unphased unrelated data**. Each pair of columns (beginning with columns 3-4) gives the genotype for each unrelated individual. If the Beagle file in Example 1 contains unphased data for unrelated individuals, the first individual has genotypes A/G, T/T, G/T, the second individual has genotypes G/G, C/T, T/T, and the third individual has genotypes A/G, T/T, G/T for markers rs2289311, rs1248628, and rs10762764 respectively. Unphased unrelated data is specified with the `unphased` command line argument (see Section 3.2.1 Arguments for specifying).
- **Phased unrelated data**. Each column (beginning with column 3) gives a phased haplotype, and for diploid data, each pair of columns gives the pair of phased haplotypes for each diploid individual. If the Beagle file in Example 1 contains phased data for unrelated individuals, then the first individual has haplotypes ATG and GTT, the second

individual has haplotypes equal to GTT and GCT, and the third individual has haplotypes ATG and GTT for markers rs2289311, rs1248628, and rs10762764 respectively. When Beagle is used to phase genotype data for unrelated individuals, the input Beagle file contains *unphased* unrelated data, and the output Beagle file contains *phased* unrelated data. Phased unrelated data files are specified with the phased command line argument (see Section 3.2.1 Arguments for specifying).

- **Unphased trio data**. Each set of six consecutive columns (beginning with columns 3-8) gives the genotype data for one parent-offspring trio. In each set of six columns, the first two columns give the genotypes for the first parent, the middle two columns give the genotypes for the second parent, and the last two columns give the genotypes for the offspring. If the Beagle file in Example 1 contains unphased trio data, the first parent has identifier 1001 (columns 3-4), the second parent has identifier 1002 (columns 5-6), and the child has identifier 1003 (columns 7-8). Unphased trio data files are specified with the trios command line argument (see 3.2.1 Arguments for specifying). Unphased trio data is not permitted to have any Mendelian inconsistencies. Genotypes with Mendelian inconsistencies for a trio must be replaced with missing genotypes.
- **Phased trio data**. Each set of four consecutive columns (beginning with columns 3-6) gives the transmitted and untransmitted haplotypes for one parent-offspring trio. In each set of four columns, the first column is the first parent's transmitted haplotype, the second column is the first parent's untransmitted haplotype, the third column is the second parent's transmitted haplotype, and the fourth column is the second parent's untransmitted haplotype. If the Beagle file in Example 1 contains unphased trio data, then one can tell by inspection that the first parent transmits the ATG haplotype and the second parent transmits the GTT haplotype. Thus in the preceding example, the corresponding phased trio file is obtained by deleting the offspring data (sample id 1003) in columns 7-8. When Beagle is used to phase genotype data for parent-offpspring trios, the input Beagle file contains *unphased* trio data, and the output Beagle file contains *phased* trio data. Phased offspring genotypes can be reconstructed from the first column of each parent in the phased trio data.
- **Unphased pair data**. Each set of four consecutive columns (beginning with columns 3-6) gives the genotype data for one parent-offspring pair. In each set of four columns, the first two columns give the genotypes for the genotyped parent, and the last two columns give the genotypes for the offspring. If the Beagle file in Example 2 below contains unphased pair data, the genotyped parent has identifier 1001 (columns 3-4), and the offspring has identifier 1002 (columns 5-6). Unphased pair data files are specified with the pairs command line argument (see 3.2.1 Arguments for specifying). Unphased pair data is not permitted to have any Mendelian inconsistencies. Genotypes with Mendelian inconsistencies for a pair must be replaced with missing genotypes.
- **Phased pair data**. Each set of three consecutive columns (beginning with columns 3-5) gives the transmitted and untransmitted haplotypes for one parent-offspring pair. In each set of three columns, the first column is the genotyped parent's transmitted haplotype, the second column is the genotyped parent's untransmitted haplotype, and the third column is the ungenotyped parent's transmitted haplotype. If the Beagle file in Example 2 below contains unphased pair data, then one can tell by inspection that the genotyped parent transmits the GTT haplotype. Thus the corresponding phased pair file is obtained by

deleting the last column in Example 2. When Beagle is used to phase genotype data for parent-offspring pairs, the input Beagle file contains *unphased* pair data, and the output Beagle file contains *phased* pair data. Phased offspring genotypes can be reconstructed from the columns containing the transmitted haplotypes from the genotyped and ungenotyped parents.

Example 2 - Sample Beagle file with parent-offspring pair data

```
I       id          1001   1001   1002   1002
A       diabetes    1      1      2      2
M       rs2289311   G      A      G      G
M       rs1248628   T      T      T      C
M       rs10762764  T      G      T      T
```

BEAGLE imposes very few constraints on your data files:

- ❖ Input data for haplotype phase inference can have missing alleles or genotypes. After phasing data, all missing data are imputed.
- ❖ Alleles and marker identifiers can be any sequence of characters that does not contain white space and that does not equal the user-specified missing allele code. Marker alleles are not restricted to A/C/G/T or to 1/2/3/4.
- ❖ Data fields (e.g. marker alleles) on each line can be separated by a space, a tab, or any combination of spaces and tabs.
- ❖ Markers can have up to 128 different alleles. In particular, triallelic SNPs and microsatellite markers can be used.

Beagle files contain two sections: **header lines** and **marker lines**. **Header lines** are all lines that precede the first marker line (M ...). Header lines contain non-marker data. In the preceding two example Beagle files, the header lines are the identifier line (I ...) and the affection status line (A ...). **Marker lines** are the lines beginning with the first marker line (M ...) and ending with the last line of the file (inclusive). BEAGLE ignores any line in the markers line section whose first field is not 'M'.

Each line in a Beagle file must have the same number of fields, unless the first field is the hash character '#'. A line whose first field is the hash character, '#', is called a comment line. Comment lines (# ...) are ignored. All comment lines in the header line section are copied to the output Beagle files.

If an identifier line (I ...) is not included in your file, an identifier line with the format

```
I       id      col.3   col.4   col.5   col.6   col.7   col.8   ...
```

is automatically added and included in output Beagle files. If multiple identifier lines are included in the header lines, only the first one will be used by BEAGLE to identify columns. You can also include a pedigree identifier (P ...) line, a father identifier (Fid ...) line, a mother identifier (Mid ...) line, and a population stratum line (S ...) in the header line section.

The header line section also contains all phenotype variables, including binary traits (A ...), quantitative traits (T ...), and categorical covariates (C ...). Affection status data are used for association testing, but are not used for phasing or for building the haplotype frequency model. Quantitative trait and covariate data are not currently used by BEAGLE, but can be used by other programs, such as R [1].

BEAGLE can use transmitted and untransmitted haplotypes from parent-offspring data (affected individuals and their parents) to test for association with a binary affection status. With transmitted and untransmitted haplotypes, you will need to add a new affection status line with the affection status of transmitted haplotypes coded as 2 and the affection status of untransmitted haplotypes coded as 1, and you will need to use the diplotypes=false option (see Section 4.2.3 Arguments for association testing). When diplotypes=false, the analysis assumes haplotypes are independent, and haplotypes in adjacent columns (e.g. columns 3-4, 5-6, etc.) are not required to have the same affection status.

When inferring haplotype phase and missing data, you will typically create a separate Beagle file for each chromosome. The markers in the Beagle file must be in chromosomal order.

2.2 Markers file format

If the data for a chromosome is divided among two or more Beagle files, you must use a **markers file**. A markers file is required to reconstruct the marker order because BEAGLE does not require an input file to contain all the markers. For example, when imputing ungenotyped markers in a sample using a reference panel, the ungenotyped markers may be omitted from the Beagle file with the sample's data.

Each line of the marker file will contain four or more white-space delimited fields. The first field is the marker identifier. The second field is the marker position. The remaining fields are the marker alleles. The markers must be given in chromosomal order. The marker positions on a chromosome can be base pair positions or genetic positions. The marker position is not currently used by BEAGLE.

In the Beagle file given in Example 1 above, the corresponding markers file when marker positions are given in NCBI Build 35 coordinates is

```
rs2289311     79235661     A     G
rs1248628     79236165     C     T
rs10762764    79236371     G     T
```

If you include data for the triallelic SNP, rs2032582, the corresponding line of the marker file would be:

```
rs2032582     86205269     G     T     A
```

3 Inferring haplotype phase and missing data with BEAGLE

Beagle can perform haplotype phase inference and missing data imputation using data from unrelated individuals, parent-offspring trios, parent-offspring pairs, and phase-known haplotypes.

3.1 Quick start guide

To run BEAGLE, enter the following command at the computer prompt:

```
java -Xmx<Mb>m -jar beagle.jar <arguments>
```

where <Mb> is the number of Megabytes of memory available (e.g. -Xmx1000m) and <arguments> is a space separated list of arguments. Each argument has the format **parameter=value**. There is no white-space between the **parameter** and **=** or between **=** and the **value**. Large data sets with thousands of samples may require several gigabytes of memory (see Section 5 Using BEAGLE with large data sets).

New BEAGLE users should use the arguments for specifying files and the missing allele code, but should not need to specify any other arguments. Other BEAGLE arguments are optional and have sensible default values. The format for input Beagle files is described in Section 2 Input file format.

The commands for inferring haplotype phase and imputing missing data are very simple. Here are three example command lines:

1. java -Xmx1000m -jar beagle.jar unphased=fileA.bgl missing=?
2. java -Xmx1000m -jar beagle.jar unphased=fileA.bgl phased=fileB.bgl markers=markers.txt missing=?
3. java -Xmx1000m -jar beagle.jar unphased=fileA.bgl unphased=fileB.bgl trios=fileC.bgl pairs=fileD.bgl markers=markers.txt missing=?

Line 1 shows how to infer haplotype phase and impute sporadic missing data. In line 1, fileA.bgl contains unphased unrelated data with missing alleles coded as "?". If fileA.bgl contains unphased parent-offspring trios, replace unphased=fileA.bgl with trios=fileA.bgl. If fileA.bgl contains unphased parent-offspring pairs, replace unphased=fileA.bgl with pairs=fileA.bgl

Line 2 is a typical command for imputing ungenotyped markers in a file called fileA.bgl that have been genotyped in a reference panel called fileB.bgl. The markers file lists the combined set of markers in chromosomal order. Missing alleles in the input file are coded as "?". In line 2, fileA.bgl contains unphased unrelated data, and the reference panel (fileB.bgl) is phased, unrelated data. If the reference panel is unphased, unrelated data, replace phased=fileB.bgl with unphased=fileB.bgl. If the reference panel is parent-offspring trio data, replace phased=fileB.bgl with trios=fileB.bgl. Reference panels can have low levels (< 7%) of sporadic missing genotypes.

Line 3 illustrates how to infer haplotype phase and imputing sporadic missing data simultaneously in multiple cohorts. Using multiple cohorts permits large sample sizes, which increases the accuracy of the inferred haplotypes and missing data. In line three there are four cohorts. Two cohorts contain unphased, unrelated data (fileA.bgl, fileB.bgl), one cohort contains parent-offspring trio data (fileC.bgl), and one cohort contains parent-offspring pair data (fileD.bgl). The markers file (markers.txt) contains the markers included in the analysis, and missing alleles in the input files are coded as "?". If any markers in the markers file are not present in an input file, genotypes for the missing markers will be imputed.

A markers file is required when there is more than one input data file. The format of the markers file is described in Section 2.1 Beagle file format. When using more than one input file, it is recommended that at least one of the input files contain genotype data for all the markers in the markers file. For example, if you are imputing ungenotyped markers in an unphased sample using phased HapMap data as a reference panel, it is recommended that the markers file not include markers that are absent (ungenotyped) in the HapMap data, even if they are genotyped in unphased sample. If an input Beagle file is missing any markers present in the markers file, data for the missing markers will be imputed and included in the output files. If an input file contains a marker that is not present in the markers file, the marker will be ignored.

3.2 Command line arguments

This section describes the BEAGLE command line arguments for inferring haplotype phase and missing data.

During analysis BEAGLE will create several temporary files in your system's default temporary-file directory. If your system's default temporary-file directory has insufficient space, you can specify the temporary-file directory by replacing the initial "java" in the command line with the "java -Djava.io.tmpdir=<directory>" argument where "<directory>" is the name of an alternate directory for storing temporary files.

3.2.1 Arguments for specifying input data

- unphased=<unphased unrelated file> where <unphased unrelated file> is the name of a Beagle file containing **unphased unrelated** genotype data (see Section 2.1 Beagle file format). You may use multiple unphased arguments if data from different cohorts are in different files.
- phased=<phased unrelated file> where <phased unrelated file> is the name of a Beagle file containing **phased unrelated** data (see Section 2.1 Beagle file format). You may use multiple phased arguments if data from different cohorts are in different files.
- trios=<unphased trio file> where <unphased trio file> is the name of a Beagle file containing unphased parent-offspring trio data (see Section 2.1 Beagle file format). You may use multiple trios arguments if data from different cohorts are in different files.
- pairs=<unphased pairs file> where <unphased pairs file> is the name of a Beagle file containing unphased parent-offspring pair data (see Section 2.1 Beagle file format). You may use multiple pairs arguments if data from different cohorts are in different files.
- markers=<markers file> where <markers file> is the name of the markers file containing marker identifiers, positions, and alleles described in Section 2.2 Markers file format. The markers argument is optional if you specify only one Beagle file, and is required if you specify more than one Beagle file.
- missing=<missing code> where <missing code> is the character or sequence of characters used to represent a missing allele (e.g. missing=-1 or missing=?). The missing argument is required.

3.2.2 Other phasing arguments

- niterations=<number of iterations> where <number of iterations> is a positive even integer giving the number of iterations of the phasing algorithm. If an odd integer is specified, the next even integer is used. The niterations argument is optional. The default value is niterations=10. The default value typically gives good accuracy.
- nsamples=<number of samples> where <number of samples> is positive integer giving the number of haplotype pairs to sample for each individual during each iteration of the phasing algorithm. The nsamples argument is optional. The default value is nsamples=4.

If you are phasing an extremely large sample (say > 4000 individuals), you may want to use a smaller nsamples parameter (e.g. 1 or 2) to reduce computation time. If you are phasing a small sample (say < 200 individuals), you may want to use a larger nsamples parameter (e.g. 10 or 20) to increase accuracy.

- log=<log file prefix> where <log file prefix> is the prefix for the output .log file. The log argument is required. You can have the .log file written to a different directory by specifying a pathname.
- out=<output file prefix> where <output file prefix> is the prefix for the output filenames (except for the .log output file, which is specified with the log argument). The different output files are described in Section 3.3 Output files. The out argument is optional.
  - If no output file prefix is specified, each output file will be written to the working directory. If an output file prefix is specified and is a directory, each output file will be written to the specified directory. Output files corresponding to an input file called z.bgl will be named z.bgl.[ext] where [ext] describes the data in the output file.
  - If the output file prefix is specified and is a pathname, output files corresponding to an input file called z.bgl will be [output file prefix].z.bgl.[ext] where [ext] describes the data in the output file.
- gprobs=<true/false> where <true/false> is false if genotype probability files should not be produced (see Section 3.3.3 genotype probability file [.gprobs]). The gprobs argument is optional. The default value is gprobs=true. If the input Beagle files have sporadic missing data and you are not imputing data for ungenotyped markers, we recommend setting gprobs=false.
- seed=<random seed> where <random seed> is an integer seed for the random number generator. The seed argument is optional. The default value is seed=-99999.
- lowmem=<true/false> where <true/false> is true if a memory-efficient, but slower, implementation of the sampling algorithm should be used. If lowmem=true the running time will increase by a factor ≤ 2, and the memory usage will be essentially independent of the number of markers. The lowmem argument is optional. The default value is lowmem=false.
- excludecolumns=<excluded columns file> where <excluded columns file> is the name of file containing column identifiers (one identifier per line) that will be excluded from the analysis and output files. Beagle will check the column identifiers in the first identifier line (I ...) of each Beagle input file and exclude any columns whose identifier is in the excluded samples file.
- excludemarkers=<excluded markers file> where <excluded markers file> is the name of file containing marker identifiers (one identifier per line) that will be excluded from the analysis and output files.
- verbose=<true/false> where <true/false> is true if running time and graphical model statistics are printed to the .log file for each iteration of the algorithm. The verbose argument is optional. The default value is verbose=false.
- nimputations=<number of imputations> where <number of imputations> is a nonnegative integer giving the number of sampled, phased Beagle files to create for each

input Beagle file (see Section 3.3.5 sampled haplotype file [.sample]). These files will have phased unrelated data, phased parent-offspring trio data, or phased parent-offspring pair data, according to the data in the input BEAGLE file. The nimputations argument is optional. The default value is nimputations=0. If multiple runs of BEAGLE are used to sample haplotypes, a different random seed should be specified for each run (see the seed argument discussed earlier in this section).

3.2.3 Advanced options not intended for general use

- maxwindow=<maximum window size> where <maximum window size> is a positive integer giving the maximum number of consecutive markers that will be considered when building the haplotype frequency model. The maxwindow argument is optional. The default value is maxwindow=500. The default value should be sufficiently large for most data sets. For example, if your marker density is one marker per kilobase, BEAGLE will consider markers in a 500 kilobase window when using the default maxwindow parameter.

3.3 Output files

After inferring haplotype phase and missing data, a .log file is created that summarizes the analysis, and a .phased file is created for each input data. The .phased file gives the most likely haplotype phasing. Depending upon the command line arguments, additional output files (.gprobs, .r2, and .sample) may also be created.

3.3.1 log file [.log]

The .log file gives a summary of the analysis that includes the BEAGLE version, a list of the command line arguments for the analysis, a description of the command line arguments, and the running time for the analysis. If the verbose=true is included as a command line argument, the .log file will also include the running time for each iteration of the phasing algorithm and descriptive statistics for the graphical haplotype frequency model at each iteration of the phasing algorithm. The filename for the .log file is set with the log command line argument (see Section 3.2.2 Other phasing arguments).

3.3.2 phased file [.phased]

Each input Beagle file will have a corresponding .phased output Beagle file that gives imputed missing data and inferred haplotypes. The .phased file gives the most likely haplotype pair for each individual conditional upon the genotypes for the individual and the haplotype frequency model. If the input file contains data for unrelated individuals, the corresponding .phased output file will contain phased unrelated data. If the input file contains unphased parent-offspring trio data, the corresponding .phased output file will contain phased trio data. If the input file contains unphased parent-offspring pair data, the corresponding .phased output file will contain phased pair data (see Section 2.1 Beagle file format). A directory or a filename prefix for the .phased output files can be set with the out command line argument (see Section 3.2.2 Other phasing arguments).

3.3.3 genotype probability file [.gprobs]

Each input Beagle file that contains only diallelic markers (e.g. SNPs) and unphased unrelated data will have a corresponding genotype probability file (.gprobs) unless the gprobs=false argument is specified (see section 3.2.2 Other phasing arguments). Lines of the

.gprobs file correspond to markers in the same order as the output .phased file. The first column gives the marker identifier. The second gives the A allele and the third column gives the B allele for each diallelic marker. The remaining columns give genotype probabilities for the A/A, A/B, and B/B genotypes in that order for each individual (3 columns per individual). For example, if a line in the .gprobs file begins

```
rs3768203   C   T   0.000   0.989   0.011   0.982   0.018   0.000   . . .
```

then the individual in columns 3-4 of the input Beagle file has heterozygous genotype (C/T) with probability 0.989 and homozygous T/T genotype with probability 0.011, and the individual in columns 5-6 of the input Beagle file has homozygous genotype (C/C) with probability 0.982 and heterozygous C/T genotype with probability 0.018. If an individual is genotyped for a marker, the posterior probability of the genotype will be 1. A directory or a filename prefix for the .gprobs output files can be set with the out command line argument (see Section 3.2.2 Other phasing arguments).

3.3.4 allelic R^2 file [.r2]

The allelic R^2 file (.r2) file is produced whenever a .gprobs file is produced (see Section 3.3.3 genotype probability file [.gprobs]). The allelic R^2 file contains two columns, the first column gives the marker identifier, and the second column gives the estimated squared correlation ($0 \leq R^2 \leq 1$) between the allele dosage with highest posterior probability in the .gprobs file and the true allele dosage for the marker. Larger values of allelic R^2 denote more accurate genotype imputation. The estimated allelic R^2 for a marker is “NaN” (Not-a-Number) when the estimated allelic R^2 cannot be computed. The allelic R^2 cannot be computed for a marker when the most likely genotype is the same for each individual (i.e. only one genotype is observed), or when there is no data for the marker in any Beagle input file. The estimated squared correlation can be used when deciding whether to retain or discard a marker in downstream analyses. The allelic R^2 file can also be used to detect inter-cohort differences in imputation accuracy. A directory or a filename prefix for the .r2 output files can be set with the out command line argument (see Section 3.2.2 Other phasing arguments).

3.3.5 sampled haplotype file [.sample]

Sampled haplotype files (.*k*.sample, where *k* is a positive integer) are generated only when the nimputations command line argument is used (see Section 3.2.2 Other phasing arguments). If nimputations=*K* is specified (*K = 1, 2, ...*), each input Beagle file will have a *K* phased output files with names ending in *k*.sample for *k = 1, 2, ..., K*. Each .sample output file is a phased Beagle file (see Section 2.1 Beagle file format). The .sample output files will have phased unrelated data, phased parent-offspring trio data, or phased parent-offspring pair data, according to the data in the input Beagle file. In contrast to the .phased output file which gives the most likely haplotypes for each individual, each .sample output file gives randomly sampled haplotypes for each individual. The randomly sampled haplotypes are sampled conditional on the genotypes for the individual and the haplotype frequency model. A directory or a filename prefix for the .sample output files can be set with the out command line argument (see Section 3.2.2 Other phasing arguments).

4 Association testing with BEAGLE

BEAGLE can perform single marker and haplotypic tests for association for case-control studies and for parent-offspring trio studies with affected offspring. BEAGLE can also calculate multiple-testing adjusted P-values using permutation of the trait status. BEAGLE performs haplotypic association testing by building a graphical model of haplotype frequencies, clustering haplotypes that are similar near each marker, and testing the haplotype clusters for association with the trait status.

BEAGLE can be used in conjunction with the PRESTO software package (http://www.stat.auckland.ac.nz/~browning/presto/presto.html) and the pseudomarker utility program (see Section 7.6 pseudomarker). PRESTO can compute empirical distributions of order statistics, analyze stratified data, and determine significance levels for one and two-stage genetic association studies.

4.1 Quick start guide

To run BEAGLE, enter the following command at the computer prompt:

```
java -Xmx<Mb>m -jar beagle.jar <arguments>
```

where <Mb> is the number of Megabytes of memory available (e.g. -Xmx1000m) and <arguments> is a space separated list of arguments. Each argument has the format **parameter=value**. There is no white-space between the **parameter** and **=** or between **=** and the **value**. Large data sets with thousands of samples may require several hundred megabytes of memory (see Section 5 Using BEAGLE with large data sets).

The commands for building a graphical model of haplotype frequencies and for performing single marker and haplotypic association tests are very simple. New BEAGLE users should initially use only the arguments for specifying the data file, the trait, and the association tests illustrated in the three example command lines given below. Other BEAGLE arguments are optional and have sensible default values.

Here are three examples command lines for performing association tests or building a graphical model of haplotype frequencies.

1. `java -Xmx800m -jar beagle.jar data=data.bgl trait=T2D`
2. `java -Xmx800m -jar beagle.jar data=data.bgl trait=CD test=adr`
3. `java -Xmx800m -jar beagle.jar data=data.bgl`

The input file for these examples is a phased Beagle file called data.bgl. The input phased Beagle file cannot have missing data. Line 1 performs allelic tests for association with an affection status variable named T2D. Line 2 performs allelic (a), dominant (d), and recessive (r) tests for association with an affection status variable named CD. Line 3 builds a graphical model of haplotype frequencies without performing association testing.

BEAGLE can also perform association testing with transmitted and untransmitted haplotypes from parent-offspring data with affected offspring. See the discussion at the end of Section 2.1 Beagle file format.

4.2 Command line arguments

This section describes the different BEAGLE command line arguments for performing association testing.

4.2.1 Argument for specifying input data

- data=<phased Beagle file> where <phased Beagle file> is the name of a Beagle file containing phased data with no missing alleles. The phased data can be any combination of phased unrelated data, phased parent-offspring trio data, or phased parent-offspring pair data (see Section 2.1 Beagle file format). The data argument is required.
- trait=<affection status identifier> where <trait> is the name of the affection status variable to use for association testing. The trait argument is optional. If the trait argument is omitted, then a graphical model of haplotype frequencies will be created and written to a .dag file (see Section 4.3.4 The model file [.dag]), but association testing will not be performed.
- out=<output file prefix> where <output file prefix> is the prefix for the output files. The different output files are described in Section 4.3 Output files. The out argument is optional.
  - If no output file prefix is specified, each output file will be written to the working directory. If an output file prefix is specified and is a directory, each output file will be written to the specified directory. Output files corresponding to an input file called z.bgl will be named z.bgl.[ext] where [ext] describes the data in the output file.
  - If the output file prefix is specified and is a pathname, output files corresponding to an input file called z.bgl will be [output file prefix].z.bgl.[ext] where [ext] describes the data in the output file.

4.2.2 Arguments for building the model

The scale and shift parameters control the number of haplotype clusters in the graphical model for the phased data. Having too many or too few haplotype clusters can reduce the power of association testing. The default values for the scale and shift parameters have performed well in simulation studies and real data analyses [2, 3].

The scale and shift parameters determine whether pairs of nodes will be merged during construction of the graphical model. If node A represents n_A haplotypes and node B represents n_B haplotypes, the two nodes will not merge if their similarity score [2, 4] is greater than

$$m\sqrt{\frac{1}{n_A}+\frac{1}{n_B}}+b$$

where is m the scale parameter and b is the shift parameter.

- scale=<threshold scale> where <threshold scale> is a positive floating point number giving the scale parameter used when deciding whether to merge a pair of nodes [2]. The scale argument is optional. The default value is scale=4.0. Decreasing the scale parameter will increase the number of haplotype clusters in the graphical model.
- shift=<threshold shift> where <threshold shift> is nonnegative floating point number less than or equal to 1.0 giving the shift parameter used when deciding whether to merge nodes a pair of nodes [2]. The shift argument is optional. The default value is shift=0.2. Decreasing the shift parameter will increase the number of haplotype clusters in the graphical model.

4.2.3 Arguments for association testing

- test=<association tests> where <association tests> is one or more characters from the set "ardo" where
  - a = allelic test.
  - r = recessive test (groups major allele homozygotes and heterozygotes).
  - d = dominant test (groups minor allele homozygotes and heterozygotes).
  - o = overdominant test (groups minor and major allele homozygotes).
- For example test=a or test=ardo. Fisher's exact test for 2 x 2 table is used for each test. If a marker has more than two alleles, each allele defines a diallelic marker by grouping the other alleles. Thus a triallelic marker will define 3 diallelic markers and these diallelic markers will be tested. The test argument is optional. The default value is test=a. Genotypic tests (r, d, and o) are not permitted when diplotypes=false. See the diplotypes argument in this section for more details.
- seed=<random seed> where <random seed> is an integer seed for the random number generator. The seed argument is optional. The default value is seed=-99999.
- nperms=<number of permutations> where <number of permutations> is a nonnegative integer giving the number of permutations of the affection status to use for permutation testing. Permutation testing determines multiple-testing adjusted p-values for all the haplotypes and single markers that are tested (see Section 4.3.2 The P-value file [.pval]). You can skip permutation testing by setting nperms=0. The nperms argument is optional. The default value is nperms=1000. The computation time for permutation testing is linear in the number of permutations.
- edgecount=<minimum edge count> where <minimum edge count> is a positive integer giving the minimum number of haplotypes in a haplotype cluster that is required to test the cluster for association with a trait. Each haplotype cluster is defined by an edge of the graphical model. See [4] for more details. The edgecount argument is optional. The default value is edgecount=20. The default value should work well, but expert users may want to adjust the edgecount parameter based on the sample size and a priori knowledge of the disease allele frequency.
- othercount=<minimum other count> where <minimum other count> is a nonnegative integer giving the minimum number of haplotypes in the set of edges that merge with an edge E that is required to test the haplotype cluster defined by *E*. Edges E_1 and E_2 are said to merge if E_1 and E_2 point to the same child node [4]. The othercount argument is optional. The default value is othercount=1. Increasing the othercount parameter will decrease the number of haplotype clusters (i.e. edges of the graphical model) tested for association with the affection status. A value of 0 is permitted but generally is not recommended because it will result in testing non-merging edges.
- diplotypes=<true/false> where <true/false> is true if the affection status is permuted for haplotype pairs so that both haplotypes for each individual have the same permuted trait status and <true/false> is false if the affection status is permuted for individual haplotypes (rather than for haplotype pairs). Only the allelic test can be performed when diplotypes=false (see the test argument in this section). The diplotypes argument is

optional. The default value is diplotypes=true. The diplotypes argument should not be used unless the phased data consist of transmitted and untransmitted haplotypes, and the affection status identifies the transmitted and untransmitted haplotypes as described at the end of Section 2.1 Beagle file format.

4.2.4 Advanced options not intended for general use

- maxwindow=<maximum window size> where <maximum window size> is a positive integer giving the maximum number of consecutive markers that will be considered when building the graphical model of haplotype frequencies. The maxwindow argument is optional. The default value is maxwindow=500. The default value should be sufficiently large for most data sets. For example, if your marker density is one marker per kilobase, BEAGLE will consider markers in a 500 kilobase window when using the default maxwindow parameter.

4.3 Output files

Depending on the parameters for the analysis, BEAGLE can produce up to four output files. Most users will be interested in only two output files: the log file (.log), and the P-value file (.pval). The remaining two files, the null P-value file (.null) and the model file (.dag), will be useful to some, but not most, users.

4.3.1 The log file [.log]

A log file is generated each time BEAGLE is run. The log file gives a summary of the analysis that includes the BEAGLE version, list of the command line arguments for the analysis, a description of the command line arguments, and the running time for the analysis.

If the trait parameter is specified, the log file gives a list of all markers or haplotype clusters from the P-value file (.pval) with a permutation P-value < 0.2.

4.3.2 The P-value file [.pval]

The P-value file records the P-values from testing markers and haplotype clusters for association with the affection status. Haplotype clusters are defined by edges of a graphical model. Each haplotype defines a path between the initial and terminal node of the graph [4]. For each edge E in the graphical model we define a haplotype cluster C_E to be the set of all haplotypes whose path from initial to terminal node traverses edge E. Haplotype clusters define diallelic markers, that we call **pseudomarkers**. Given a haplotype cluster C, the diallelic pseudomarker for a haplotype is 2 if the haplotype is in C and 1 if the haplotype is not in C. Representing haplotype clusters as pseudomarkers enables us to test the haplotype cluster for association with the trait status in the same way we test other diallelic markers: using Fisher's exact test for 2 x 2 tables. The pseudomarker program (see Section 7.6 pseudomarker) can create a phased Beagle file of pseudomarkers representing the haplotype clusters in the graphical model. The phased Beagle file of pseudomarkers can be imported into a program like R [1] for statistical analysis. The first line of the P-value file is a header line describing the columns of the file. Each line (except the header line) gives the P-values from testing one marker or pseudomarker for association with the trait status. First, the P-values for the genotyped markers are given, then the P-values for the pseudomarkers are given. The edgecount and othercount parameters described in Section 4.2.3 Arguments for association testing determine which pseudomarkers are selected for testing.

Example 3 - First and last lines of a P-value [.pval] file

```
Marker    Allele   allelic_p   min _p    min_p_perm
m14954    0        0.4065      0.4065    1.000
m14992    1        0.1064      0.1064    1.000
m15081    1        0.7968      0.7968    1.000
```

...skipped lines of P-value file...

```
m28524    0.1      0.9121      0.9121    1.000
m28524    1.0      0.3161      0.3161    1.000
m28662    1.0      0.3266      0.3266    1.000
m28662    0.0      0.7873      0.7873    1.000
m28662    0.1      0.3983      0.3983    1.000
```

The first field on the line is the marker identifier. The second field is the marker allele that is tested. If the marker has more than two alleles, the allele is used to define a diallelic marker by grouping all other alleles as the second allele (see the test parameter in Section 4.2.3 Arguments for association testing). For pseudomarkers defined by graph edges, the allele field has the format "parent.allele" where parent is the parent node number and allele is the marker allele for the edge (see the last 5 lines of Example 3). The marker identifier, the parent node number, and the marker allele uniquely determines the edge of the graphical model (see Section 4.3.4 The model file [.dag], and [2, 4] for more discussion of the graphical model). After the marker field and marker allele field, the next columns give the P-values for allelic, recessive, dominant, and overdominant tests. Columns corresponding to tests which were not performed are omitted (see the test command line argument in Section 4.2.3 Arguments for association testing). The second-to-last column gives the minimum P-value observed for the marker. If only one test is performed, as is the case in the preceding P-value file excerpt, the minimum P-value equals the P-value for that test. The final column gives the permutation P-value.

The permutation P-value is a measure of significance that accounts for multiple testing. For example, if your significance level is $\alpha = 0.05$, and a marker allele or pseudomarker allele has a permutation P-value $p < 0.05$, the association is significant after accounting for multiple testing. More generally, for a given significance level α $(0 \leq \alpha \leq 1)$, the probability of observing one or more marker alleles or pseudomarker alleles with a permutation P-value less than α is less than or equal to α under the null hypothesis that the trait and marker data are independent [5].

Given a set of tests (specified with the test parameter), the permutation test randomly permutes the trait status and tests the marker alleles and pseudomarker alleles for association with the permuted trait status. If diplotypes=true, which is the default setting, the trait status is permuted for the individuals so that both haplotypes for each individual have the same permuted trait status. When the data consists of transmitted and untransmitted haplotypes, the diplotypes=false argument must be used so that the trait status is permuted for the haplotypes rather than for the individuals (see Section 5.4).

For each permuted trait status, the set of tests (determined by the test parameter) is applied to all markers and the minimum P-value (minimized over all markers and pseudomarkers and all tests) is saved and written to the null P-value file (see Section 4.3.3 The null P-value file

[.null]). If a marker or pseudomarker has a minimum P-value of p_0 (minimized over all tests for that marker or pseudomarker) when tested for association with the unpermuted trait status, and if for k out of N permutations of the trait status there exists at least one marker or pseudomarker with a minimum P-value less than or equal to p_0 when tested for association with the permuted trait status, the permutation P-value for the marker is $(k + 1)/(N + 1)$ [5]. Under the null hypothesis, expect most alleles to have a permutation P-value of 1.000 since the P-value of a single allele is being compared to the minimum P-value from all alleles of all markers. The P-value file is designed to be imported into a spreadsheet or a statistical software package. However, if you want to quickly identify the most significant markers, look in the output .log file. The .log file contains a list of all markers and pseudomarkers with a permutation P-value < 0.2.

4.3.3 The null P-value file [.null]

The null P-value file lists the minimum P-values (minimized over all markers and pseudomarkers, alleles and all tests) observed when testing the marker and pseudomarker alleles for association with a randomly permuted trait status. The P-values in the null P-value file (.null) are used to determine the multiple-testing adjusted P-value given in the .pval file (see Section 4.3.2 The P-value file [.pval]). The j-th line gives the minimum P-value from the j-th permuted trait status for $j = 1,2, \dots, N$ where N equals the value of the nperms parameter (see Section 4.2.3 Arguments for association testing). The null P-value file gives an empirical distribution of the minimum P-value under the null hypothesis. If the argument nperms=0 is used, the null P-value file will be empty. The sequence of permutations of the trait status is determined by the seed argument (see Section 4.2.3 Arguments for association testing).

4.3.4 The model file [.dag]

The model file gives the graphical model for the haplotype clusters. The graphical model is a directed acyclic graph (DAG): levels of the DAG correspond to markers and edges of the DAG correspond to haplotype clusters [2, 4].

The first line of the file is a header line describing the columns in the file. Each line describes an edge of the graph. Edges corresponding to the same marker are grouped together and preceded and succeeded by a blank line. The first columns and lines of the model file look like this:

```
Level  Marker   Parent  Child  Allele  Count  Haplotype identifiers
0      m14954   0       0      0       2964   0    1   5    6    7    8    ...
0      m14954   0       1      1       1036   2    3   4    10   12   15   ...

1      m14992   0       0      1       2207   0    5   6    7    8    14   ...
1      m14992   0       1      0       757    1    9   11   13   27   28   ...
1      m14992   1       2      1       1036   2    3   4    10   12   15   ...
```

The first six fields in a line of the model file are:

1. The level of the parent node of the edge. If there are M markers, the level numbers are 0, 1, 2, ... , M − 1.
2. The marker identifier given in the input BEAGLE file corresponding to the level of the parent node.

3. The node number of the parent node of the edge. Node numbering begins at 0 for each level.
4. The node number of the child node of the edge. Node numbering begins at 0 for each level, and the level of the child node is one more than the level of the parent node.
5. The marker allele that is carried by all haplotypes in the cluster defined by the edge.
6. The number of haplotypes in the cluster defined by the edge.

If sixth field is *K*, then there are *6+K* fields on the line. The final *K* fields are nonnegative integers identifying the haplotypes in the phased Beagle file specified with the data argument (see Section 4.2.1 Argument for specifying input data) that are in each haplotype cluster. The haplotypes are numbered 0, 1, 2, ... in the order they appear as columns in the phased Beagle file (see Section 2.1 Beagle file format), so 0 refers to column 3 in the phased Beagle file, 1 refers to column 4 in the phased Beagle file, and so on.

5 Using BEAGLE with large data sets

5.1 Memory management

You can increase the amount of memory available to the java interpreter using the -Xmx command line argument. If [Mb] is a positive integer, then -Xmx[Mb]m sets the maximum amount of memory that will be used by the java interpreter to [Mb] megabytes. It is helpful to set the -Xmx parameter somewhat higher than the minimum memory required to analyze your data because having the additional memory available can result in decreased computation time.

For association testing, memory usage depends on the number of individuals in the analysis, but is independent of the number of markers when analyzing more than 1000 markers. In general, association testing will not generally require more than 500-2000 Mb of memory.

For haplotype phase inference and imputation of missing data with default BEAGLE options, memory usage increases with the number of markers. It you need to reduce the amount of memory BEAGLE is using, try one or more of the following techniques:

1. Use the lowmem=true command line argument (see Section 3.2.2 Other phasing arguments). The lowmem option makes memory requirements essentially independent of the number of markers.
2. Divide longer chromosomes into two parts or more parts, and infer haplotype phase and perform genotype imputation on each part separately. Dividing a chromosome in the middle of a long genomic segment without markers is recommended. Dividing a chromosome at points where there is dense marker coverage is less desirable because imputation accuracy will decrease somewhat for markers near the dividing point.
3. If you are imputing ungenotyped markers, you can divide the sample into subsamples, and perform imputation on each subsample separately (but do not divide the reference panel). **If you divide the sample, the proportion of cases and controls should be approximately the same in each subsample**. It is okay to have cases and controls for a subsample in separate input files as long as cases and controls are analyzed together in the same Beagle analysis. A sample can be divided into subsamples by using the cut

utility program (see Section 7.3 cut), or by specifying individuals to exclude from the subsample using the excludecolumns argument (see Section 3.2.2 Other phasing arguments). Output files can be pasted together using the paste utility program (see Section 7.4 paste). A unix shell program called divide.sample is included in the BEAGLE software distribution to facilitate imputation analysis using subsamples of the data.

Long stretches of consecutive markers with completely missing data in a parent-offspring trio can require enormous amounts of memory. If a trio is missing all its genotypes in a genomic region, the region may need to be analyzed separately with the problematic trio removed.

5.2 Genomewide association studies

If you want to perform association testing on data from a genomewide association study, first divide the data by chromosome, and then phase the data for each chromosome separately. If your input data is divided among multiple input files, use the paste utility to combine the resulting phased Beagle files for a chromosome into a single phased Beagle file (see Section 7.4 paste). When there are multiple input Beagle files, make sure the multiple phased output files are combined in the same order for each chromosome. After phasing each chromosome, concatenate the phased Beagle files for each chromosome (e.g. with the unix cat command), and test the resulting phased Beagle file (see Section 4 Association testing with BEAGLE). It is important to test all markers in a single run so that BEAGLE can perform permutation testing to determine statistical significance.

6 Example BEAGLE analyses

6.1 Inferring haplotype phase and missing data

The BEAGLE software distribution includes sample files to illustrate genotype imputation. The following input files are contained in the example/imputation folder:

1. hapmap.markers - A markers file listing 100 markers on chromosome 1. The four columns give the marker identifier, the base position, and the two alleles.
2. hapmap.phased.bgl - A phased Beagle file with 58 phased individuals genotyped on the 100 markers in the hapmap.markers file. The data is from the HapMap CEU panel.
3. hapmap.unphased.bgl - An unphased Beagle file with 2 individuals genotyped on 50 markers in the hapmap.markers file. The data is from the HapMap CEU panel.

The missing 50 markers in hapmap.unphased.bgl can be imputed using the data in hapmap.phased.bgl as a reference panel. The BEAGLE command line is

```
java -Xmx500m -jar beagle.jar markers=hapmap.markers phased=hapmap.phased.bgl
unphased=hapmap.unphased.bgl missing=? log=imputation
```

Five output files are created by the preceding command line:

1. **hapmap.phased.bgl.phased** - phased Beagle output file corresponding to the hapmap.phased.bgl input file (see Section 3.3.2 phased file [.phased]). Any missing data in the input Beagle file is imputed using the most likely haplotypes.

2. **hapmap.unphased.bgl.phased** - phased Beagle file corresponding to the hapmap.unphased.bgl input file (see Section 3.3.2 phased file [.phased]). The .phased output file gives the most likely haplotype pair for each individual.

3. **hapmap.unphased.bgl.gprobs** - genotype probability file corresponding to the individuals in the hapmap.unphased.bgl file (see Section 3.3.3 genotype probability file [.gprobs]). Posterior genotype probabilities are given for all markers. The posterior genotype probability will be 1.0 for all assayed (i.e. non-imputed) genotypes.

4. **hapmap.unphased.bgl.r2** - allelic R^2 file corresponding to the hapmap.unphased.bgl file (see section 3.3.4 allelic R^2 file [.r2]). For each genotyped or imputed marker, the allelic R^2 file gives the estimated correlation between the allele dosage with highest posterior probability and the true allele dosage. In this example analysis, the allelic R^2 file can be ignored because the sample size (2 individuals) is too small to obtain accurate estimates.

5. **imputation.log** - log file summarizing the imputation analysis (see Section 3.3.1 log file [.log]).

6.2 Association testing

The BEAGLE software distribution includes sample files to illustrate single marker and haplotypic association testing using Beagle. The example/testing folder contains an input Beagle file with phased unrelated data called data.bgl. The example data are 200 markers for 4000 haplotypes (1000 case and 1000 control individuals), generated using Cosi version 1.0 [6]. The affections status variable in the input file is named T2D. The command for association testing is

```
java  -Xmx500m -jar  beagle.jar  data=data.bgl  trait=T2D
```

Four output files are created by the preceding command line:

1. **data.bgl.log** - a log file summarizing the association analysis (see Section 4.3.1 The log file [.log])

2. **data.bgl.pval** - a P-value file with association test statistics and multiple-testing adjusted P-values for markers and haplotype clusters (see Section 4.3.2 The P-value file [.pval])

3. **data.bgl.null** - a null P-value file with minimum test statistics observed under the null hypothesis (see Section 4.3.3 The null P-value file [.null])

4. **data.bgl.dag** - a model file describing the graphical model of haplotype clusters (see Section 4.3.4 The model file [.dag])

The output log file (data.bgl.log) contains the following excerpt listing the 3 marker alleles and 3 pseudomarkers alleles (i.e. haplotype clusters) that have permutation P-values less than 0.2:

Alleles with permutation P-values less than 0.2:

```
Marker    Allele    allelic_p    min_p        min_p_perm
m23508    0         0.0004167    0.0004167    0.1179
m23612    0         0.0002015    0.0002015    0.06593
m24828    0         0.0006412    0.0006412    0.1908
m23171    2.1       7.046e-05    7.046e-05    0.06593
m23892    9.1       4.559e-05    4.559e-05    0.01898
m24828    1.1       0.0006412    0.0006412    0.1908

6 alleles with permutation P-value < 0.2
```

One test was significant at the $\alpha = 0.05$ level after accounting for multiple testing (permutation P-value 0.01898). Allele 9.1 of marker m23892 is the haplotype cluster with parent node 9 and allele 1 at marker m23892. The haplotypes in this cluster are recorded in the output model file (data.bgl.dag).

The allele sequence that defines the haplotypes in the associated haplotype cluster can be identified using the cluster2haps utility program (see Section 7.7 cluster2haps). The cluster2haps command line is:

```
java -jar cluster2haps.jar dag=data.bgl.dag phased=data.bgl trait=T2D marker=m23892
parent=9 allele=1 out=m23892_9_1.haps
```

The output data from in the file m23892_9_1.haps shows that the cluster can be defined by many different allele sequences. One of the allele sequences defining the cluster is the allele sequence 1 - 0 - 1 for markers m23490 - m23628 - m23892.

7.1 Utility programs

Input files for the utility program described in this section can be compressed with the gzip algorithm. The utility programs assume that any file that has a name ending in “.gz” is compressed with gzip.

7.1 linkage2beagle

The linkage2beagle program creates a Beagle file from a data file and a pedigree file. The format for the data and pedigree files is similar to linkage and QTDT format. QTDT format is described at http://www.sph.umich.edu/csg/abecasis/QTDT/docs/data.html.

A pedigree file has rows corresponding to individuals and columns corresponding to variables. The first five columns are fixed and give the pedigree identifier, individual identifier, father’s identifier, mother’s identifier, and gender. If any of these variables are unknown or undefined, then 0 can be used. The pedigree identifier, father’s identifier, and mother’s identifier need not be defined for case-control data. After the first five fixed columns, the remaining columns are variable and specified by the data file.

The lines of the data file describe the variables in the pedigree file in the order they appear as columns in the pedigree file, beginning with the sixth column. Each line of the data file has two white-space separated fields. The first field identifies the type of data in the column, and the second field is the identifier for the variable. If the first field is “M” (for marker), the variable is a genotype and corresponds to two columns of the pedigree file; otherwise, the variable corresponds to a single column of the pedigree file. (Note: the S[n] code used in QTDT is not supported by **linkage2beagle**).

The **linkage2beagle** program creates a Beagle file corresponding to the pedigree and data files.

To run the **linkage2beagle** program enter

```
java -Xmx500m -jar linkage2beagle.jar <arguments>
```

where <arguments> is a space separated list of arguments. Each argument has the format **parameter=value**. There is no white space between the **parameter** and **=** or between **=** and the **value**. The arguments are

- pedigree=<pedigree file> where <pedigree file> is the filename of a pedigree file. The pedigree argument is required.
- data=<data file> where <data file> is the filename of a data file that describes the columns of the pedigree file. The markers in the data file must be in chromosomal order. The data argument is required.
- beagle=<Beagle file> where <Beagle file> is the filename of the Beagle file that will be created from the pedigree and data files. The beagle argument is required.
- standard=<true/false> where <true/false> is true if the first five columns of the pedigree file are the standard first five columns (pedigree ID, sample ID, father's ID, mother's ID, and gender, in that order), and false otherwise. If standard=true, then the data file must describe columns 6 and higher. If standard=false, the data file must describe all columns, starting with column 1. The standard argument is optional. The default value is standard=true. When standard=true, the data in the first five columns will be given in the first five rows of the output Beagle file, and these first five rows will begin (P pedigree, I id, fID father, mID mother, and, C gender).

For example, when using the default standard=true argument, if the data file is

```
A   diabetes
M   rs1248696
M   rs2289311
T   BMI
C   age.of.onset
```

and the associated pedigree file is

```
0   1001   0   0   1   1   A  G    T  T      23.0  X
0   1002   0   0   1   2   G  G    T  T      24.0  34.5
0   1003   0   0   2   2   G  G    T  C      25.0  67.8
```

then the following BEAGLE file will be created:

```
P    pedigree     0    0    0    0    0     0
I    id           1001 1001 1002 1002 1003  1003
fID  father       0    0    0    0    0     0
mID  mother       0    0    0    0    0     0
C    gender       1    1    1    1    2     2
A    diabetes     1    1    2    2    2     2
T    BMI          23.0 23.0 24.0 24.0 25.0  25.0
C    age.of.onset X    X    34.5 34.5 67.8  67.8
M    rs1248696    A    G    G    G    G     G
M    rs2289311    T    T    T    T    T     C
```

Notice that all non-marker variables in the output Beagle file have been placed in the header line section (see Section 2.1 Beagle file format).

7.2 phased2beagle

phased2beagle creates a phased Beagle from a file that contains one phased haplotype per line. To run the phased2beagle program enter

```
java -Xmx600m -jar phased2beagle.jar <arguments>
```

where <arguments> is a space separated list of arguments. Each argument has the format **parameter=value**. There is no white space between the **parameter** and = or between = and the **value**. The arguments are

- phased=<phased data file> where <phased data file> is the name of a file containing phased haplotypes (one haplotype per line). The alleles on a line must be separated by white space. If a line in the phased data file has only one field, each character of the field is interpreted as an allele. The phased argument is required.
- beagle=<phased Beagle file> where <phased Beagle file> is the filename of the Beagle file that will be created from the phased haplotype data. The beagle argument is required.
- markers=<markers file> where <markers file> is the filename of a file containing the marker identifiers (one marker per line) in the order they appear on a line in the phased data file. The markers argument is optional. If the markers argument is not used, markers will be labelled marker_1, marker_2, marker_3, and so on.

7.3 cut

The **cut** program is used to extract columns from a Beagle file. The commands for running the **cut** program are:

java -jar cut.jar [in file] a_1:b_1 a_2:b_2 a_3:b_3 … [out file]

where [in file] is an input Beagle file, $1 \le a_i \le b_i$ are the first and last column indices (inclusive) of a set of column indices that will be extracted, and [out file] is the output file with the extracted columns. Column indices start at 1.

Example: To create a Beagle file called **out.bgl** consisting of 500 individuals in columns 502-1002 of Beagle file called **in.bgl**, enter

```
java -jar cut.jar  in.bgl  1:2  502:1002  out.bgl
```

The 1:2 in the preceding command extract the first two columns of the input Beagle file. The first two columns describe and give the variable name for each line of data.

The cut program can be used with any white-space delimited text file, not just Beagle files.

7.4 paste

The **paste** program is used to combine Beagle files that record data for same markers, but different individuals. The commands for running the **paste** program are

```
java -jar paste.jar  [initial columns]  [in 1]  [in 2]  ...  [out]
```

where [initial columns] is the number (0, 1, 2, ...) of initial, shared columns which are identical in all input files, [in #] are input files, and [out] is the output file. The output file will contain one copy of the initial, shared columns, followed by the remaining non-initial columns of each input file. The input files are pasted together in the order they are given on the command line. Thus, columns of the first input file are given first, followed by the non-initial columns of the second input file, followed by the non-initial columns of the third input file, and so on.

Example: To paste together three Beagle files called **in1.bgl**, **in2.bgl**, and **in3.bgl** which have the same first two columns, and to write the results to the file **out.bgl**, enter

```
java -jar paste.jar  2  in1.bgl  in2.bgl  in3.bg  out.bgl
```

The **paste** program can also be used to paste together output .gprobs files. Output .gprobs files have 3 initial columns giving the marker identifier and the two allele identifiers. Thus, to paste together two output .gprobs files called **in1.gprobs** and **in2.gprobs** which have the same first 3 columns, and to write the results to the file called **out.gprobs**, enter

```
java -jar paste.jar  3  in1.gprobs  in2.gprobs  out.gprobs
```

7.5 splitbeagle

The **splitbeagle** program is used to split a Beagle file into two Beagle files according to the values of a specified variable. The commands for running the **splitbeagle** program are:

```
java -jar splitbeagle.jar [in file] [variable]  [value]  [prefix]
```

where [in file] is an input Beagle file, [variable] is the name of a variable (2nd field in the line) in the input Beagle file, [value] is a value of the variable that is used to split the input Beagle file into two output Beagle files, and [prefix] is the filename prefix for the two output Beagle files. The output file called [prefix].with.bgl contains all the columns with the specified variable value, and the output file [prefix].without.bgl contains all columns without the specified variable value. The output files will have the same first two columns as the input Beagle file. Comment lines (# ...) in the input Beagle file are copied to both output Beagle files. Comment lines are ignored when searching for the line of the input Beagle file with the specified variable name. If more than one line of the input Beagle file has the specified variable name, then the first line will be used to split the input Beagle file.

Example: You cannot have individuals with unknown affection status (0) when performing association analysis with BEAGLE. If the Beagle file is called in.bgl and if the affection status variable is named “CD”, you can remove individuals with unknown affection status using the splitbeagle utility with the command:

```
java -jar splitbeagle.jar  in.bgl  CD  0  in.0
```

Two output Beagle files will be created. The first called in.0.with.bgl will contain all individuals with *unknown* affection status. The second, called in.0.without.bgl will contain all individuals with *known* affection status.

7.6 pseudomarker

Haplotype clusters can be represented as diallelic pseudomarkers (see Section 4.3.2 The P-value file [.pval]). The **pseudomarker** program constructs pseudomarkers from the haplotype clusters in a Beagle output .dag file and writes the pseudomarkers to a phased Beagle file (see Section 2.1 Beagle file format). The phased Beagle file of pseudomarker data can be imported into a statistical software package like R [1] for analysis using standard statistical methods such as logistic regression (to allow for covariates) or analysis of variance for quantitative trait data. The commands for running the **pseudomarker** program are

```
java -jar pseudomarker.jar <arguments>
```

where <arguments> is a space separated list of arguments. Each argument has the format **parameter=value**. There is no white-space between the **parameter** and **=** or between **=** and the **value**. The arguments are

- dag=<Beagle .dag file> where <Beagle .dag file> is the filename of a Beagle output .dag file. The dag argument is required.
- out=<output file> where <output file> is the phased Beagle file that will be created. The markers of the output file will correspond to haplotype clusters in the specified Beagle .dag file. The out argument is required.
- edgecount=<minimum edge count> where <minimum edge count> is a positive integer giving the minimum number of haplotypes in a haplotype cluster defined by an edge *E* that is required to create a diallelic pseudomarker m_E. For more details, see Section 4.2.3 Arguments for association testing. The edgecount argument is optional. The default value is edgecount=1.
- othercount=<minimum other count> where <minimum other count> is a nonnegative integer giving the minimum number of haplotypes on the set of edges that merge with an edge *E* that is required to create a diallelic pseudomarker m_E. For more details, see Section 4.2.3 Arguments for association testing. Edges E_1 and E_2 are said to merge if E_1 and E_2 point to the same child node. The othercount argument is optional. The default value is othercount=0.

The edgecount and othercount arguments are similar to the arguments of the same name in Section 4.2.3 Arguments for association testing, but their default values are different. With default edgecount and othercount parameters, the pseudomarker program creates **multi-allelic** pseudomarkers which correspond to **markers** in the specified .dag file. With non-default values for the edgecount or othercount parameter, the pseudomarker program creates **diallelic** pseudomarkers which correspond to **edges** in the specified .dag file.

If the edgecount and othercount arguments are not used (or if their default values are specified), a multi-allelic pseudomarker is created for each level (i.e. marker) of the graphical model. In the output Beagle file, the multi-allelic pseudomarkers will have the same name and order as the markers in the specified .dag file. For a given marker, a haplotype has allele *k* (*k* = 1, 2, . . .), if it is in the *k*-th edge that connects the given marker with the next marker in the specified .dag file.

If at least one of the edgecount and othercount parameters is set to a non-default value, diallelic pseudomarkers are created for some of the edges. (This is differs from the preceding paragraph, where multi-allelic psedomarkers are created for each marker). A diallelic pseudomarker is created for an edge only if the edge contains at least edgecount haplotypes and only if the edge merges with edges whose haplotype clusters contain at least othercount haplotypes. The name of the diallelic pseudomarker has the format **marker_node**.**allele** where **marker** is the name of the marker that identifies the level of the parent node of the edge, **node** is the parent node number for the edge, and **allele** is the marker allele that labels the edge (see Section 4.3.2 The P-value file [.pval] or [2, 4]). A haplotype has allele 2 if it is in the haplotype cluster corresponding to the edge and has allele 1 if it is not in the haplotype cluster.

7.7 cluster2haps

The **cluster2haps** program is used to identify the haplotypes that are present in a haplotype cluster from the Beagle software program. The **cluster2haps** program prints out all allele sequences that are present in a specified haplotype cluster for increasingly large marker windows, tests the allele sequences for association with the trait status, and reports the P-values

The command line syntax for the program is similar to the syntax for Beagle:

```
java -Xmx600m -jar cluster2haps.jar <arguments>
```

where <arguments> is a space separated list of arguments. Each argument has the format **parameter=value**. There is no white-space between the **parameter** and = or between = and the **value**. The arguments are

- dag=<dag file> where <dag file> is the filename of the Beagle output .dag file giving the graphical haplotype frequency model. The dag argument is required.
- phased=<phased BEAGLE file> where <phased BEAGLE file> is the filename of the phased Beagle file used to build the graphical haplotype frequency model of haplotype structure. The markers in the phased file must be in chromosomal order, and the phased Beagle file must contain an affection status line giving the affection status for each allele of each individual. The data argument is required.
- trait=<trait ID> where <trait ID> is the name of the affection status variable (A <trait ID> ...) used in the association analysis. The trait argument is required.
- out=<output file> where <output file> is the name of the output file for the analysis. The out argument is required.
- marker=<marker ID> where <marker ID> is the identifier of the marker locus for the haplotype cluster given in the .pval file. The marker argument is required.

- parent=<parent node> where <parent node> is the parent node number for the haplotype cluster. The parent node number is given before the dot (".") in the Allele column of the .pval file. The parent argument is required.
- allele=<allele id> where <allele id> is the identifier of the marker allele for the haplotype cluster. The marker allele for the haplotype cluster is given after the dot (".") in the Allele column of the Beagle output P-value (.pval) file. The allele argument is required.
- maxhaps=<maximum haplotypes> where <maximum haplotypes> controls the size of the marker windows that will be considered. The largest marker window considered will be the smallest window for which the number of distinct haplotypes in the clusters is greater than or equal to the maximum number of haplotypes. The maxhaps argument is optional. The default value is maxhaps=8.

cluster2hap has 7 required arguments: 3 arguments to specify files (dag, phased, and out), 1 argument to specify the affection status variable (trait), and 3 arguments to specify the haplotype cluster (marker, parent, and allele).

Example 4 - A sample cluster2haps analysis

Here is an example using quality-control filtered Wellcome Trust Case Control consortium type 1 diabetes data for the IL2RA region [7]. The line in the Beagle output P-value (.pval) file for the most significantly haplotype cluster in the region is

Marker	Allele	allelic_p	min_p	min_p_perm
rs12722489	2.C	5.104e-09	5.104e-09	0.0009990

We can use cluster2haps to analyze this significantly associated haplotype cluster. The command line for the cluster2haps program is:

```
java -jar cluster2haps.jar dag=ex.dag data=ex.phased trait=T1D out=cluster.out
marker=rs12722489 parent=2 allele=C
```

The cluster localizes to the marker rs12722489. The "Allele" field for the haplotype cluster gives the parent node number (2), and the allele (C) for the haplotype cluster. In this example the .dag file was "ex.dag", the Beagle file of phased haplotypes was "ex.phased", the name of the affection status variable was "T1D", and the output file was "cluster.out".

An excerpt of the output file from this **cluster2haps** analysis with results for the three shortest marker windows is given below. For each marker window that ends with marker rs12722489, the output file gives

1. The allele sequences that are contained in the haplotype cluster
2. For each allele sequence in 1, the number of haplotypes with the allele sequence.
3. For each allele sequence in 1, the number (and proportion) of haplotypes with the allele sequence that are in the haplotype cluster
4. For each allele sequence in 1, the P-value from Fisher's exact allelic test of association of the allele sequence with the trait.

Excerpt from cluster2haps output:

```
9802 haplotypes:          3926  cases / 5876 controls
                           971  in cluster / 8831 not in cluster

Markers:  rs12722489
allele sequence                    count               # in cluster            P-value
C                                   8147               971  (11.9%)              0.492

Markers:  rs17149458  rs12722489
allele sequence                    count               # in cluster            P-value
T C                                 8147               971  (11.9%)              0.492

Markers:  rs2104286  rs17149458 rs12722489
allele sequence                    count               # in cluster            P-value
C T C                                989               971  (98.2%)           1.02e-08
```

From the preceding output excerpt, we see that for the 3 shortest marker windows there is only one allele sequence present in the haplotype cluster. For the C-T-C allele sequence (for markers rs2104286 rs17149458 rs12722489), 971 of the 989 of the haplotypes with this allele sequence are in the haplotype cluster, and the C-T-C allele sequence is strongly associated with type 1 diabetes ($p = 1.02 \times 10^{-8}$). Thus the C-T-C haplotype accounts for nearly the entire association signal for the haplotype cluster. The output also shows that the haplotype counts for the C and T-C allele sequences are the same. Thus the middle T allele in the C-T-C sequence is not necessary (due to linkage disequilibrium), and the C-C haplotype (rs2104286-rs12722489) gives a simpler characterization of the haplotype cluster.

8 References

1. R Development Core Team, *R: A Language and Environment for Statistical Computing*. 2006, Vienna, Austria: R Foundation for Statistical Computing.
2. Browning, B.L. and S.R. Browning, *Efficient multilocus association testing for whole genome association studies using localized haplotype clustering.* Genetic Epidemiology, 2007. **31**(5): p. 365-75.
3. Browning, B.L. and S.R. Browning, *Haplotypic analysis of Wellcome Trust Case Control Consortium data.* Human Genetics, 2008. **123**(3): p. 273-80.
4. Browning, S.R., *Multilocus association mapping using variable-length Markov chains.* American Journal of Human Genetics, 2006. **78**(6): p. 903-13.
5. Besag, J. and P. Clifford, *Sequential Monte-Carlo p-values.* Biometrika, 1991. **78**(2): p. 301-304.
6. Schaffner, S.F., et al., *Calibrating a coalescent simulation of human genome sequence variation.* Genome Research, 2005. **15**(11): p. 1576-83.
7. The Wellcome Trust Case Control Consortium, *Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls.* Nature, 2007. **447**(7145): p. 661-78.