

Documentation for BEAGLE 2.1

Brian L. Browning and Sharon R. Browning

Department of Statistics
The University of Auckland
Auckland
New Zealand

September 14, 2007

CONTENTS

1. Introduction	2
2. Citing BEAGLE	2
3. Files in the BEAGLE software distribution	3
4. BEAGLE file format	3
5. Running BEAGLE	5
6. Output files	9
7. Using BEAGLE with large data sets	13
8. An example BEAGLE analysis	14
References	15
Appendix A. Utility programs for creating files in BEAGLE format	15

1. INTRODUCTION

BEAGLE is a state of the art software program for accurately inferring haplotypes from genotypes of unrelated individuals and for genetic association analysis. BEAGLE provides haplotype phasing, powerful multilocus association analysis, and permutation testing for whole genome association studies with hundreds of thousands of markers genotyped on thousands of samples. BEAGLE can

- phase genotype data (i.e. infer haplotypes).
- infer missing genotype data.
- build a graphical model of the linkage disequilibrium structure.
- cluster similar haplotypes at each location in the genome.
- test single markers and haplotypes clusters for association with a binary trait.
- perform permutation testing to assess statistical significance.

BEAGLE is written in java and runs on most computing platforms (e.g. Windows, Linux, Solaris).

2. CITING BEAGLE

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

If you use BEAGLE for **inferring haplotype phase** or **missing data inference**, please cite

S R Browning and B L Browning. Rapid and accurate haplotype phasing and missing data inference for whole genome association studies using localized haplotype clustering. *Am J Hum Genet.* In Press.

If you use BEAGLE for **association testing**, please cite

B L Browning and S R Browning (2007) Efficient multilocus association mapping for whole genome association studies using localized haplotype clustering. *Genet Epidemiol* 31:365-375.

If appropriate, you may also want to cite the paper that introduced the graphical linkage disequilibrium model used by BEAGLE:

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

3. 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:

1. **beagle.jar**, the executable file for running BEAGLE (see Section 5).
2. **beagle_2.1.pdf**, the documentation for Beagle 2.1.
3. A folder called **example** containing files used in the example BEAGLE analysis in Section 8. The **example** folder contains two input files (**example.bgl** and **example.trait**), and five output files (**example.log**, **example.phased**, **example.dag**, **example.null**, and **example.pval**).
4. A folder called **utility** containing three utility programs that create files in BEAGLE format. BEAGLE format is described in Section 4. The three utility programs are:
 - a. **unphased2beagle.jar**, a utility program that creates an unphased BEAGLE file from a data file and a pedigree file in linkage or QTDT format (see Appendix A.1)
 - b. **phased2beagle.jar**, a utility program that creates a phased BEAGLE file from an output file of a haplotype phasing program (see Appendix A.2).
 - c. **pseudomarker.jar**, a utility program that creates a phased BEAGLE file of pseudo-markers from a BEAGLE output model (.dag) file (see Appendix A.3).

4. BEAGLE FILE FORMAT

A BEAGLE file has a simple format. The data for each genetic marker or affection status is given in a single line. The first column contains single characters on each line (“M” or “A”) describing the information on each line: “M” for marker allele data and “A” for affection status. The second column gives the name for the marker or affection status whose data is given on each line. Columns 3-4 give the marker alleles or affection status for the first individual, columns 5-6 give the marker alleles or affection status for the second individual, and so on. Note that for a line of affection status data, the affection status is given twice for each individual (once in each column for the individual), and that the affection status for both alleles of an individual must be the same (**1** for unaffected individuals and **2** for affected individuals). The markers in the BEAGLE file must be in chromosomal order.

The BEAGLE file can have either phased or unphased data. For an unphased BEAGLE file, each pair of columns (beginning with columns 3-4) gives the genotype for each individual. For a phased BEAGLE file, each pair of columns in the phased data file gives the two haplotypes for each individual. Thus column 3 is one haplotype and column 4 is the other haplotype for the first individual. When you use BEAGLE to phase genotype data, the output phased data file (.phased) is a phased BEAGLE file.

Affection status data are not used for phasing or for building the graphical model of the haplotype structure, but affection status data are necessary for association testing. The

affection status data can either be in the same file as the marker data or in a separate BEAGLE file.

You can also include comment lines in your BEAGLE file. A comment line is any line whose first field is not “M” or “A”. Comment lines are ignored by BEAGLE and can have any format. To ensure compatibility with later versions of BEAGLE, we recommend using the hash character, “#”, as the first field of a comment. You can use the “#” character to comment out data lines of a BEAGLE file or to include additional information (e.g. sample identifiers) in a BEAGLE file.

Below is an example of a BEAGLE file for three individuals genotyped for three markers:

```
# sampleID      1001  1001  1002  1002  1003  1003
A diabetes      1     1     2     2     2     2
M rs1248696     1     3     3     3     3     3
M rs2289310     2     2     2     2     1     2
M rs2289311     4     4     2     2     4     2
```

The first line (# ...) is a **comment line** that is ignored by BEAGLE. The second line (A ...) is an **affection status line** that gives the diabetes affection status for each allele (1 = unaffected, 2 = affected). The last three lines (M ...) are **marker lines** that give marker alleles for the three markers (**rs1248696**, **rs2289310**, and **rs2289311**).

In this example BEAGLE file, the first individual (columns 3-4) is unaffected, and the second and third individuals (columns 5-6 and 7-8) are affected. If this example file contains unphased data then the first individual has genotypes 1/3, 2/2, 4/4, the second individual has genotypes 3/3, 2/2, 2/2, and the third individual has genotypes 3/3, 1/2, 4/2 for markers rs1248696, rs2289310, and rs2289311 respectively. If this example file contains phased data, then the first individual has haplotypes 124 and 324, the second individual has both haplotypes equal to 322, and the third individual has haplotypes 314 and 322.

BEAGLE imposes very few constraints on your data files. For example:

- Unphased data can have missing alleles or genotypes (see the **missing** parameter described in Section 5.1). After phasing data, all missing data is imputed.
- The fields on each line can be separated by one or more white space characters (e.g. any combination of one or more spaces and tabs).
- Markers can have up to 128 different alleles. In particular, triallelic SNPs and microsatellite markers can be used.
- The allele for each marker can be any sequence of characters that does not contain white space.

It is also possible to use BEAGLE to analyze transmitted and untransmitted haplotypes from trio studies (affected individuals and their parents). Use the **diploypes=false** option described in Section 5.4, and code the affection status of transmitted haplotypes as **2** and the affection status of untransmitted haplotypes as **1**. When using the **diploypes=false** option, 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.

5. RUNNING BEAGLE

BEAGLE is written in java and requires a java interpreter. A java interpreter is probably already installed on your computer. However, if it is not installed or if it is an old version, the java interpreter can be downloaded free of charge from the java.sun.com web site. You will need to download and install the standard edition (SE) Java Runtime Environment (JRE) 5.0 (or later version).

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

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

where `<arguments>` is a space separated list of arguments, and each argument has the format `parameter=value`. There is no white-space between the parameter and the “=” character or between the “=” character and the value.

If you are analyzing more than 3000 diploid individuals, you may need to increase the maximum amount of heap memory available to the java interpreter. This is set by the `-Xmx` argument. See Section 7 for details.

There are four types of command line arguments: arguments for specifying files (Section 5.1), arguments for phasing the data (Section 5.2), argument for building the model (Section 5.3), and arguments for association testing (Section 5.4).

If you are a new BEAGLE user, we recommend that you start with using only the arguments for specifying files given in Section 5.1. The other BEAGLE arguments are optional and have sensible default values.

5.1. Argument for specifying files.

- `unphased=<unphased BEAGLE file>` where `<unphased BEAGLE file>` is the name of the BEAGLE file containing unphased genotype data. The BEAGLE file format is described in Section 4. When the `unphased` parameter is used, BEAGLE will phase the data. If the `trait` parameter is also specified, BEAGLE will create a graphical model of the phased data and perform association testing. Either the `unphased` or the `phased` argument (described next) is required.
- `phased=<phased BEAGLE file>` where `<phased BEAGLE file>` is the name of the BEAGLE file containing phased haplotype data (e.g. the phased data file described in Section 6.2). The BEAGLE file format is described in Section 4. When the `phased` argument is used BEAGLE will create a graphical model of the phased data and will perform association testing if the `trait` parameter is also specified. Do not use the `phased` parameter if the data is not yet phased, use the `unphased` parameter (described above) instead.
- `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 if the `unphased` argument is used, and prohibited if the `phased` argument is used.

- **trait**=<trait file> where <trait file> is the name of the BEAGLE file containing affection status data. The BEAGLE file format is described in Section 4. The trait file can be the same file that was specified with the **unphased** or **phased** parameter or it can be a different file. If the trait file contains multiple affection status lines (i.e. lines whose first field is “A”), then only the first affection status line will be used for association testing. The **trait** argument is optional. If the **trait** argument is used then association testing will be performed. A graphical model of the linkage disequilibrium structure will be created (Section 6.6) if the **trait** or **phased** arguments is used.
- **out**=<output prefix> where <output prefix> is the prefix for the BEAGLE output files. For example, if **out=beagle** is specified, the analysis summary is written to the file **beagle.log**. The **out** argument is required. See Section 6 for a description of the different output files that are created by BEAGLE.

5.2. Arguments for phasing the data.

- **nsamples**=<number of samples> where <number of samples> is a 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**.

The default value for **nsamples** performs well for samples of 100 or more individuals. However, to fine tune the performance of BEAGLE we suggest that you set **nsamples** so that the product of the **nsamples** parameter and the number of individuals is between 2000 and 4000. For example, we use **nsamples=25** for phasing 100 individuals, **nsamples=4** for phasing 1000 individuals and **nsamples=1** for phasing 3000 or more individuals.

- **pilot**=<true/false> where <true/false> is **true** if you want to perform a pilot study of a subset of the data, and <true/false> is **false** if you want to phase all the data. The pilot study measures the accuracy of the haplotype phasing with the specified command line arguments by masking genotypes prior to phasing and comparing the inferred alleles to the true alleles. The **pilot** parameter is optional. The default value is **pilot=false**. When a pilot study is performed, results are reported in the output log file (.log), and only a log file is generated (see Section 6).

A pilot study randomly selects 100 subsets of markers, and for each subset, 1% of the genotypes are randomly selected and set to missing. The subsets are phased and masked data is imputed using the phasing parameters specified on the command line. The accuracy of the phasing is measured by the accuracy of the imputed masked genotypes. Each subset of markers is required to contain at least 200 consecutive markers and at least 500,000 genotypes (if this is not possible, each subset is the whole data set). For large data sets with several thousand samples, the pilot

study may take several hours to complete when using the default settings of the `niterations` and `nsamples` parameters.

- `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`. The random number generator is used when sampling haplotype pairs during haplotype phasing and when permuting the trait status during permutation testing.

5.3. Arguments for building the model. The `scale` and `shift` parameters control the number of haplotype clusters in the graphical model for the phased data. The `scale` and `shift` parameters are not used for phasing genotype data.

Having too many or too few haplotype clusters can reduce the power of association testing. A recent simulation study showed that the default values for the `scale` and `shift` parameters performed well for samples of 2000 individuals [2]. However, if your sample size is small (say less than 400 individuals) you may want to decrease the `scale` and `shift` parameters to increase the number of clusters.

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 then nodes A and B will not merge if their similarity score [3, 2] is greater than

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

where m is 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 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 a nonnegative floating point number less than or equal to 1.0 giving the shift parameter used when deciding whether to merge 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.

5.4. Arguments for association testing.

- `test=<association tests>` where `<association tests>` is one or more characters from the set `{a, r, d, o}` 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 a 2×2 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 3 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 `diploypes=false`. See the `diploypes` argument in this section for more details.

- `seed=<random seed>`. See the `seed` argument in Section 5.2.
- `nperms=<number of permutations>` where `<number of permutations>` is a non-negative integer giving the number of permutations of the affection status used for permutation testing. Permutation testing is used to determine significance levels that account for all the haplotypes and single markers that are tested (see Section 6.4). You can skip permutation testing by setting `nperms=0`. The `nperms` argument is optional. The default value is `nperms=1000`. The running 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. Each haplotype cluster is defined by an edge of the graphical model. See [3] 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. See [3] for more details. The `othercount` argument is optional. The default value is `othercount=1`. Increasing the other count 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 usually is not recommended because it will result in testing non-merging edges.
- `diploypes=<true/false>` where `<true/false>` is `true` if the trait 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 trait status is permuted for individual haplotype (rather than for haplotype pairs). Only the allelic test can be performed when `diploypes=false` (see the `test` argument in this section). The `diploypes` argument is optional. The default value is `diploypes=true`. The `diploypes` argument should not be used unless the phased data consist of transmitted and untransmitted haplotypes as described in the last paragraph of Section 4.

5.5. Arguments that are not intended for general use.

- `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`. Increasing the `niterations` parameter beyond the default gives very little increase in accuracy.
- `nimputations=<number of imputations>` where `<number of imputations>` is a nonnegative integer giving the number of haplotype pairs to sample for each individual conditional on the graphical model and the genotypes for the sample. If $k > 0$ imputations are specified, the sampled haplotypes are written to a file with a “.k.imp” extension where `k` is the specified number of imputations (see Section 6). The `nimputations` argument is optional and can only be used when the `unphased` argument is used (Section 5.1). The default value is `nimputations=0`. If multiple runs of BEAGLE are used to sample haplotypes, a different seed should be specified for each run (see the `seed` parameter in section 5.2).
- `maxwindow=<maximum window size>` where `<maximum window size>` is a positive integer giving the maximum number of consecutive markers that will be considered when deciding whether the haplotype clusters represented by those nodes should be merged. The `maxwindow` argument is optional. The default value is `maxwindow=500`. The default value should be sufficiently large for almost all 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.

6. OUTPUT FILES

Depending on the parameters for the analysis, BEAGLE can produce up to six output files. Most users will be interested in only two files: the **log file** (.log), and either the **phased data file** (.phased) or the **p-value file** (.pval) depending on whether your objective is phasing data or association testing.

The remaining three files, the imputed haplotypes file (.imp), the null p-value file (.null) and the model file (.dag), will be useful to some, but not most, users.

6.1. 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, a description of the command line arguments, a list of the command line arguments for the analysis, and the running time for the analysis.

If the `trait` parameter is used (Section 5.4), the log file gives a list of all markers or haplotype clusters from the p-value file (.pval) with a permutation p-value less than 0.2. If the `pilot=true` argument (Section 5.2) is specified, then the log file gives the estimated accuracy of inferred alleles and genotypes with 95% confidence intervals.

6.2. The phased data file (.phased). A phased data file is generated when the `unphased` parameter is used (Section 5.2). The phased data file is a phased BEAGLE file (see Section 4) and includes the marker data, but not affection status data or comments. The phased data gives the most likely haplotype pair for each individual conditional upon the genotypes for the individual and the haplotype model generated in the last iteration of the phasing algorithm (see [4] for details).

6.3. The imputed haplotype file (.imp). The imputed data file is generated when the `nimputations` parameter is greater than 0 (Section 5.5). The imputed haplotype file is a phased BEAGLE file (see Section 4) and includes marker data, but not affection status data or comments. Unlike the phased data file (Section 6.2) which gives the most likely haplotype pair for each individual, the imputed haplotype file gives randomly sampled haplotypes where the sampling is conditional on the genotypes for the individual and the haplotype model generated in the last iteration of the phasing algorithm. The `nimputations` parameter gives the number of haplotype pairs sampled per individual. For example, if `nimputations=3` then columns 3-4, 5-6, 7-8 give three imputed haplotype pairs for the genotypes in columns 3-4 of the unphased data, columns 9-10, 11-12, 13-14 give three imputed haplotype pairs for the genotypes in columns 5-6 of the unphased data, and so on (see Section 4).

The `nimputations` parameter is not intended for general use, and standard association analysis that assumes that all haplotype pairs are independent is not valid when the number of imputations is greater than 1.

6.4. The p-value file (.pval). The p-value file records the p-values from testing markers and haplotype clusters for association with the trait status.

Haplotype clusters are defined by edges of graphical model. Each haplotype defines a path between the initial and terminal node of the graph [3]. For each edge E in the graphical model we define the diallelic **pseudo-marker** m_E to be 2 for each haplotype whose path includes edge E and to be 1 for each haplotype whose path does not include edge E .

Representing haplotype clusters as pseudo-markers 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 Appendix A.3) can be used to create a phased BEAGLE file of pseudo-markers representing the haplotype clusters in the graphical model. The phased BEAGLE file of pseudo-markers created by the `pseudomarker` program can be imported into a program like R [5] 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 pseudo-marker for association with the trait status. First, the markers in the BEAGLE file are tested in the order they appear in the BEAGLE file (see Section 4). Then the pseudo-markers defined by the haplotype clusters (edges) of the graphical model are tested. The `edgecount` and `othercount` parameters described in Section 5.4 determine which pseudo-markers are selected for testing.

The first and last few lines of the p-value file look like this:

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 5.4 for more details). For pseudo-markers 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 the previous p-value file excerpt). The marker identifier, the parent node number, and the marker allele uniquely determines the edge of the graphical model (see Section 6.6 and [3, 2] for more discussion of the graphical model).

After the marker and allele fields, the next columns give the p-values for allelic, recessive, dominant, and overdominant tests. Columns corresponding to tests which were not performed are omitted (Section 5.4). 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 pseudo-marker 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 pseudo-marker 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 [1].

Given a set of tests (specified with the `test` parameter), the permutation test randomly permutes the trait status and tests the marker alleles and pseudo-marker alleles for association with the permuted trait status. If `diploypes=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 `diploypes=false` argument (see Section 5.4) must be used so that the trait status is permuted for the haplotypes rather than for the individuals.

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 pseudo-markers and all tests) is saved and written to the null p-value file (see Section 6.5). If a marker or pseudo-marker has a minimum p-value of p_0 (minimized over all tests for that marker or pseudo-marker) 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 pseudo-marker 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 $(k + 1)/(N + 1)$ [1]. 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 for each permutation of the trait status.

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 (.log). The log file contains a list of all markers and pseudo-markers with a permutation p-value less than 0.2.

6.5. The null p-value file (.null). The null p-value file lists the minimum p-values (minimized over all marker and pseudo-marker alleles and all tests) observed when testing for association of a randomly permuted trait status with the marker and pseudo-marker alleles (see Section 6.4). One p-value is listed per line. 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 5.4). The null p-value file can be used to obtain an empirical distribution of the minimum p-value under the null hypothesis. If `nperms=0` the null p-value file will be empty. The sequence of permutations of the trait status is determined by the `seed` argument (see Section 5.4).

6.6. 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 (see [3, 2] for more details).

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 $0, 1, 2, \dots, M - 1$.
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 that are in the haplotype cluster. The haplotypes are numbered $0, 1, 2, \dots$ in the order they appear as columns in the phased BEAGLE file (Section 4), so 0 refers to column 3 in the phased BEAGLE file, 1 refers to column 4 in the phased BEAGLE file, and so on. The phased BEAGLE file is either the phased BEAGLE file (section 6.2) created by BEAGLE when the `unphased` parameter is used (Sections 5.1) or is the phased BEAGLE file specified with the `phased` parameter (Section 5.1).

7. USING BEAGLE WITH LARGE DATA SETS

If you want to perform association testing on data from multiple chromosomes (e.g. a whole genome association study), first divide the data by chromosome, and phase the data for each chromosome separately. You can phase the data without performing association testing by omitting the `trait` argument (see Section 5.1).

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 using the `phased` and `trait` arguments (see Section 5.1). It is important to test all markers in a single run so that BEAGLE can perform permutation testing to determine statistical significance.

If you are phasing more than 2000 individuals and more than 100 markers, you should set the `nsamples` parameter to 1 (`nsamples=1`) because additional samples will significantly increase memory usage without significantly increasing the phasing accuracy.

You can increase the amount of memory available to the java interpreter by changing the `-Xmx600m` argument in the command line. In general, 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.

The memory requirements for building the model and association testing depend on the number of individuals with phased haplotypes, but not the number of markers. You will need approximately 200 Mb of memory (RAM) per 1000 diploid individuals for model building and for association testing. For example, if you are using BEAGLE for association testing

of phased data from 7000 individuals, expect BEAGLE to require 1400 Mb of memory and use `-Xmx1400m` in the command line argument.

The memory requirement for phasing genotype data depend on the number of genotyped individuals and the number of markers. You will need approximately 600 Mb of memory to phase 3,000 diploid individuals genotyped for 50,000 markers on a single chromosome, and memory usage is approximately linear in the number of individuals and the number of markers. For example, if you have 6000 diploid individuals genotyped for 100,000 markers on a single chromosome, then expect BEAGLE to use $(6000/3000)*(100000/50000)*600 = 2400$ Mb of memory. If necessary one can break a chromosome into a few segments and phase each segment separately to reduce the amount of computer memory required.

You may be able to obtain faster running times by permitting the java interpreter to use more memory than BEAGLE requires. For example, if BEAGLE only requires 400 Mb of memory, allocating 1500 Mb of memory (`-Xmx1500m`) on a computer with 2000 Mb of memory may reduce running times.

8. AN EXAMPLE BEAGLE ANALYSIS

Files from an example BEAGLE analysis are included in this software distribution. The example data are 200 markers for 4000 haplotypes (1000 case and 1000 control individuals). The example data were generated using the *Cosi* program, version 1.0 [6].

The example files are

- | | |
|-------------------|---|
| 1. example.bgl | BEAGLE file with unphased genotype data (see Section 4) |
| 2. example.trait | BEAGLE file with affection status data (see Section 4) |
| 3. example.log | BEAGLE output log file (see Section 6.1) |
| 4. example.phased | BEAGLE output phased file (see Section 6.2) |
| 5. example.pval | BEAGLE output p-value file (see Section 6.4) |
| 6. example.null | BEAGLE output null p-value file (see Section 6.5) |
| 7. example.dag | BEAGLE output model file (see Section 6.6) |

The BEAGLE output files (files 3-7) are created from the BEAGLE files (files 1-2) using the command:

```
java -Xmx600m -jar beagle.jar unphased=example.bgl missing=? trait=example.trait
out=example
```

The BEAGLE output log file (example.log) contains the following excerpt listing the 3 marker alleles and 3 pseudo-markers 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	0.1239
m23612	0	0.0002015	0.0002015	0.06693
m24828	0	0.0006412	0.0006412	0.1878
m23171	1.1	7.046e-05	7.046e-05	0.03596
m23892	6.1	4.559e-05	4.559e-05	0.02498
m24830	1.0	0.0006412	0.0006412	0.1878

6 alleles with permutation p-value < 0.2

Two of the six alleles have permutation p-values of $p = 0.03596$ and $p = 0.02498$ which are significant at the $\alpha = 0.05$ level after accounting for multiple testing. Allele 1.1 of marker m23171 is the haplotype cluster with parent node 1 and labeled with allele 1 at the marker m23171 level of the graphical model. Allele 6.1 of marker m23892 is the haplotype cluster with parent node 6 and labeled with allele 1 at the marker m23892 level of the graphical model.

Searching the output BEAGLE model file (example.dag) for m23171 reveals that marker m23171 is level 117 of the graphical model and that allele 1.1 represents a cluster of 281 haplotypes defined on line 892 of example.dag. Similarly, marker m23892 is level 128 of the graphical model and allele 6.1 represents a cluster of 201 haplotypes defined on line 1054 of example.dag.

The association can be localized and the risk haplotype(s) corresponding to specific edges can be identified using the haplotype clusters in the BEAGLE output model file (example.dag) as illustrated in [3, pp. 908-9], or by using another genetic analysis program.

REFERENCES

- [1] J. Besag and P. Clifford. Sequential Monte Carlo p-values. *Biometrika*, 78:301–304, 1991.
- [2] B. L. Browning and S. R. Browning. Efficient multilocus association mapping for whole genome association studies using localized haplotype clustering. *Genet Epidemiol*, in press.
- [3] S. R. Browning. Multilocus association mapping using variable-length Markov chains. *American Journal of Human Genetics*, 78:903–913, 2006.
- [4] S. R. Browning and B. L. Browning. Rapid and accurate haplotype phasing and missing data inference for whole genome association studies using localized haplotype clustering. In preparation.
- [5] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.
- [6] S. F. Schaffner, C. Foo, S. Gabriel, D. Reich, M. J. Daly, and D. Altshuler. Calibrating a coalescent simulation of human sequence variation. *Genome Research*, 15:1576–1583, 2005.

APPENDIX A. UTILITY PROGRAMS FOR CREATING FILES IN BEAGLE FORMAT

A.1. **unphased2beagle**. The **unphased2beagle** 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 explained at <http://www.sph.umich.edu/csg/abecasis/QTDT/docs/data.html>.

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

The lines of the data file correspond to the variables in the pedigree file (in the order they appear as columns in the pedigree file, beginning with column six). Each line of the data file has two fields. The first field is a single character identifying 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 `unphased2beagle` since the first column of the data file must be a single character).

The `unphased2beagle` program creates a BEAGLE format file whose first two columns are the first two columns of the data file. The third and fourth columns give the two alleles for the first individual in the pedigree file, the fifth and sixth columns give the two alleles for the second individual in the pedigree file, and so on.

For example, suppose the data file is

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

Then the pedigree file will have five fixed columns and seven variable columns (one column each for the A, T, and C variables and two columns each for the two M variables).

If the pedigree file associated with the data 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:

```
A  diabetes      1  1  2  2  2  2
M  rs1248696    A  G  G  G  G  G
M  rs2289311    T  T  T  T  T  C
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
```

Recall that lines of the beagle file whose first character is not A or M will be treated as comments (see Section 4).

To run the `unphased2beagle` program enter

```
java -Xmx600m -jar unphased2beagle <arguments>
```

where `<arguments>` is a space separated list of arguments and each argument has the format `parameter=value`. There is no white space between the parameter and the “=” character or between the “=” character 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. The markers in the data file must be in chromosomal order. The `data` argument is required.
- `beagle=<unphased BEAGLE file>` where `<BEAGLE file>` is the filename of the BEAGLE file that will be created from the the pedigree and data files. The `beagle` argument is required.
- `skip=<number of columns to skip>` where `<number of columns to skip>` is the number of columns of fixed data in the pedigree file. Typically the first five columns are fixed; however, if you have retained the subject identifier and gender columns out of the first five columns in the pedigree file, and deleted the pedigree identifier, father identifier, and mother identifier columns, then set `skip=2`.

A.2. **phased2beagle**. It is also possible to use BEAGLE with the output of other haplotype phasing programs, provided that missing data is imputed. Haplotype phasing programs typically write the haplotypes as lines of an output file, but the BEAGLE file has the haplotypes as columns. It can be difficult to transpose the rows and columns when the data file is larger than the available computer memory, so we have included a utility program, `phased2beagle` that creates a BEAGLE file from a file of phased haplotype data. To run the `phased2beagle` program enter

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

where `<arguments>` is a space separated list of arguments and each argument has the format `parameter=value`. There is no white space between the parameter and the “=” character or between the “=” character and the value. The arguments are

- `phased=<phased data file>` where `<phased data file>` is the filename 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 no white space, each character 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=<marker file>` where `<marker 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 labeled `marker_1`, `marker_2`, `marker_3`, and so on.

A.3. pseudomarker. Haplotype clusters can be represented as diallelic pseudo-markers (see Section 6.4). The `pseudomarker` program constructs pseudo-markers from the haplotype clusters in a BEAGLE output `.dag` file and writes the pseudo-markers to a phased BEAGLE file (described in Section 4). The BEAGLE file of pseudo-markers can be imported into a statistical software package like R [5] for analysis using standard statistical methods such as logistic regression (to allow for covariates) or analysis of variance for quantitative trait data.

To run the `pseudomarker` program enter

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

where `<arguments>` is a space separated list of arguments and each argument has the format `parameter=value`. There is no white space between the parameter and the “=” character or between the “=” character 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 whose markers correspond to haplotype clusters in the BEAGLE output `.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 pseudo-marker m_E (see Section 6.4). 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 pseudo-marker m_E (see Section 6.4). Edges E_1 and E_2 are said to merge if E_1 and E_2 point to the same child node. See [3] for more details. 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 5.4 but the default values are different.

If the `edgecount` and `othercount` arguments are not used (or if they are set equal to their default values), a multiallelic pseudo-marker is created for each level (i.e. marker) of the graphical model. The multiallelic pseudo-marker will have the same name as the marker associated with its level. For a given level (i.e. marker) of the graph, each edge that connects the given level with the next level is assigned a unique allele denoted by a positive integer (1, 2, ...), and each haplotype is assigned the allele of the edge (i.e. haplotype cluster) that contains it.

If at least one of the `edgecount` and `othercount` parameters is set to a non-default value, a diallelic pseudo-marker is created for each edge that contains at least `edgecount` haplotypes and that merges with edges whose haplotype clusters contain at least `othercount` haplotypes. The name of the diallelic pseudo-marker has the format `marker_node.allele` where `marker` is the name of the marker given in the model file (.dag) that identifies the level of the graphical model, `node` is the parent node number for the edge, and `allele` is the marker allele that labels the edge (see Section 6.4 or [3, 2]). 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.