Fast two-stage phasing of large-scale sequence data

Brian L. Browning,[1,2,*] Xiaowen Tian,[3] Ying Zhou,[4] and Sharon R. Browning[2]

Summary

Haplotype phasing is the estimation of haplotypes from genotype data. We present a fast, accurate, and memory-efficient haplotype phasing method that scales to large-scale SNP array and sequence data. The method uses marker windowing and composite reference haplotypes to reduce memory usage and computation time. It incorporates a progressive phasing algorithm that identifies confidently phased heterozygotes in each iteration and fixes the phase of these heterozygotes in subsequent iterations. For data with many low-frequency variants, such as whole-genome sequence data, the method employs a two-stage phasing algorithm that phases high-frequency markers via progressive phasing in the first stage and phases low-frequency markers via genotype imputation in the second stage. This haplotype phasing method is implemented in the open-source Beagle 5.2 software package. We compare Beagle 5.2 and SHAPEIT 4.2.1 by using expanding subsets of 485,301 UK Biobank samples and 38,387 TOPMed samples. Both methods have very similar accuracy and computation time for UK Biobank SNP array data. However, for TOPMed sequence data, Beagle is more than 20 times faster than SHAPEIT, achieves similar accuracy, and scales to larger sample sizes.

Introduction

Haplotype phasing is the estimation of the haplotypes that are inherited from each parent. Genotypes obtained from a SNP array or from sequencing are typically unphased, and statistical methods must be used for inferring the sequence of alleles on each inherited chromosome.

Haplotype phasing is a common analysis because phased haplotypes are required or desirable for many downstream analyses, including genotype imputation,[1–4] detection of deleterious compound heterozygotes,[5] genetic association testing,[6,7] detection of identity-by-descent segments,[8–10] inference of population ancestry at a locus,[11–13] and testing for natural selection.[10,14–16]

The accuracy of haplotype phasing increases with sample size, and this has motivated the development of increasingly powerful phasing methods. The fastPHASE,[17] Beagle,[18] long-range phasing,[19] and Mach[20] methods were among the first methods designed for genome-wide data. The next major advance came from methods such as HAPI-UR,[21] SHAPEIT,[22] and EAGLE[23] that could analyze much larger datasets and whose computation time and memory scaled linearly or nearly linearly with sample size.

Further improvements in computation time came from incorporating methodological ideas from genotype imputation and from data compression, such as the use of a small, custom reference panel for each individual,[24,25] and from the use of the positional Burrows-Wheeler transform[26] for efficiently identifying long shared allele sequences.[27,28]

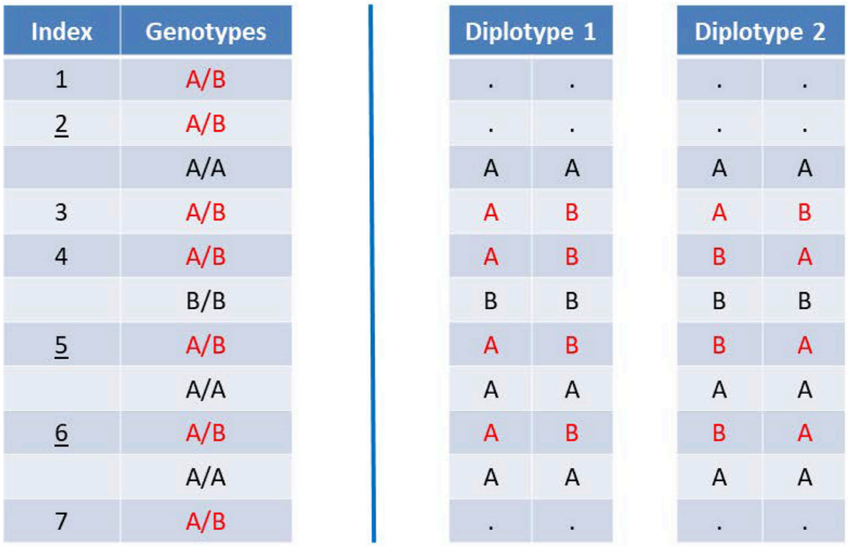
Large sequence datasets with hundreds of millions of markers pose new challenges for phasing methods. In this paper, we present a haplotype phasing method, implemented in Beagle 5.2, that is designed for these data. The method uses built-in marker windowing to limit the data that must be stored in memory. It employs a Li and Stephens hidden Markov model (HMM)[29] with a parsimonious state space of composite reference haplotypes[3] to achieve linear scaling with sample size, and it uses a computationally efficient two-stage phasing algorithm. The two-stage algorithm first phases high-frequency markers by using a progressive phasing methodology that incrementally expands the set of phased heterozygotes. The second stage uses the phased high-frequency markers as a haplotype scaffold for allele imputation and infers phase at low-frequency markers from imputed allele probabilities.[1,30]

The resulting method is computationally fast, multithreaded, and memory efficient. We compare Beagle 5.2 and SHAPEIT 4.2.1 by using default parameters for phasing UK Biobank SNP array data[31] and TOPMed sequence data.[32] We find that the two methods have similar accuracy and computation time on UK Biobank SNP array data. However, Beagle is more than 20 times faster than SHAPEIT on TOPMed sequence data, achieves similar accuracy, and scales to larger sample sizes.

Subjects and methods

The Beagle 5.2 phasing method uses an iterative algorithm. The estimated haplotypes at the beginning of each iteration determine an HMM,[33] which is used for updating the estimated haplotypes.

[1]Department of Medicine, Division of Medical Genetics, University of Washington, Seattle, WA 98195, USA; [2]Department of Biostatistics, University of Washington, Seattle, WA 98195, USA; [3]Statistical Innovation, Oncology Biometrics, AstraZeneca, Gaithersburg, MD 20878, USA; [4]Public Health Sciences Division, Fred Hutchinson Cancer Research Center, Seattle, WA 98109, USA
*Correspondence: browning@uw.edu
https://doi.org/10.1016/j.ajhg.2021.08.005.



Index	Genotypes	Diplotype 1		Diplotype 2	
1	A/B	.	.	.	.
2	A/B	.	.	.	.
	A/A	A	A	A	A
3	A/B	A	B	A	B
4	A/B	A	B	B	A
	B/B	B	B	B	B
5	A/B	A	B	B	A
	A/A	A	A	A	A
6	A/B	A	B	B	A
	A/A	A	A	A	A
7	A/B	.	.	.	.

Figure 1. Two possible diplotypes after heterozygote masking
The left side lists eleven genotypes in chromosome order whose alleles are labeled A and B. The seven heterozygous genotypes are in red font and are indexed in the left column. Indices of three heterozygous genotypes are underlined (2, 5, and 6). Each of these three heterozygous genotypes is "finished," which means that the heterozygote has known phase with respect to the preceding heterozygote. Alleles are labeled so that each heterozygote with known phase has the A allele on the same haplotype as the preceding heterozygote. The right side shows the two possible diplotypes after heterozygote masking when phasing the 4th heterozygote with respect to the 3rd heterozygote. The 2nd heterozygote, which has known phase with respect to the 1st heterozygote, is masked because the 3rd heterozygote has unknown phase with respect to the 2nd heterozygote.

Beagle 5.2 uses the Li and Stephens HMM,[3,20,29,34] which is described in Appendix A.

The HMM state transition probabilities depend on a user-specified effective population size parameter. Because an appropriate value for this parameter may be unknown for some species, Beagle uses the HMM determined by the initial effective population size parameter to estimate and update this parameter via the algorithm described in Appendix B. In the results, we show that this parameter estimation produces good phase accuracy even if the initial parameter value is several orders of magnitude too small or too large.

Marker windows

Beagle uses a sliding marker window. The default window length is 40 cM with 2 cM overlap between adjacent windows. Beagle processes windows in chromosome order and phases the genotypes in each window. If the window is not the first window on the chromosome, Beagle uses the estimated haplotypes from the previous window in the first half of the overlap region.

Beagle's memory requirements can be controlled by adjusting the length of the sliding marker window. We investigate the relationship between window size, computation time, and memory use in the results.

Progressive phasing

Beagle 5.2 employs a progressive phasing algorithm. Each heterozygous genotype is either "in progress" or "finished." Initially, all heterozygotes are in progress. In each iteration, we estimate and update the phase of each in-progress heterozygote with respect to the preceding heterozygote. At the end of each phasing iteration, the most confidently phased in-progress heterozygotes are marked as finished. Once a heterozygote is finished, its phase with respect to the previous heterozygote is fixed and cannot be changed in later iterations. At the end of the final iteration, any remaining in-progress heterozygotes are marked as finished.

Our method for estimating the phase of an in-progress heterozygote produces a ratio that measures the confidence in the estimated phase (see updating phase). In each phasing iteration, we rank the remaining in-progress heterozygotes by this ratio and mark a proportion p of these heterozygotes that are the most confidently phased as finished. The proportion p depends on the individual and the number of heterozygous genotypes the individual has in the marker window. We use the same proportion p in each of the individual's phasing iterations. If there are I phasing iterations ($I = 12$ by default), and if an individual has L in-progress heterozygotes at the start of the phasing iterations, we choose p to solve $L(1-p)^I = 1$, so that $p = 1 - \sqrt[I]{1/L}$. With this choice of p, the individual will have approximately one in-progress heterozygote remaining at the end of the final phasing iteration.

Updating phase

In each iteration, we update the phase of each in-progress heterozygote with respect to the preceding heterozygote. When more than one computing thread is used, phase updating is parallelized by individual. When phasing a target in-progress heterozygote, we mask (i.e., set to missing) all heterozygous genotypes for the individual except the target heterozygote, the preceding heterozygote, and heterozygotes whose phase is known with respect to the target heterozygote or with respect to the preceding heterozygote. For example, if the heterozygote following the target heterozygote is finished, the target heterozygote will have known phase with respect to the following heterozygote. Figure 1 provides an example of this heterozygote masking. After heterozygote masking, there are only two diplotypes consistent with the non-masked genotypes, and these two diplotypes correspond to the two possibilities for the phase of the target heterozygote with respect to the preceding heterozygote.

We calculate the probability of each haplotype in each of the two diplotypes by using the HMM forward-backward algorithm[33] and Equation A4 in Appendix A evaluated at the marker preceding the target heterozygote. We assume Hardy-Weinberg equilibrium and multiply the probabilities of the two haplotypes in a diplotype to obtain the diplotype probability. The diplotype with larger probability determines the updated phase of the target heterozygote with respect to the previous heterozygote. The ratio of the larger diplotype probability to the smaller diplotype probability is a measure of confidence in the inferred phase.

We avoid duplicate calculations when calculating haplotype probabilities by computing, storing, and reusing HMM forward and backward algorithm results when all heterozygotes are masked. When we estimate the probability of a haplotype by using Equation A4 of Appendix A, the HMM forward calculations are identical to HMM forward calculations with all heterozygotes masked until the first non-masked heterozygote is encountered. Similarly, the HMM backward calculations are identical to HMM backward calculations with all heterozygotes masked until the last non-masked heterozygote is encountered.

The HMM model state space for each individual is constructed from a fixed number of composite reference haplotypes (see composite reference haplotypes below). Consequently, the computational complexity of the HMM calculations for each individual is fixed, and the computation time for updating the phase of all individuals scales linearly with the number of individuals.

In each iteration, after a sample's haplotype phase is updated, missing alleles are imputed with haploid imputation. For each possible allele, we sum the HMM state probabilities for states (i.e., reference haplotypes) that carry that allele and then impute the missing allele to be the allele with maximal probability.[1–4,30,35]

Burn-in iterations

We obtain an initial phasing by using the SHAPEIT 4.0 initialization algorithm that is based on the positional Burrows-Wheeler transform (PBWT).[26,27] This initialization algorithm phases one marker at a time in chromosome order. When phasing marker m, the reversed haplotypes for the preceding phased markers (from marker $m-1$ to marker 1) are sorted in lexicographic order with the PBWT. Homozygous genotypes at marker m determine the alleles carried on that individual's haplotypes, and these alleles are assigned to haplotypes that are nearby in the PBWT sorting via an algorithm that ensures the assignment is consistent with the genotype data.[27] Once alleles are assigned to all haplotypes at marker m, the marker is phased, and the algorithm proceeds to the next marker.

After obtaining an initial phasing, Beagle 5.2 performs three burn-in iterations. During each burn-in iteration, Beagle updates the phase of each heterozygous genotype (see updating phase) but does not mark any heterozygotes as finished (see progressive phasing). If fewer than 1% of heterozygotes have their phase changed in a burn-in iteration, the remaining burn-in iterations are skipped.

Composite reference haplotypes

In each iteration, we construct a set of composite reference haplotypes[3] for each individual. These composite reference haplotypes are the reference haplotypes in the HMM that is used for updating the individual's haplotypes. Each composite reference haplotype is a mosaic of haplotype segments from the estimated haplotypes in other individuals at the start of the iteration. Each of these haplotype segments contains an allele sequence that the target individual shares identical by state with another individual.

The algorithm for constructing a set of composite reference haplotypes has been described previously in the context of genotype imputation.[3] In the remainder of this section, we summarize the construction algorithm and describe the modifications that we make to the algorithm parameters when constructing composite reference haplotypes for genotype phasing.

The algorithm parameters are as follows:

(1) the number, J, of composite reference haplotypes to be constructed;

(2) non-overlapping intervals, $I_1, I_2, \ldots, I_K$, that partition the marker window;

(3) sets $S_1, S_2, \ldots, S_K$ of haplotypes such that haplotypes in S_k are identical by state with the target haplotype in interval I_k.

For application to genotype phasing, we make the following changes to the values of these parameters. We construct $J = 140$ composite reference haplotypes for each haplotype in the target individual. We combine the composite reference haplotypes for each haplotype in the target individual to obtain an HMM state space of 280 composite reference haplotypes.

We define the intervals $I_1, I_2, \ldots, I_K$ by partitioning the marker window into non-overlapping intervals such that each interval includes all markers whose cM distance from the first marker in the interval is less than three times the median inter-marker cM distance in the marker window.

We define the set S_k to be a singleton set with one haplotype from another individual that has the same allele sequence as the target haplotype in interval I_k. If there are no other haplotypes with the same allele sequence in interval I_k, S_k is the empty set. We use the PBWT[26] to identify a set of T haplotypes from other individuals that have the longest allele sharing with the target haplotype, where length is measured in number of intervals, starting from I_k and working backward toward the first interval. We select a random haplotype that is identical by state with the target haplotype in interval I_k from these T haplotypes, and this random haplotype is the singleton element of the set S_k. The value of T is 100 in the burning iterations and decreases linearly with each phasing iteration, starting with $T = 90$ in the first phasing iteration and ending with $T = 5$ in the final phasing iteration.

Once the sets S_k are determined, we construct the set of J composite reference haplotypes. Each composite reference haplotype is a sequence of haplotype segments. The sequence of haplotype segments is represented by a list of the first markers and a list of the haplotypes in the sequence of segments. The last marker in a segment is the marker preceding the first marker in the following segment or the last marker in the window if there is no following segment. We add a new haplotype segment to a composite reference haplotype by adding the first marker in the segment to the list of first markers and the haplotype to the list of haplotypes.

We process the S_k in order of increasing k. Each non-empty, singleton set $S_k = \{h\}$ generates a haplotype segment that is added to a composite reference haplotype. The copied haplotype is h. The starting marker is midway between the first marker in the interval I_k for the current S_k and the first marker in the most recent interval $I_{k'}$ ($k' < k$) whose singleton set $S_{k'}$ contains the haplotype in the preceding segment of the composite reference haplotype.[3]

Computation time for construction of composite reference haplotypes scales linearly with sample size because the PBWT scales linearly with sample size, and the S_k sets are obtained for all target haplotypes via a single run of the PBWT algorithm.

Two-stage phasing

Sequence data from large samples of individuals contain many markers with very low minor allele frequency. Beagle divides markers into low- and high-frequency markers depending on whether all non-major alleles have frequency < 0.002 or ≥ 0.002 (markers may be multi-allelic). If less than 25% of the markers in a window have low frequency, Beagle phases all variants via the progressive phasing algorithm described above. If more than

25% of the markers in a window have low frequency, Beagle employs a two-stage phasing algorithm that substantially reduces the computational effort required to phase the low-frequency markers.

In the first stage, Beagle ignores low-frequency markers and phases the high-frequency markers via the progressive phasing algorithm described above. In the second stage, Beagle uses the phased high-frequency markers and Beagle's genotype imputation methodology[3,30] to phase the remaining low-frequency heterozygotes. The second stage performs the HMM forward-backward algorithm once per sample and does not use an iterative algorithm. There are two important differences between our use of imputation for phasing and the imputation of ungenotyped markers: our method does not require an external reference panel, and we use imputation to infer the phase of heterozygous genotypes rather than the alleles in missing genotypes.

Because Beagle's genotype imputation method has been described previously,[3] we outline the parts of the second stage algorithm that are borrowed without change, and focus on describing how Beagle's genotype imputation method is modified to phase low-frequency heterozygous genotypes.

For each target haplotype, we construct a reference panel of composite reference haplotypes at high-frequency markers from the estimated haplotypes in the other individuals. We estimate the HMM haplotype state probabilities by using the HMM forward-backward algorithm at the high-frequency markers.[33] We then use linear interpolation on genetic distance to estimate HMM state probabilities (probabilities of which composite reference haplotype is being copied) at each low-frequency marker for which the target sample has a heterozygous or missing genotype.[30]

For each composite reference haplotype, we must assign an allele to the haplotype at each low-frequency marker because low-frequency markers were not phased in the first stage. If the reference individual contributing the haplotype segment to the composite reference haplotype is homozygous for an allele at the low-frequency marker, the composite reference haplotype carries that allele. If the reference individual is heterozygous, we assign the lower-frequency allele if the target individual carries that allele and the higher-frequency allele otherwise.

For each haplotype in a target individual, we obtain posterior allele probabilities at a missing or heterozygous low-frequency marker by summing the state probabilities for all reference haplotypes that have been assigned the same allele.[3,30] If the genotype is missing, we choose the allele with highest probability for each target haplotype. If the genotype is heterozygous, we assume Hardy-Weinberg equilibrium and multiply posterior allele probabilities to calculate the posterior probability of the two possible phased heterozygous genotypes. We then choose the phased heterozygous genotype with higher posterior probability.

A phased haplotype scaffold was also used in the 1000 Genomes Project[36] for estimating phased haplotypes from genotype likelihoods.[37] In the 1000 Genomes Project, external SNP array data for samples were phased separately for production of a haplotype scaffold. During phasing of the 1000 Genomes Project sequence data, estimated haplotypes in each iteration were required to be consistent with this scaffold. In our two-stage method, the input data are genotypes instead of genotype likelihoods, the haplotype scaffold is generated from internal data instead of external data, non-scaffold markers are phased with genotype imputation instead of an iterative phasing algorithm and the two-stage algorithm is used for decreasing computation time rather than improving haplotype accuracy.

Phasing of IBD2 regions

Special care must be taken when phasing a pair of individuals in regions where the two individuals share both haplotypes identically by descent. Such regions are called IBD2 regions, and they commonly occur in full siblings. Two individuals have identical genotypes in an IBD2 region except at sites where an allele is miscalled or a mutation has arisen since the common ancestor. Consequently, both individuals can have identical estimated haplotypes in an IBD2 region. If the estimated haplotypes in the two individuals are identical, they can contain many phase errors and yet still be the most probable haplotypes determined by the HMM.

To protect against elevated phase error rates in IBD2 regions, Beagle 5.2 does not allow either related individual to contribute a haplotype segment to a composite reference haplotype for the other related individual in an IBD2 region.

Beagle detects IBD2 regions by using markers with minor allele frequency ≥ 0.05. These markers are thinned to ensure a minimum 0.002 cM inter-marker spacing and divided into disjoint intervals that are 1 cM in length, unless there are less than 50 markers in an interval, in which case, the interval is extended to include 50 markers.

For each pair of individuals, we select all intervals that have concordant genotypes at all markers (i.e., the same genotype in both individuals or a missing genotype in at least one individual). We merge any of these intervals separated by a gap of less than 4 cM into a single interval that contains the intervening gap. We then extend the boundaries of the intervals to include any adjacent markers with concordant genotypes. If any interval is longer than 2 cM after this extension, the interval is recorded as an IBD2 segment for the pair of individuals.

The computation time for IBD2 segment detection scales linearly with increasing sample size if the number of pairs of individuals with concordant genotypes in the 1 cM intervals scales linearly with the number of individuals. IBD2 segment detection scales linearly with sample size for the UK Biobank SNP array data and TOPMed sequence data analyzed in this study.

UK Biobank data

We downloaded the UK Biobank autosomal genotype data. After we removed withdrawn individuals, there were 488,332 individuals and 784,256 autosomal diallelic markers before quality control filtering. We excluded markers with more than 5% missing genotypes ($n = 70{,}247$), markers that had only one individual carrying a minor allele ($n = 5{,}126$), and markers that failed one or more of the UK Biobank's batch quality control tests ($n = 1{,}527$).[31] There were 711,651 autosomal markers after excluding markers that failed one or more of these filters.

We then excluded 968 individuals that were identified by the UK Biobank as outliers with respect to their proportion of missing genotypes or proportion of heterozygous genotypes, and we excluded nine individuals that were identified by the UK Biobank as showing third degree or closer relationships with more than 200 individuals, which indicates possible sample contamination. There were 487,355 individuals remaining after these exclusions.

We identified parent-offspring trios by using the kinship coefficients and the proportion of markers that share no alleles (IBS0) that are reported by the UK Biobank.[31,38] Pairs of individuals with kinship coefficient between $2^{-2.5}$ and $2^{-1.5}$ were considered to be first-degree relatives. First-degree relatives with IBS0 < 0.0012 were considered to have a parent-offspring relationship. These are the same kinship coefficient and IBS0 thresholds

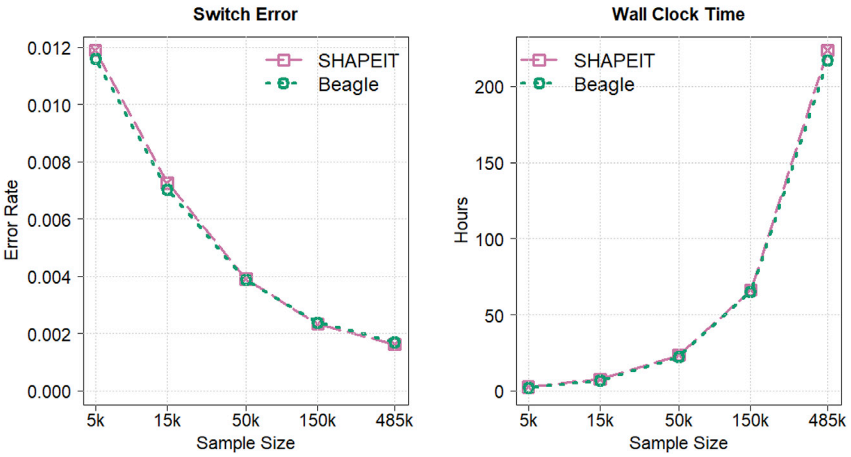


Figure 2. Phase accuracy and computation time for autosomal UK Biobank SNP array data
Switch error rate and wall clock computation time for Beagle 5.2 and SHAPEIT 4.2.1 when phasing 5,000, 15,000, 50,000, 150,000, and 485,301 UK Biobank individuals genotyped for 711,651 autosomal markers with default parameter values. Sample size is plotted on the log scale. Switch error rate is calculated with heterozygous genotypes in 1,064 offspring whose phase is determined from parental data that were excluded from the phasing analysis. All analyses were run with 20 threads on a computer server with 20 CPU cores and 256 GB memory.

used by the UK Biobank.[31] We considered an individual to be the offspring in a parent-offspring trio if the individual has a parent-offspring relationship with exactly one male and exactly one female (the putative parents) and the kinship coefficient for the male and female is less than $2^{-4.5}$. Using this procedure, we identified 1,064 parent-offspring trios having 2,054 distinct parents.

We phased heterozygous genotypes in trio offspring by using Mendelian inheritance constraints when both parent genotypes are non-missing and at least one parent genotype is homozygous. We used these phased genotypes in the trio offspring as the truth when estimating the phase error rate.

We then excluded the 2,054 trio parents, leaving 485,301 individuals. We listed the 1,064 trio offspring followed by the remaining 484,237 individuals in random order. We created five datasets by restricting the UK Biobank data to the first 5,000, 15,000, 50,000, 150,000, and 485,301 individuals in this list.

TOPMed project data

We downloaded Freeze 8 data from the Trans-Omics for Precision Medicine (TOPMed) Program[32] for the following studies and dbGaP[39] accession numbers: Barbados Asthma Genetics Study (dbGaP: phs001143), Mount Sinai BioMe Biobank (dbGaP: phs001644), Cleveland Clinic Atrial Fibrillation Study (dbGaP: phs001189), Framingham Heart Study (dbGaP: phs000974), Hypertension Genetic Epidemiology Network (dbGaP: phs001293), Jackson Heart Study (dbGaP: phs000964), My Life Our Future (dbGaP: phs001515), Severe Asthma Research Program (dbGaP: phs001446), Venous Thromboembolism Project (dbGaP: phs001402), Vanderbilt Genetic Basis of Atrial Fibrillation (dbGaP: phs001032), and Women's Health Initiative (dbGaP: phs001237).

We merged and filtered the TOPMed data by using bcftools.[40] The merged data contain 39,961 sequenced individuals. We restricted the data to polymorphic SNVs with "PASS" in the VCF filter field, leaving 318,858,817 autosomal markers, which include 7,209,890 chromosome 20 markers.

We used the pedigree data for the 1,022 sequenced Barbados Asthma Genetics Study (BAGS) individuals and for the 4,166 sequenced Framingham Heart Study (FHS) individuals to identify 217 BAGS and 669 FHS parent-offspring trios for which the offspring was not a parent in another parent-offspring trio. We phased heterozygous genotypes in trio offspring by using Mendelian inheritance constraints when both parent genotypes are non-missing and at least one parent genotype is homozygous. We used these phased genotypes in the trio offspring as the truth when estimating the phase error rate.

We then excluded the 1,574 parents of the trio offspring, leaving 38,387 individuals. We listed the 886 trio offspring followed by the remaining 37,501 individuals in a random order. We created four datasets by restricting the TOPMed data to the first 5,000, 10,000, 20,000, and 38,387 individuals in this list.

Results

We phased UK Biobank SNP array data and TOPMed sequence data by using Beagle 5.2 (28Jun21.202 release) and SHAPEIT 4.2.1. We ran each method with default parameters. SHAPEIT's default parameters for sequence data were applied by including its "–sequencing" argument when phasing TOPMed sequence data.

All analyses were run on a 20-core 2.4 GHz computer with Intel Xeon E5-2640 processors and 256 GB of memory. Beagle and SHAPEIT were each run with 20 computational threads. Wall clock computation time was measured with the Linux time command.

We measured phase error by using switch error rate, which is the proportion of heterozygous genotypes that are phased incorrectly with respect to the preceding heterozygous genotype.

Figure 2 shows autosomal switch error rate and wall clock computation time when phasing expanding subsets of 485,301 UK Biobank individuals. Beagle 5.2 and SHAPEIT 4.2.1 have very similar phase error and computation time for all sample sizes. Beagle's computation time for phasing the UK Biobank SNP array data scales linearly with sample size, and Beagle's phase error decreases with increasing sample size.

Figure 3 shows switch error rate and computation time for chromosome 20 when phasing expanding subsets of 38,387 sequenced TOPMed individuals. The left panel shows switch error rate in the BAGS trio offspring, the middle panel shows switch error rate in the FHS trio offspring, and the right panel shows wall clock computation time. As with the UK Biobank data, the switch error rate decreases with increasing sample size. SHAPEIT results for phasing

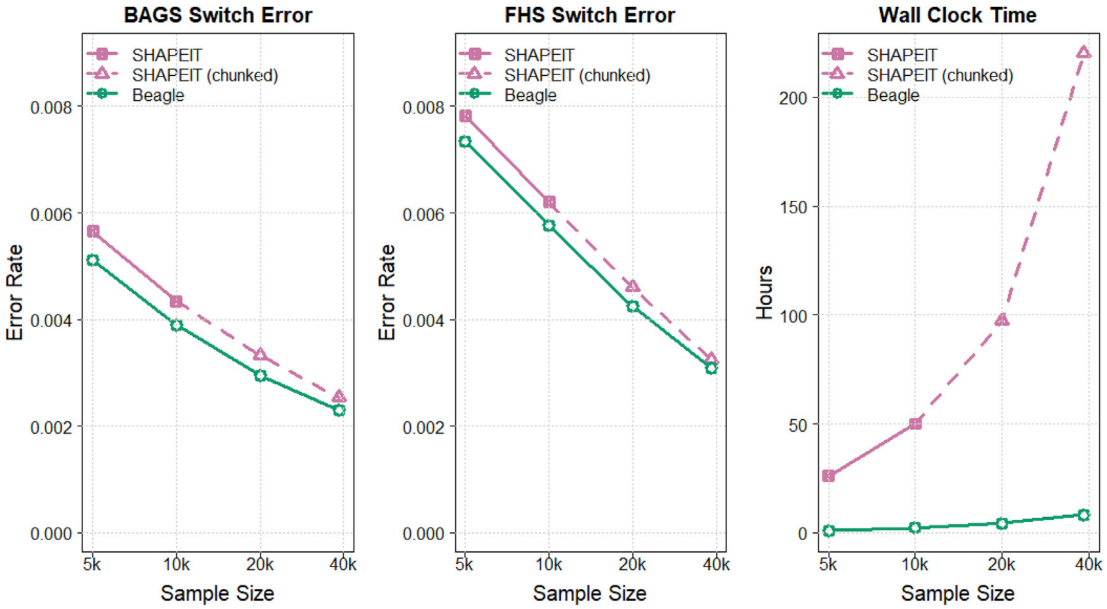


Figure 3. Phase accuracy and computation time for phasing TOPMed chromosome 20 sequence data
Switch error rate and wall clock computation time for Beagle 5.2 and SHAPEIT 4.2.1 when phasing 5,000, 10,000, 20,000, and 38,387 sequenced TOPMed individuals genotyped for 7,209,890 chromosome 20 markers with default parameter values. Sample size is plotted on the log scale. Switch error rate is computed with heterozygous genotypes in 217 BAGS and 669 FHS offspring whose phase is determined from parental data that were excluded from the phasing analysis. All analyses were run with 20 threads on a computer server with 20 CPU cores and 256 GB of memory. The SHAPEIT whole-chromosome phasing of the 20,000 and 38,387 individuals did not complete because of insufficient memory. SHAPEIT results for the 20,000 and 38,387 individuals were obtained by dividing the chromosome into two and three chunks, respectively, and phasing each chunk separately. Adjacent chunks had 500 kb overlap. The SHAPEIT wall clock for each sample size is the sum of the wall clock times for the individual chunks. The procedure for merging the individual chunks for each sample size is described in the subjects and methods.

chromosome 20 in its entirety are available only for the 5,000 and 10,000 sequenced individuals because SHAPEIT could not phase the 20,000 and 38,387 sequenced individuals within the 256 GB of available computer memory when default parameters were used. Consequently, we ran a chunked SHAPEIT phasing analysis by dividing the chromosome into two segments for the 20,000 samples and dividing the chromosome into three segments for the 38,387 samples. We included 500 kb of overlap between adjacent segments. After the chunked phasing analysis, we created a single phased VCF output file by aligning the haplotypes in adjacent chunks by using the phase of a heterozygote in the middle of the overlap and then splicing haplotypes in adjacent chunks with a splice point in the middle of the overlap.

Beagle's computation time for phasing the TOPMed sequence data scales linearly with sample size, and Beagle's phase error decreases with increasing in sample size. Beagle 5.2 and SHAPEIT 4.2.1 have similar error rates on the TOPMed sequence data, but Beagle's computation time ranged from 23.0 times faster (for 20,000 samples) to 26.7 times faster (for 38,387 samples). For each sample size, the phase error rate is lower in the BAGS trio offspring than in the FHS trio offspring, which could be due to differences in population demographic history.

The computational efficiency of two-stage phasing in the sequence data can be seen by comparing the wall clock time for phasing 5,000 sequenced TOPMed samples and 5,000 UK Biobank samples. The TOPMed chromosome 20 data have 7,209,890 markers and the UK Biobank chromosome 20 data have 18,424 markers. Although the TOPMed sequence data have 391-fold more markers than the UK Biobank SNP array data, there is only a 17-fold difference in wall clock time (62.4 versus 3.6 min). For the TOPMed sequence data, Beagle's wall clock time for the second stage of phasing is approximately one-third of the wall clock time for the first stage of phasing.

One can reduce Beagle's memory requirements to permit analysis of larger datasets by shortening the marker window. The default window length is 40 cM. We repeated the Beagle phasing of the 38,387 TOPMed individuals with 5, 10, 20, and 40 cM window lengths and default values for all other parameters. For each window length, we ran phasing analyses with different limits on the amount of memory available to the Java virtual machine in order to determine the minimum amount of memory (in units of 5 GB) required to successfully complete the analysis.

Figure 4 shows results for 5, 10, 20, and 40 cM marker windows. Accuracy is similar for all window lengths. Reducing the window length from the default 40 cM to 10 cM reduces memory use by 57% (from 185 GB to 80 GB) at the cost of a 28% increase in running time. Further reducing the window length from 10 cM to 5 cM gives an

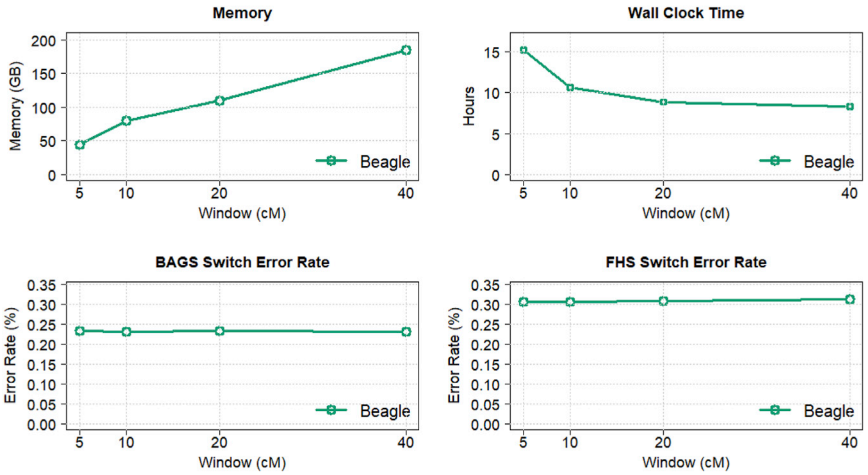


Figure 4. Memory and computation time as a function of window length
Beagle 5.2 memory use, wall clock time, and switch error rate when phasing 38,387 sequenced TOPMed individuals genotyped for 7,209,890 chromosome 20 markers when using 5, 10, 20, and 40 cM marker windows. The default window length is 40 cM. All other parameters were set to default values. Switch error rate is computed with heterozygous genotypes in 217 BAGS and 669 FHS offspring whose phase is determined from parental data that were excluded from the phasing analysis. All analyses were run with 20 threads on a computer server with 20 CPU cores and 256 GB of memory.

additional 44% reduction in memory use (from 80 GB to 45 GB) and an additional 43% increase in running time. Computational efficiency decreases as the window length decreases because the 2 cM overlap between windows occupies an increasing proportion of each window.

We tested Beagle's methods for estimating the effective population size parameter used in the HMM transition probabilities by using Beagle and SHAPEIT to phase expanding samples of the chromosome 20 UK Biobank data. For each program, we varied the user-specified effective population size parameter to range from three orders of magnitude smaller than the default value to three orders of magnitude larger than the default value. All other parameters were set to their default values. Figure 5 shows that Beagle's parameter estimation results in phase accuracy that is independent of the user-specified effective population size. The SHAPEIT results in Figure 5 illustrate that without such estimation the switch error rate can be inflated when the user-specified effective population size is too large unless the sample size is very large.

Discussion

We have introduced a haplotype phasing method that is implemented in Beagle 5.2. The computation time for this phasing method scales linearly with sample size, and it can phase hundreds of thousands of array-genotyped samples and tens of thousands of sequenced samples.

We compared phase accuracy by using default settings for Beagle 5.2 with SHAPEIT 4.2.1 when phasing UK Biobank autosomal SNP array data[31] and TOPMed chromosome 20 sequence data.[32] Both methods had similar accuracy on both types of data and similar computation time when phasing SNP array data. However, when phasing sequence data, Beagle was more than 20 times faster and was able to analyze larger samples within the available computer memory.

Beagle uses a sliding marker window that limits memory use and enables whole-chromosome phasing of large sequence datasets in a single analysis. Memory use can be decreased without loss of accuracy by decreasing the window length. For the TopMed data, a reduction of window length from 40 cM to 5 cM reduced memory use by 76% at the cost of an 82% increase in computation time.

Beagle version 5.2 introduces a two-stage, progressive phasing methodology, and it incorporates methodological ideas that were originally developed for genotype imputation, such as composite reference haplotypes[3] and linear interpolation of HMM state probabilities.[30]

Progressive phasing identifies confidently phased heterozygotes and fixes the phase of these heterozygotes in subsequent iterations. As a result, the information available to phase the heterozygotes with more ambiguous phasing increases with each phasing iteration.

Two-stage phasing gives a large reduction in computation time when a high proportion of variants have low frequency, as is generally the case with sequence data. The first stage phases the limited number of high-frequency variants and these variants become a haplotype scaffold for imputing alleles at low-frequency variants in the second stage. The two-stage approach reduces computation time because the low-frequency markers are excluded from the time-consuming iterative phasing algorithm.

For non-human species, the effective population size may be unknown. Beagle 5.2 protects against a misspecified effective population size by estimating and updating the effective population size parameter during its burn-in iterations. The Mach phasing method also estimates and updates this parameter.[20] Beagle calculates its estimate from HMM state probabilities as described in Appendix B, while Mach calculates its estimate by using data obtained from sampled haplotypes.[20]

Memory continues to be a constraint when phasing large sequence datasets, and reducing memory requirements is an area for future work. Improved memory efficiency would enable larger datasets to be analyzed on computers that have less memory. Beagle 5.2 is implemented in Java, and memory deallocation is under the control of the Java virtual machine. Implementing Beagle in a systems

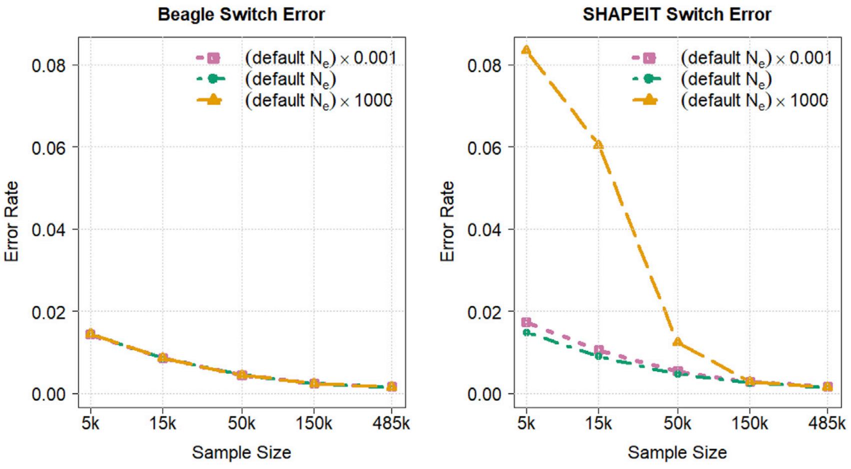


Figure 5. Phase accuracy as a function of user-specified effective population size
Switch error rate for Beagle 5.2 and SHAPEIT 4.2.1 when phasing 5,000, 15,000, 50,000, 150,000, and 485,301 UK Biobank individuals genotyped for 18,424 chromosome 20 markers for three different user-specified values of the effective population size parameter: the program's default parameter value, a value 1,000 times smaller than the default value, and a value 1,000 times larger than the default value. All other analysis parameters are set to their default values. Switch error rate is calculated with heterozygous genotypes in 1,064 offspring whose phase is determined from parental data that were excluded from the phasing analysis. Beagle's switch error rate does not depend on the user-specified effective population size parameter because Beagle estimates and updates this parameter.

programming language that gives fast execution times and fine-grained control of memory has the potential to significantly reduce memory use and computation time. Another important area for future work will be adding the capability to incorporate existing phase information, such as phase information extracted from sequence reads.[27] Extraction of phase information from sequence reads is feasible for smaller sample sizes that have a manageable amount of sequence read data.

As genotype datasets have grown, phasing methodology has responded. The release of the UK Biobank SNP array data[31] motivated the development of a series of statistical phasing methods that advanced the state of the art, including Eagle1,[23] SHAPEIT3,[41] Eagle2,[28] Beagle 5.1, and SHAPEIT4.[27] The advent of whole-genome sequence datasets has provided a new impetus for further development of phasing methodology. We show that Beagle 5.2 scales to tens of thousands of sequenced individuals. Further advances in phasing methodology will be needed to address the challenge of future datasets with hundreds of thousands or millions of sequenced individuals.

Appendix A: Hidden Markov model

We use the Li and Stephens HMM to calculate the probability of a haplotype of alleles from M markers.[3,20,29,34] The model states are a set of H reference haplotypes. Let $\mathcal{S}_m \in \{1, 2, \ldots, H\}$ denote the unobserved HMM state at marker $m \in \{1, 2, \ldots, M\}$. In the Li and Stephens HMM, a target haplotype is modeled as a mosaic of reference haplotype segments, and the state $\mathcal{S}_m$ represents the reference haplotype being copied at marker m. The initial HMM probabilities are $P(\mathcal{S}_1 = h) = 1/H$ for each reference haplotype h. The emission probabilities are determined by the reference haplotype alleles. A state $\mathcal{S}_m = h$ emits the allele carried by reference haplotype h at marker m with probability $(1-\varepsilon)$ and emits a different allele with probability ε, where $\varepsilon = \theta/(2\theta + 2H)$ and $\theta = 1/(\log H + 0.5)$.[34]

Transition probabilities depend on the genetic distance, effective population size N_e, and number of reference haplotypes H.[34] If markers $m-1$ and m are separated by d_m Morgans and if

$$\tau_m = 1 - e^{-4N_e d_m/H}, \qquad \text{(Equation A1)}$$

then the state transition probabilities are defined as[34]

$$P(\mathcal{S}_m = h' | \mathcal{S}_{m-1} = h) = \begin{cases} \tau_m/H, \ h \neq h' \\ (1-\tau_m) + \tau_m/H, \ h = h' \end{cases}. \qquad \text{(Equation A2)}$$

An observed haplotype O is a sequence of alleles $O_1, O_2, \ldots, O_M$ where O_m denotes the allele at the m-th marker. Because the observed haplotype is assumed to correspond to an unobserved sequence of HMM states $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M$, we can use the HMM forward-backward algorithm[33] to calculate the haplotype probability $P(O)$. The forward probabilities $\alpha_m(h)$ and backward probability $\beta_m(h)$ for reference haplotype h at marker m for the observed haplotype O are defined as

$$\alpha_m(h) = P(\mathcal{S}_m = h, O_1, O_1, \ldots, O_m)$$

$$\beta_m(h) = P(O_{m+1}, O_{m+2}, \ldots, O_M | \mathcal{S}_m = h). \qquad \text{(Equation A3)}$$

These probabilities are calculated via a dynamic programming algorithm.[33] For any marker m, the probability of the observed allele sequence is

$$P(O) = \sum_{h=1}^{H} P(O_1, O_2, \ldots, O_M, \mathcal{S}_m = h) = \sum_{h=1}^{H} \alpha_m(h)\beta_m(h). \qquad \text{(Equation A4)}$$

The probability of an individual state, conditional on the observed data, is

$$P(\mathcal{S}_m = h | O) = \frac{\alpha_m(h)\beta_m(h)}{\sum_{h'=1}^{H} \alpha_m(h')\beta_m(h')}. \qquad \text{(Equation A5)}$$

We use Equation A5 to impute missing alleles on a haplotype. We obtain posterior allele probabilities by summing the state probabilities for all reference haplotypes that carry the same allele and choosing the allele with highest posterior probability.[1,30]

Appendix B: Estimating effective population size

The HMM transition probabilities in Equation A2 of Appendix A depend on the effective population size, which may be unknown for non-human populations. Consequently, after the initial haplotypes are determined and after each burn-in iteration, Beagle 5.2 estimates and updates the effective population size via an iterative expectation-maximization-type procedure.[33,42]

At each marker m, Beagle estimates τ_m by setting $h = h'$ in Equation A2, conditioning on an observed haplotype O, and then summing the conditional transition probabilities for the haplotypes weighted by the state probabilities $P(\mathcal{S}_{m-1} = h|O)$:

$$\begin{aligned}(1-\tau_m)+\frac{\tau_m}{H} &= \sum_{h=1}^{H} P(\mathcal{S}_m = h|\mathcal{S}_{m-1} = h, O)P(\mathcal{S}_{m-1} = h|O)\\ &= \sum_{h=1}^{H} P(\mathcal{S}_m = h, \mathcal{S}_{m-1} = h|O)\\ &= \frac{1}{P(O)}\sum_{h=1}^{H} P(\mathcal{S}_m = h, \mathcal{S}_{m-1} = h, O).\end{aligned}$$

Solving the preceding equation for τ_m gives the estimate

$$\widehat{\tau}_m = \frac{H}{H-1}\left(1-\frac{1}{P(O)}\sum_{h=1}^{H} P(\mathcal{S}_m = h, \mathcal{S}_{m-1} = h, O)\right).$$

We calculate $P(O)$ by using Equation A4. We calculate $P(\mathcal{S}_m = h,\ \mathcal{S}_{m-1} = h, O)$ by using the forward and backward probabilities defined in Equation A3 and the transition probability defined by Equations A1 and A2 and the current value of the N_e parameter:

$$\begin{aligned}P(\mathcal{S}_m = h, \mathcal{S}_{m-1} = h, O) &= \beta_m(h)P(O_m|\mathcal{S}_m = h)\\ &\times P(\mathcal{S}_m = h|\mathcal{S}_{m-1} = h)\alpha_{m-1}(h) = \beta_m(h)\\ &\times P(O_m|\mathcal{S}_m = h)\left((1-\tau_m)+\frac{\tau_m}{H}\right)\alpha_{m-1}(h).\end{aligned}$$

For small genetic distances, $\tau_m \approx \left(\frac{4N_e}{H}\right)d_m$ by Equation A1. We select a random set of 500 samples or all samples if there are fewer than 500 samples. After we estimate the $\widehat{\tau}_{m,h}$ for each marker m and each haplotype h in these samples, we estimate the effective population size N_e as

$$\widehat{N}_e = \frac{H\sum_{m,h}\widehat{\tau}_{m,h}}{4\sum_{m,h}d_m}.$$

We update the effective population size parameter N_e to be the estimated value. This procedure for updating the N_e parameter is repeated until a stopping condition is met. The iterative procure stops when the updated value of N_e results in less than a 10% change in the value of $\left(\frac{4N_e}{H}\right)$.

Data and code availability

Beagle is licensed under the GNU General Public License (version 3) and is freely available for academic and commercial use. The Beagle 5.2 source code is written in Java and can be downloaded from https://faculty.washington.edu/browning/beagle/beagle.html.

Supplemental information

Supplemental information can be found online at https://doi.org/10.1016/j.ajhg.2021.08.005.

Acknowledgments

Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under award number HG008359. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This research has been conducted with the UK Biobank Resource under application number 19934. Molecular data for the Trans-Omics in Precision Medicine (TOPMed) program was supported by the National Heart, Lung, and Blood Institute (NHLBI). Core support including centralized genomic-read mapping and genotype calling along with variant quality metrics and filtering were provided by the TOPMed Informatics Research Center (3R01HL-117626-02S1; contract HHSN268201800002I). Core support including phenotype harmonization, data management, sample-identity QC, and general program coordination were provided by the TOPMed Data Coordinating Center (R01HL-120393; U01HL-120393; contract HHSN268201800001I). We gratefully acknowledge the studies and participants who provided biological samples and data for TOPMed. Funding for the Barbados Asthma Genetics Study was provided by National Institutes of Health (NIH) R01HL104608, R01HL087699, and HL104608 S1. The Framingham Heart Study was supported by contracts NO1-HC-25195, HHSN268201500001I, and 75N92019D00031 from the NHLBI and grant supplement R01 HL092577-06S1; genome sequencing was funded by HHSN268201600034I and U54HG003067. See supplemental information for acknowledgments of additional individual studies in the TOPMed data.

Declaration of interests

The authors declare no competing interests.

Received: May 21, 2021
Accepted: August 10, 2021
Published: September 2, 2021

Web resources

Beagle 5.2, https://faculty.washington.edu/browning/beagle/beagle.html

References

1. Howie, B., Fuchsberger, C., Stephens, M., Marchini, J., and Abecasis, G.R. (2012). Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. Nat. Genet. *44*, 955–959.
2. Das, S., Abecasis, G.R., and Browning, B.L. (2018). Genotype Imputation from Large Reference Panels. Annu. Rev. Genomics Hum. Genet. *19*, 73–96.
3. Browning, B.L., Zhou, Y., and Browning, S.R. (2018). A One-Penny Imputed Genome from Next-Generation Reference Panels. Am. J. Hum. Genet. *103*, 338–348.
4. Rubinacci, S., Delaneau, O., and Marchini, J. (2020). Genotype imputation using the Positional Burrows Wheeler Transform. PLoS Genet. *16*, e1009049.
5. Larsen, L.A., Fosdal, I., Andersen, P.S., Kanters, J.K., Vuust, J., Wettrell, G., and Christiansen, M. (1999). Recessive Romano-Ward syndrome associated with compound heterozygosity for two mutations in the KVLQT1 gene. Eur. J. Hum. Genet. *7*, 724–728.
6. Browning, B.L., and Browning, S.R. (2007). Efficient multilocus association testing for whole genome association studies using localized haplotype clustering. Genet. Epidemiol. *31*, 365–375.
7. Browning, B.L., and Browning, S.R. (2008). Haplotypic analysis of Wellcome Trust Case Control Consortium data. Hum. Genet. *123*, 273–280.
8. Gusev, A., Lowe, J.K., Stoffel, M., Daly, M.J., Altshuler, D., Breslow, J.L., Friedman, J.M., and Pe'er, I. (2009). Whole population, genome-wide mapping of hidden relatedness. Genome Res. *19*, 318–326.
9. Zhou, Y., Browning, S.R., and Browning, B.L. (2020). A Fast and Simple Method for Detecting Identity-by-Descent Segments in Large-Scale Data. Am. J. Hum. Genet. *106*, 426–437.
10. Browning, S.R., and Browning, B.L. (2020). Probabilistic Estimation of Identity by Descent Segment Endpoints and Detection of Recent Selection. Am. J. Hum. Genet. *107*, 895–910.
11. Maples, B.K., Gravel, S., Kenny, E.E., and Bustamante, C.D. (2013). RFMix: a discriminative modeling approach for rapid and robust local-ancestry inference. Am. J. Hum. Genet. *93*, 278–288.
12. Baran, Y., Pasaniuc, B., Sankararaman, S., Torgerson, D.G., Gignoux, C., Eng, C., Rodriguez-Cintron, W., Chapela, R., Ford, J.G., Avila, P.C., et al. (2012). Fast and accurate inference of local ancestry in Latino populations. Bioinformatics *28*, 1359–1367.
13. Salter-Townshend, M., and Myers, S. (2019). Fine-Scale Inference of Ancestry Segments Without Prior Knowledge of Admixing Groups. Genetics *212*, 869–889.
14. Sabeti, P.C., Reich, D.E., Higgins, J.M., Levine, H.Z., Richter, D.J., Schaffner, S.F., Gabriel, S.B., Platko, J.V., Patterson, N.J., McDonald, G.J., et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. Nature *419*, 832–837.
15. Hanchard, N.A., Rockett, K.A., Spencer, C., Coop, G., Pinder, M., Jallow, M., Kimber, M., McVean, G., Mott, R., and Kwiatkowski, D.P. (2006). Screening for recently selected alleles by analysis of human haplotype similarity. Am. J. Hum. Genet. *78*, 153–159.
16. Zhang, C., Bailey, D.K., Awad, T., Liu, G., Xing, G., Cao, M., Valmeekam, V., Retief, J., Matsuzaki, H., Taub, M., et al. (2006). A whole genome long-range haplotype (WGLRH) test for detecting imprints of positive selection in human populations. Bioinformatics *22*, 2122–2128.
17. Scheet, P., and Stephens, M. (2006). A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. Am. J. Hum. Genet. *78*, 629–644.
18. Browning, S.R., and Browning, B.L. (2007). Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. Am. J. Hum. Genet. *81*, 1084–1097.
19. Kong, A., Masson, G., Frigge, M.L., Gylfason, A., Zusmanovich, P., Thorleifsson, G., Olason, P.I., Ingason, A., Steinberg, S., Rafnar, T., et al. (2008). Detection of sharing by descent, long-range phasing and haplotype imputation. Nat. Genet. *40*, 1068–1075.
20. Li, Y., Willer, C.J., Ding, J., Scheet, P., and Abecasis, G.R. (2010). MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. Genet. Epidemiol. *34*, 816–834.
21. Williams, A.L., Patterson, N., Glessner, J., Hakonarson, H., and Reich, D. (2012). Phasing of many thousands of genotyped samples. Am. J. Hum. Genet. *91*, 238–251.
22. Delaneau, O., Marchini, J., and Zagury, J.F. (2011). A linear complexity phasing method for thousands of genomes. Nat. Methods *9*, 179–181.
23. Loh, P.-R., Palamara, P.F., and Price, A.L. (2016). Fast and accurate long-range phasing in a UK Biobank cohort. Nat. Genet. *48*, 811–816.
24. Howie, B., Marchini, J., and Stephens, M. (2011). Genotype imputation with thousands of genomes. G3 (Bethesda) *1*, 457–470.
25. Delaneau, O., Zagury, J.F., and Marchini, J. (2013). Improved whole-chromosome phasing for disease and population genetic studies. Nat. Methods *10*, 5–6.
26. Durbin, R. (2014). Efficient haplotype matching and storage using the positional Burrows-Wheeler transform (PBWT). Bioinformatics *30*, 1266–1272.
27. Delaneau, O., Zagury, J.F., Robinson, M.R., Marchini, J.L., and Dermitzakis, E.T. (2019). Accurate, scalable and integrative haplotype estimation. Nat. Commun. *10*, 5436.
28. Loh, P.R., Danecek, P., Palamara, P.F., Fuchsberger, C., A Reshef, Y., K Finucane, H., Schoenherr, S., Forer, L., McCarthy, S., Abecasis, G.R., et al. (2016). Reference-based phasing using the Haplotype Reference Consortium panel. Nat. Genet. *48*, 1443–1448.
29. Li, N., and Stephens, M. (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. Genetics *165*, 2213–2233.
30. Browning, B.L., and Browning, S.R. (2016). Genotype Imputation with Millions of Reference Samples. Am. J. Hum. Genet. *98*, 116–126.
31. Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L.T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., et al. (2018). The UK Biobank resource with deep phenotyping and genomic data. Nature *562*, 203–209.
32. Taliun, D., Harris, D.N., Kessler, M.D., Carlson, J., Szpiech, Z.A., Torres, R., Taliun, S.A.G., Corvelo, A., Gogarten, S.M., Kang, H.M., et al. (2021). Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. Nature *590*, 290–299.
33. Rabiner, L.R. (1989). A Tutorial on Hidden Markov-Models and Selected Applications in Speech Recognition. Proc. IEEE *77*, 257–286.

34. Marchini, J., Howie, B., Myers, S., McVean, G., and Donnelly, P. (2007). A new multipoint method for genome-wide association studies by imputation of genotypes. Nat. Genet. *39*, 906–913.

35. Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A.E., Kwong, A., Vrieze, S.I., Chew, E.Y., Levy, S., McGue, M., et al. (2016). Next-generation genotype imputation service and methods. Nat. Genet. *48*, 1284–1287.

36. Auton, A., Brooks, L.D., Durbin, R.M., Garrison, E.P., Kang, H.M., Korbel, J.O., Marchini, J.L., McCarthy, S., McVean, G.A., Abecasis, G.R.; and 1000 Genomes Project Consortium (2015). A global reference for human genetic variation. Nature *526*, 68–74.

37. Delaneau, O., Marchini, J.; 1000 Genomes Project Consortium; and 1000 Genomes Project Consortium (2014). Integrating sequence and array data to create an improved 1000 Genomes Project haplotype reference panel. Nat. Commun. *5*, 3934.

38. Manichaikul, A., Mychaleckyj, J.C., Rich, S.S., Daly, K., Sale, M., and Chen, W.-M. (2010). Robust relationship inference in genome-wide association studies. Bioinformatics *26*, 2867–2873.

39. Mailman, M.D., Feolo, M., Jin, Y., Kimura, M., Tryka, K., Bagoutdinov, R., Hao, L., Kiang, A., Paschall, J., Phan, L., et al. (2007). The NCBI dbGaP database of genotypes and phenotypes. Nat. Genet. *39*, 1181–1186.

40. Danecek, P., Bonfield, J.K., Liddle, J., Marshall, J., Ohan, V., Pollard, M.O., Whitwham, A., Keane, T., McCarthy, S.A., Davies, R.M., and Li, H. (2021). Twelve years of SAMtools and BCFtools. Gigascience *10*, giab008.

41. O'Connell, J., Sharp, K., Shrine, N., Wain, L., Hall, I., Tobin, M., Zagury, J.-F., Delaneau, O., and Marchini, J. (2016). Haplotype estimation for biobank-scale data sets. Nat. Genet. *48*, 817–820.

42. Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. J. R. Stat. Soc. B *39*, 1–22.